A Pilot Study on Consumer IoT Device Vulnerability Disclosure and Patch Release in Japan and the United States

Asuka Nakajima
NTT Secure Platform Laboratories
asuka.nakajima.db@hco.ntt.co.jp

Takuya Watanabe
NTT Secure Platform Laboratories
takuya.watanabe.yf@hco.ntt.co.jp

Eitaro Shioji
NTT Secure Platform Laboratories
eitaro.shioji.es@hco.ntt.co.jp

Mitsuaki Akiyama
NTT Secure Platform Laboratories
akiyama@ieee.org

Maverick Woo
Carnegie Mellon University
pooh@cmu.edu

ABSTRACT

With our ever increasing dependence on computers, many governments have started to investigate regulations on vulnerabilities and their lifecycle management. Although many previous works have studied this problem space for mainstream software packages and web applications, few studies have targeted consumer IoT devices. As a first step towards filling this void, this paper presents a pilot study on the vulnerability disclosures and patch release behaviors related to 3 prominent consumer IoT vendors in Japan and 3 in the United States. Our goals include (i) characterizing trends and risks using accurate data that spans a long period, and (ii) identifying problems, challenges, and potential approaches for future studies of this problem space. To this end, we collected all published vulnerabilities and their patches for the consumer IoT products by the included vendors between 2006 and 2017; then, we analyzed our data from multiple perspectives such as the severity of the vulnerabilities and the timing of patch releases with respect to disclosures and exploits. Our work has uncovered several findings that may inform future studies, including (i) a stark contrast in the vulnerability disclosures between the two countries and (ii) three alarming vendor practices that may pose significant risks of 1-day exploits.

CCS CONCEPTS

• **Security and privacy → Vulnerability management**; • **Social and professional topics → Consumer products policy**; • **General and reference →** *Measurement*.

KEYWORDS

consumer IoT; vulnerability disclosure; patch; exploit; measurement

ACM Reference Format:
Asuka Nakajima, Takuya Watanabe, Eitaro Shioji, Mitsuaki Akiyama, and Maverick Woo. 2019. A Pilot Study on Consumer IoT Device Vulnerability Disclosure and Patch Release in Japan and the United States. In *ACM Asia Conference on Computer and Communications Security (AsiaCCS '19), July 9–12, 2019, Auckland, New Zealand.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3321705.3329849

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
AsiaCCS '19, July 9–12, 2019, Auckland, New Zealand
© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-6752-3/19/07…$15.00
https://doi.org/10.1145/3321705.3329849

1 INTRODUCTION

As our society continues to increase its reliance on computers large and small, vulnerabilities and their lifecycle management are gradually becoming a matter of public safety. In response, many governments have started to investigate regulating computer security through legislation/standards setting, e.g., [18, 22]. As we might expect, understanding the past and current practices of the stakeholders in IoT vulnerability lifecycle management and identifying potential improvements are important steps in these efforts.

Existing work on *vulnerability lifecycle management* can be classified into two groups based on where it falls with respect to the time of discovery of a vulnerability. At a high level, work in the *pre-discovery* group focuses on the prevention and discovery of vulnerabilities, whereas work in the *post-discovery* group focuses on the disclosure and notification of vulnerabilities as well as mitigations. In large part due to the myriad of ways and places for vulnerabilities to creep in and the depth and breadth of vulnerability prevention and discovery techniques, the literature in the pre-discovery group is vast and continues to expand rapidly.

In comparison, the literature in the post-discovery group is considerably smaller and there are two lines of work that are particularly relevant to this paper. The first line studies the *patch release behavior* of commercial and/or open-source developers (collectively referred to as "*vendors*"). For example, previous studies have investigated the timeliness and the prioritization of patch releases, the behavioral differences between commercial and open-source vendors, and external factors that may improve their behaviors [1, 2, 7–9, 26, 27]. The second line studies the *characteristics of patches and vulnerabilities*. This includes measurements of diverse properties such as the number of exploitations, the longevity of vulnerabilities, the size and complexity of their patches, and the rate of patch deployments [3, 6, 11, 17, 24, 30].

Incidentally, the overwhelming majority of the aforementioned studies were dominated by mainstream software packages or web-based applications. Based on our literature search, we are not aware of any prior work that focused on *consumer IoT devices*, many of which are products by vendors that have a small or even non-existent representation in the datasets used by previous studies. With the rapid rise of consumer IoT devices in recent years, we thus believe it is high time to expand our knowledge in vulnerability lifecycle management in regards to these devices and their vendors.

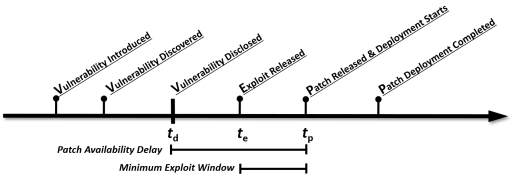


Figure 1: One possible timeline of a vulnerability lifecycle.

Country	Vendor	Ranking	#CVE-IDs[1]	Inclusion
Japan	Buffalo	#1	22	✓
	ELECOM	#2	0	
	IO-DATA	#3	29	✓
	NEC (Aterm)	#4	3	✓
United States	Synergy Digital	#1	0	
	Netgear	#2	56	✓
	Linksys	#3	56	✓
	D-Link	#4	142	✓

Figure 2: Included Vendors and #CVE-IDs.

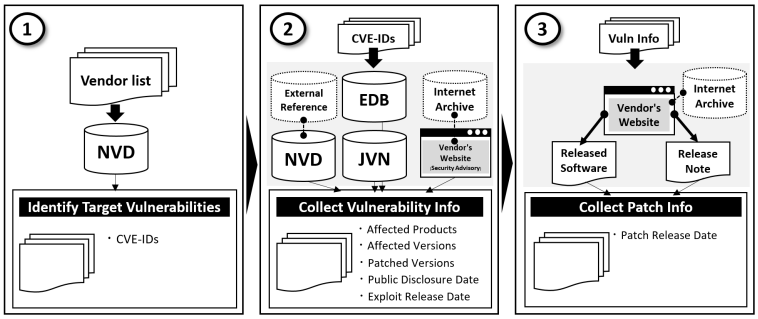


Figure 3: Data Collection Methodology. (1) Identify relevant vulnerabilities related to included vendors—§3.3. (2) Collect vulnerability information—§3.4. (3) Collect patch release information—also §3.4.

As a first step towards filling this void, this paper presents a pilot study on the vulnerability disclosures and patch release behaviors related to prominent consumer IoT vendors. A key novelty in our study is the recognition that, even though we are in a global economy, consumer purchase decisions in different countries are heavily *localized*. Therefore, individual markets may show different trends and risks, and yet these trends and risks may also be correlated due to global trade. To cater for these possibilities, we have thus decided to allocate our effort by markets and analyze them accordingly.

In this paper, we will present our study using data covering 3 prominent consumer IoT vendors in Japan (JP) and 3 in the United States (US) as well as their geographical subsidiaries in Australia (AU), China (CN), and Germany (DE). Following the data collection procedure described in §3, our dataset includes 53 and 230 completed CVE entries [19] for respectively the JP and US markets, covering the 12 years from 2006 to 2017. Aside from characterizing vendor behaviors, our data analysis in §4 also validates the importance of including multiple markets. As an example, we found that 97.9% of the disclosures in our JP dataset are coordinated, but the corresponding US figure is only 55.8%. We will drill into this difference and offer a plausible explanation in §4.3. Finally, our study also uncovered three alarming vendor practices presented in §5: (i) incremental patch release, (ii) unsynchronized patch release, and (iii) implicit end-of-support. Our analysis in an accompanying tech report [20] shows how these practices may pose significant risks of 1-day exploits due to patch-based exploit generation.

2 BACKGROUND

In this section, we will briefly review the roles and the events in vulnerability lifecycles.

Roles. Our paper follows the terminology in [10, §3]. To quote, the *vendor* is "the individual or organization that created or maintains the product that is vulnerable"; the *coordinator* is "an individual or organization that facilitates the coordinated response process"; the *finder* is "the individual or organization that identifies the vulnerability"; and the *deployer* is "the individual or organization that must deploy a patch or take other remediation action".

[1]Note that #CVE-IDs in this table counts all 2006–2017 entries involving a vendor. The corresponding numbers for just consumer IoT devices are shown in Figure 4.

Events. Figure 1 depicts one possible timeline of a vulnerability lifecycle, on which there are 6 events: (i) the vulnerability is introduced by the vendor, (ii) the vulnerability is discovered by a finder, (iii) the vulnerability is publicly disclosed by the finder, the coordinator, or the vendor, (iv) the corresponding exploit is released, (v) the corresponding patch is released, and (vi) the patch reaches complete deployment. Note that while the first two events on any timeline are always (i) and (ii), the remaining four can happen in any order so long as (v) happens before (vi).

In this paper, we denote the *public disclosure date* by t_d, the *patch release date* by t_p, and the *exploit release date* by t_e. The time difference $(t_p - t_d)$ is the *patch availability delay* and ideally it should be small or even negative. Similarly, $(t_p - t_e)$ is the *minimum exploit window*, which measures the time between patch availability and the release of the first exploit *known to us*.

3 DATA COLLECTION METHODOLOGY

A significant portion of our effort was spent in collecting accurate information on the vulnerability disclosures and patch releases related to the included products. In this section, we will detail our data collection process and justify our choices throughout. Our overall data collection methodology is depicted in Figure 3.

3.1 Identify Target Countries

Our study initially *did not* distinguish among countries and our plan was to include as many vendors as our budget allowed by selecting vendors with the most number of consumer IoT products. However, we quickly noticed that the selected vendors were strongly US-centric, which raised questions regarding the applicability of our findings to non-US markets. This gave us the inspiration to conduct a novel study that explores individual markets separately and look for differences. In the end, we settled on the markets of JP and US because: (i) these markets are large and thus important economies, (ii) they have two of the oldest national CSIRTs, namely CERT/CC (est. 1988) and JPCERT/CC (est. 1996), and (iii) our team is highly familiar with these markets. From here on, "our dataset" refers to the combined dataset; we will call out a specific subset when need, e.g., "the JP dataset".

3.2 Identify Target Vendors

Early on in our study, we were already aware that we did not have enough resource to include every consumer IoT vendor in JP and US. Not only are there a large number of them given the broad interpretation of "consumer IoT devices", but we also noticed the crawled data contained a non-trivial number of missing fields that require manual investigations. Therefore, we decided to prioritize prominent vendors and included as many as our budget allowed. To this end, we used the number of wireless routers offered by a vendor to approximate its prominence, then include *every* consumer IoT devices by these vendors. This resulted in a dataset where ~70% of the products are wireless routers and ~30% are in other categories such as network camera and NAS.

To select vendors, first we decided to use Kakaku.com and Amazon.com as representative shopping sites for JP and US respectively. Then, we ranked the vendors using the number of products returned to the query "wireless routers" on the two sites.[2] Figure 2 shows the top 4 vendors in our search conducted in April 2018. Our original plan was to include the top 3 vendors from each site, but we ended up selecting 3 out of the top 4 vendors in both. In JP, the 2nd-ranked vendor was ELECOM. Although ELECOM is a well-known consumer device vendor in JP and has many products, surprisingly no CVE ID was associated with it. In US, the top-ranked vendor was Synergy Digital. However, this vendor also had no associated CVE ID because it mainly sells batteries.

3.3 Identify Target Vulnerabilities

Having fixed the vendors, we used the NVD to identify all CVE entries related to their consumer IoT devices in three steps. First, we retrieved all CVE entries involving the included vendors from the NVD. While this should be straightforward in principle, in practice we have to screen for *orthographical variants* of vendor names. Using the vendor I-O DATA as an example, we were able to discover three different spellings in the NVD: "iodata", "i-o_data", and "i-o_data_device". Second, we excluded non-consumer-IoT entries, which explains the drop in #CVE-IDs between Figures 2 and 4. As an example, we excluded **`CVE-2017-2137`**, which pertains to an access restriction bypass in Netgear ProSAFE Plus Configuration Utility and is not related to a consumer IoT device. Third, we restricted our dataset to CVE entries in the 12 years 2006–2017 because, as we will explain below, we critically depend on the Japan Vulnerability Notes iPedia [14] but this feed dates back to 2006 only.

3.4 Collect Vulnerability & Patch Information

After determining the included CVE entries, we collected the following 6 pieces of information on each vulnerability.

(1) Affected Products and (2) Versions. We started by extracting the affected product names and versions from the NVD. Unfortunately, we encountered a number of empty results when extracting the affected versions, which mirrors the experience reported in [21]. To combat this, we cross-referenced every CVE entry in NVD with its corresponding entry in JVN and extract the missing information from the JVN entry, if available. In addition, we also extracted from vendor security advisories, blog posts, and mailing list archives, either using the current page online or from the Internet Archive [12] if a page was no longer available.

[2]The exact URLs used were, respectively, https://kakaku.com/pc/wireless-router/ and https://www.amazon.com/s/other?k=wireless%20routers&rh=n%3A172282%2Cn%3A541966%2Cn%3A172504%2Cn%3A300189&pickerToList=lbr_brands_browse-bin.

Country	Vendor	#CVE-IDs				# Products	# Patches	# Exploits
		Total	L: <4	M: >=4, <7	H: >=7			
JP	Buffalo	22(20)	1(1)	17(15)	4(4)	71	105	0
	IO-DATA	28(24)	5(3)	13(11)	10(10)	57	88	0
	NEC (Aterm)	3(3)	0(0)	3(3)	0(0)	26	35	0
JP Total		53(47)	6(4)	33(29)	14(14)	154	228	0
US	Netgear	43(25)	1(1)	17(12)	25(12)	107	106	21
	Linksys	52(17)	4(2)	18(3)	30(12)	31	40	12
	D-Link	135(61)	9(6)	74(27)	52(28)	158	177	35
US Total		230(103)	14(9)	109(42)	107(52)	296	323	68
Total		**283(150)**	20(13)	142(71)	121(66)	**450**	**551**	**68**

Figure 4: Dataset summary. The number inside brackets is #CVE-IDs where we were able to find at least one t_p date.

(3) Patched Versions. The procedure to collect patched versions started off similarly the above, i.e., we first extracted from NVD, JVN, and vendor security advisories. In addition, we also extracted from the *release notes* of vendor patches and, in a few cases, resorted to manual investigation using search engines.

(4) Public Disclosure Date. The public disclosure date of a vulnerability, denoted t_d, is the earliest date when the vulnerability was disclosed in a publicly accessible resource. Although the NVD does not contain this information, we were able to extract this information from JVN, thanks to the curation effort of JPCERT/CC and Information-technology Promotion Agency (IPA) [13].

(5) Exploit Release Date. For each included vulnerability, we have also attempted to collect the earliest release date of a publicly-known exploit when one is available in either Exploit Database (EDB) [23] or Metasploit [25]. We denote this date by t_e and it is our best-effort estimation of the exploit release date.

(6) Patch Release Date. When we extracted the patched version of an included vulnerability from a release note, we also attempted to collect the patch release date t_p. To our surprise, some release notes either do not contain a date or, in the case of Buffalo US, contain dates of the source documents from which the notes were translated. When this happened, we resorted to using the first date among the following as a best-effort estimation of t_p: (i) the release date listed by the vendor website, or the Internet Archive if the page no longer exists, (ii) the file date of the release note if it is in an archive, (iii) the file date of the archive, and (iv) the latest of other file dates in the same archive, unpacked using binwalk [5].

4 ANALYSIS OF PATCH RELEASE BEHAVIOR

4.1 Dataset Overview

Figure 4 summarizes our dataset, which contains 283 CVE entries spanning 2006–2017, involving 450 products, 551 patches, and 68 published exploits. Compared with Figure 2, we dropped 15 entries that are unrelated to consumer IoT devices and 10 entries where we were unable to fill in at least one missing field.

To help us understand the severity of the included vulnerabilities, columns 4 to 6 shows the breakdown using Common Vulnerability Scoring System (CVSS) v2. We see that vulnerabilities of medium severity form the majority in both our JP (62%) and US (47%) datasets. In addition, both datasets also contain a significant percentage of high-severity vulnerabilities (26% and 47% respectively).

4.2 Characterizations of Patch Releases

In this section, we will analyze the patch releases and characterize the vendor behaviors in our dataset. Throughout, we will pay special attention in contrasting between the JP and US datasets. In §4.3, we will provide our explanation for several of these differences.

4.2.1 Patch Availability Delay. Figure 6 shows the patch availability delay ($t_p - t_d$) of each vendor as a box-plot. We can classify a vendor based on when it tends to release security patches w.r.t. the corresponding public disclosure dates: (i) *around*: IO-DATA and D-Link; (ii) *before*: Buffalo, NEC (Aterm), and Linksys; and (iii) *after*: Netgear. Of particular note, the median delay for Netgear in our dataset is 23 days and the maximum delay is well over 3 years.

4.2.2 Minimum Exploit Window. Closely related is the minimum exploit window ($t_p - t_e$) of the included vulnerabilities, for which we found 7 exploits that were released *before* their corresponding patches. Although all 7 are in the US dataset, this may be because both exploit databases we used are international and thus may have regional bias. (We are not aware of any JP-focused exploit database.)

4.2.3 Incremental Patch Release. Our dataset shows all 6 vendors practiced *incremental patch release*, which refers to releasing a series of patches to the same vulnerability but for different devices over time. This reflects the common phenomenon that many consumer IoT devices share software components and thus also vulnerabilities. In our dataset, 62.4% of the patches were released incrementally and they are associated with 40 CVE IDs.

Unfortunately, during an incremental patch release, attackers may race to discover the vulnerability in similar devices and exploit it before the vendor releases patches for those other devices. This is effectively a form of 1-day exploitation. To characterize this risk, we have measured the time between the last and the first patch releases ($t_{pL} - t_{pF}$) for each included CVE ID involved in an incremental patch release. Our measurement showed that the average number of days between the first and last patch releases was 122.5 days, with a median of 36 days and a maximum of 1,399 days. This is visualized in Figure 7, which reveals two cases where an exploit was released amid an incremental patch release.

4.2.4 Patch Release Timeliness Over Time. Following [27], we measured the timeliness of patch releases with a breakdown on CVSS severity (L/M/H) over the covered period in order to detect any temporal trend. We categorize the patch release timing into: (i) *before*: $t_p < t_d$, (ii) *concurrent*: $t_p = t_d$, and (iii) *after*: $t_p > t_d$. Figure 5a shows the trend in our dataset. We see that patches released after disclosure (darkest) account for a large portion across all 3 categories, and that there is unfortunately no sign of reduction over time. Interestingly, once we break down the dataset by markets, Figures 5b and 5c show that timings of category (iii) are largely due to the US dataset. This sparked our investigation in §4.3.

4.2.5 Overall Patch Release Timing. Simplifying Figure 5, Figure 9 shows the number of included patches based on their timeliness, followed by breakdowns for JP and US. We see that over 1/2 of the included patches were released pre-disclosure and about 1/3 were released post-disclosure. Again, we see a stark contrast where the overwhelming majority of the included patches in the JP dataset were released pre-disclosure.

4.2.6 Fix Prioritization. We also investigated whether high-severity vulnerabilities were patched quicker, which was addressed in [17] for open-source software. Our dataset shows no discernible prioritization, as demonstrated in Figure 8a, which shows the cumulative distribution functions (CDFs) of the patch availability delay ($t_p - t_d$) for each severity category. Specifically, the CDF for high-severity vulnerabilities remains around 0.9 well into 1 year post-disclosure, while the CDFs for the other two categories have already reached 1.0 at that time. As an extreme example, Netgear released the patch for **CVE-2013-4775** (CVSS 7.8) 1,247 days years post-disclosure.

4.3 Contrasting Between JP and US

In §4.2, we witnessed that the overwhelming majority of the patches in our JP dataset were released either concurrently or before public disclosures, whether in relative proportion (Figure 5) or in absolute number (Figure 9). We therefore hypothesized that the finders in our JP dataset tended to perform coordinated disclosures. To check this hypothesis, we have spent a significant amount of effort to classify every disclosure in our dataset as *Coordinated Disclosures*, *Full Disclosures*, or *Unknown*. (Our procedure is documented in [20].)

Figure 10 depicts our result. Confirming our hypothesis, over 97% of the vulnerabilities in our JP dataset were disclosed via Coordinated Disclosures. In contrast, the corresponding figure is 55.8% for our US dataset, where a staggering 37.5% of the included vulnerabilities were disclosed via Full Disclosures instead.

To explain the above, we observed a characteristic of the finders in our JP dataset. Specifically, while a simple scan of our US dataset shows a diverse set of finders, the finders of 30 of the 53 CVE entries in our JP dataset declared affiliations with a local security company "Mitsui Bussan Secure Directions, Inc." Their declarations led us to believe that they were operating as white-hat hackers, which is a likely explanation to why they performed coordinated disclosures.

5 SIGNIFICANT 1-DAY RISKS UNCOVERED

Our data collection effort has led us to uncover three alarming vendor practices, all of which may pose significant risks of 1-day exploits. Here we will describe these practices and provide highlights of our findings. Our full data analysis can be found in [20].

Incremental Patch Release. This risk behind this practice was already discussed in §4.2.3. While potentially dangerous, it is debatable whether this risk is avoidable because of the inherent conflict between releasing patches as early as possible and delaying patches until the patches for all vulnerable products are ready.

Unsynchronized Patch Release. When researching the included vulnerabilities, we noticed that the regional subsidiaries of some vendors would often release a patch against the same vulnerability on different dates. Since the knowledge of exploit generation from patches is now wide-spread and may even be automated (e.g., [4]), this practice likely represents a significant 1-day risk for customers in regions that receive their patches late. We dub this risk "*geographical arbitrage*"—the potential to generate 1-day exploits by using a patch that was already released in another region.

To characterize this risk, we extended our dataset to include the following subsidiaries of the included vendors: Buffalo US, D-Link DE, D-Link AU, and Netgear CN. Among the patches shared with the US subsidiaries, we found that Buffalo JP leads Buffalo US in

Vendor	Region	#Patches	Average	Median	Maximum
Buffalo	Japan	12	-0.58 days	0.5 days	1 days
D-Link	Germany	103	23.7 days	2 days	366 days
	Australia	62	2.5 days	-1 day	218 days
Netgear	China	51	31 days	8 days	346 days

Table 1: Statistics of patch release gap on patches shared among the US subsidiaries and other included subsidiaries. Positive duration means the US subsidiary lags behind.

50% of the patches, D-Link US is behind D-Link DE in 58.3% of the patches but leads D-Link AU in 59.7% of the patches, and Netgear CN leads Netgear US in 59.6% of the patches. This shows the patch releases by these subsidiaries are indeed often unsynchronized.

Table 1 shows the statistics of the patch release gaps among the shared patches. Alarmingly, for both D-Link and Netgear, we were able to identify a shared patch where the corresponding US subsidiary is over 200 days behind. A more detailed investigation is presented in Table 2, which shows 15 CVE entries and the affected products where an included US subsidiary is at least 30 days behind. Finally, we also identified 5 exploits that were released amid a patch release gap. Although we do not have any evidence to believe any of these exploits was a result of geographical arbitrage, we believe the timing of their releases indisputably demonstrates that geographical arbitrage can pose a real threat.

Implicit End-of-Support (EoS). During our research, we also observed many regional subsidiaries appear to stop releasing patches to products that were still being supported in at least one other region but posted no EoS announcement. This involves 15 patches, which are shown in Table 3. Among the subsidiaries included in this paper, we were not able to discover EoS information from Buffalo US, D-Link AU, Netgear US, and Netgear CN. Unfortunately, this practice of implicit EoS also poses a serious risk to the end users in a similar manner as in unsynchronized patch releases.

6 RELATED WORK

Many previous studies have leveraged information on vulnerabilities and patches to study various aspects of the vulnerability lifecycle and its management. Here we will discuss a few lines of these related works, grouped by the target of the studies.

The first and most-related line of work aims to study the vendor patch release behavior. In their landmark study, Frei et al. [8] have performed a large-scale study using over 80,000 security advisories published between 1995 and 2006 to analyze the temporal relations among vulnerability discoveries, disclosures, and patch releases. A follow-up study to this was published in 2008 [9], which specifically focused on contrasting between Apple and Microsoft. In 2010, a large-scale study leveraging 420 vulnerabilities was published by Arora et al. [2]. However, the vulnerabilities were sampled from the period of 2000-09 to 2003-08, which was ~7 years old at the time of its publication. In 2012, Shahzad et al. [27] published another large-scale study using 46,310 vulnerabilities disclosed between 1988 and 2011. Their focus included 7 major software vendors and 11 of their products. In comparison, our study focused on consumer IoT vendors and our dataset has the natural benefit of being able to cover more recent vulnerabilities. A notable finding of our work is that we do *not* observe any improvement by the included vendors in terms of patch availability delay over the covered period, whereas [27, Fig. 9] shows a marked improvement starting around 2006.

Another line of related work focused on characterizing patches and vulnerabilities. Leveraging the WINE dataset and public data, Bilge and Tudor [3] identified 18 zero-day vulnerabilities, of which 11 were not previously known to have been used in zero-day attacks. Similarly, Nappa et al. [21] characterized patch deployments of 10 popular client software packages and their vulnerabilities due to shared code. In contrast, several previous works relied on public data only. In addition to the aforementioned study by Shahzad et al., we are also aware of [11, 24, 30], which all focused on individual open-source projects. Most recently, Li and Paxson [17] published a large-scale study on 682 open-source projects, characterizing over 4,000 bug fixes for over 3,000 vulnerabilities. Sharing a similarity with our study, their study also revealed a significant information leak. In particular, the authors observed that the patches to many vulnerabilities in open-source projects were publicly visible before disclosure, which poses significant risks of 1-day exploits.

Finally, we note that there is a long line of work on notification, which is related to our study because a natural next step to our study is to investigate the mechanisms and the effectiveness of notifying consumers in regards to vulnerable devices. For this direction, we refer interested readers to five most recent works we are aware of and the references therein: [15, 16, 28, 29, 31].

7 DISCUSSIONS

Data Quality. First, we sincerely thank JVN for providing curated public disclosure dates. Although some previous works used the entry creation date in the NVD as a best-effort estimation (e.g., [27] and [21]), we are aware that these two dates may differ greatly. Thus, we believe future studies should consider using JVN as a more accurate source for public disclosure dates.

Aside from vulnerability databases such as NVD and JVN, patch release notes are arguably the most important information source in vulnerability lifecycle management. Unfortunately, we encountered much irregularity in this source during our study. While some release notes do not contain a date (§3.4), some others do not mention the CVE ID of the vulnerability being fixed even when the patch is for a published CVE ID. This suggests future studies may consider leveraging natural language processing techniques.

Data Size. At present, vendors may silently patch a vulnerability discovered internally. This hinders any study that uses information about patch releases to measure the vendors in their vulnerability management. We believe policy makers may consider requiring vendors to publicly disclose the dates of all discovered vulnerabilities in a way akin to data breach disclosures, which in turn creates incentives for vendors to create products that are more secure.

Vendor Behaviors. Our result shows that vendors who operate regional subsidiaries should consider coordinating among their subsidiaries to synchronize their patch releases and publish End-of-Support information in every region. We believe the former is particularly important because consumers may not be able to simply install a firmware from another region onto their devices due to region-specific technicalities, such as the use of specific wireless frequencies. Although these practices likely require additional investments, we believe they are much-needed improvements—indeed, mainstream software vendors and smartphone vendors have long

adopted these best practices and consumers would (rightly) expect them from every vendor.

Finally, we recommend vendors to consider maintaining permanent machine-readable security feeds, such as release notes with JSON/XML metadata and advisories in RSS. Together with adequate disclosures suggested above, we believe these feeds would create values for participating vendors by, e.g., enabling researchers to create better tools to help the customers of participating vendors to scan and defend their networks after a vulnerability is discovered.

8 CONCLUSION

Using a dataset that covers 283 CVE entries and 450 products, we have studied the vulnerability disclosure and patch release behaviors of 3 prominent consumer IoT device vendors in JP and 3 in US over the 12 years from 2006 to 2017. Overall, we are encouraged to see that 5 out of 6 included vendors have been releasing patches in a timely manner. However, our dataset also reveals several less-desirable behaviors of the included vendors, such as (i) their patch release delays do not show significant improvement over the covered years, and (ii) they do not appear to prioritize patches for vulnerabilities that are more severe.

One interesting investigation enabled by our bi-country dataset is to contrast the two markets. Indeed, we observed notable differences between them in our stratified analyses, which led us to discover different trends in their disclosure coordination. We believe one interesting future study would be to include more markets, which likely requires further automation as we suggested in §7.

Finally, our investigation has also uncovered three alarming vendor practices which may pose significant risks of 1-day exploits. We believe vendors can reduce the risk to their customers by (i) reducing the patch release gaps during incremental patch releases or among regional subsidiaries and (ii) publishing End-of-Support information in every region.

Acknowledgement. We thank Allen Householder for insightful discussions and his suggestion of the term "geographical arbitrage".

REFERENCES

[1] Ashish Arora, Chris Forman, Anand Nandkumar, and Rahul Telang. 2010. Competition and Patching of Security Vulnerabilities: An Empirical Analysis. *Information Economics and Policy* 22, 2 (may 2010), 164–177. https://doi.org/10.1016/j.infoecopol.2009.10.002

[2] Ashish Arora, Ramayya Krishnan, Rahul Telang, and Yubao Yang. 2010. An Empirical Analysis of Software Vendors' Patch Release Behavior: Impact of Vulnerability Disclosure. *Info. Sys. Research* 21, 1 (March 2010), 115–132.

[3] Leyla Bilge and Tudor Dumitras. 2012. Before We Knew It—An Empirical Study of Zero-Day Attacks in the Real World. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*. ACM, 833–844. https://doi.org/10.1145/2382196.2382284

[4] David Brumley, Pongsin Poosankam, Dawn Song, and Jiang Zheng. 2008. Automatic Patch-Based Exploit Generation is Possible: Techniques and Implications. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy*. IEEE Computer Society, 143–157. https://doi.org/10.1109/SP.2008.17

[5] devttys0. [n.d.]. binwalk. https://github.com/ReFirmLabs/binwalk.

[6] Zakir Durumeric, James Kasten, David Adrian, J. Alex Halderman, Michael Bailey, Frank Li, Nicolas Weaver, Johanna Amann, Jethro Beekman, Mathias Payer, and Vern Paxson. 2014. The Matter of Heartbleed. In *Proceedings of the 2014 Conference on Internet Measurement Conference*. ACM, 475–488. https://doi.org/10.1145/2663716.2663755

[7] Stefan Frei. 2009. *Security Econometrics—The Dynamics of (In)Security*. Ph.D. Dissertation. ETH Zurich. https://doi.org/10.3929/ethz-a-005887804

[8] Stefan Frei, Martin May, Ulrich Fiedler, and Bernhard Plattner. 2006. Large-scale Vulnerability Analysis. In *Proceedings of the 2006 SIGCOMM Workshop on Large-Scale Attack Defense*. ACM Press, 131–138.

[9] Stefan Frei, Bernhard Tellenbach, and Bernhard Plattner. 2008. 0-Day Patch Exposing Vendors (In)security Performance. In *Black Hat Europe 08*. Black Hat, 1–15. https://www.blackhat.com/presentations/bh-europe-08/Frei/Whitepaper/bh-eu-08-frei-WP.pdf

[10] Allen D. Householder, Garret Wassermann, Art Manion, and Chris King. [n.d.]. The CERT Guide to Coordinated Vulnerability Disclosure. https://resources.sei.cmu.edu/asset_files/SpecialReport/2017_003_001_503340.pdf.

[11] Zhen Huang, Mariana DAngelo, Dhaval Miyani, and David Lie. 2016. Talos: Neutralizing Vulnerabilities with Security Workarounds for Rapid Response. In *Proceedings of the 2016 IEEE Symposium on Security and Privacy*. IEEE, 618–635. https://doi.org/10.1109/SP.2016.43

[12] Internet Archive. [n.d.]. Internet Archive. https://archive.org/.

[13] JPCERT Coordination Center and Information-technology Promotion Agency. [n.d.]. How to Use JVN iPedia. https://jvndb.jvn.jp/en/nav/jvndbhelp.html.

[14] JPCERT Coordination Center and IPA Information-technology Promotion Agency. [n.d.]. JVN: JVN iPedia. https://jvndb.jvn.jp/.

[15] Frank Li, Michael Bailey, Zakir Durumeric, Jakub Czyz, Mohammad Karami, Damon Mccoy, Stefan Savage, Michael Bailey, Damon Mccoy, Stefan Savage, and Vern Paxson. 2016. You've Got Vulnerability: Exploring Effective Vulnerability Notifications. In *Proceedings of the 25th USENIX Security Symposium*. USENIX Association, 1033–1050. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/li

[16] Frank Li, Grant Ho, Eric Kuan, Yuan Niu, Lucas Ballard, Kurt Thomas, Elie Bursztein, and Vern Paxson. 2016. Remedying Web Hijacking. In *Proceedings of the 25th International Conference on World Wide Web*. ACM Press, 1009–1019. https://doi.org/10.1145/2872427.2883039

[17] Frank Li and Vern Paxson. 2017. A Large-Scale Empirical Study of Security Patches. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM Press, 2201–2215.

[18] Ministry of Internal Affairs and Communications, the Government of Japan. [n.d.]. Conducting Survey on Vulnerable IoT Devices. JP: http://www.soumu.go.jp/menu_news/s-news/02ryutsu03_04000088.html, EN: http://www.soumu.go.jp/main_sosiki/joho_tsusin/eng/Releases/Telecommunications/170905_1.html.

[19] MITRE Corporation. [n.d.]. CVE: Common Vulnerabilities and Exposures. https://cve.mitre.org/.

[20] Asuka Nakajima, Takuya Watanabe, Eitaro Shioji, Mitsuaki Akiyama, and Maverick Woo. 2019. *A Pilot Study on Consumer IoT Device Vulnerability Disclosure and Patch Release in Japan and the United States*. Technical Report CMU-CyLab-19-001. CyLab, Carnegie Mellon University. https://www.cylab.cmu.edu/_files/pdfs/tech_reports/CMUCyLab19001.pdf

[21] Antonio Nappa, Richard Johnson, Leyla Bilge, Juan Caballero, and Tudor Dumitraş. 2015. The Attack of the Clones: A Study of the Impact of Shared Code on Vulnerability Patching. In *Proceedings of the 2015 IEEE Symposium on Security and Privacy*. IEEE Computer Society, 692–708.

[22] National Telecommunications and Information Administration. [n.d.]. Multistakeholder Process—Internet of Things (IoT) Security Upgradability and Patching. https://www.ntia.doc.gov/other-publication/2016/multistakeholder-process-iot-security.

[23] Offensive Security. [n.d.]. Exploit-DB. https://www.exploit-db.com/.

[24] Andy Ozment and Stuart E. Schechter. 2006. Milk or Wine: Does Software Security Improve with Age?. In *Proceedings of the 15th USENIX Security Symposium*. USENIX Association, 93–104. https://www.usenix.org/legacy/event/sec06/tech/ozment.html

[25] Rapid 7. [n.d.]. Metasploit Framework. https://www.metasploit.com/.

[26] Guido Schryen. 2009. A Comprehensive and Comparative Analysis of the Patching Behavior of Open Source and Closed Source Software Vendors. In *Proceedings of the Fifth International Conference on IT Security Incident Management and IT Forensics*. IEEE, 153–168. https://doi.org/10.1109/IMF.2009.15

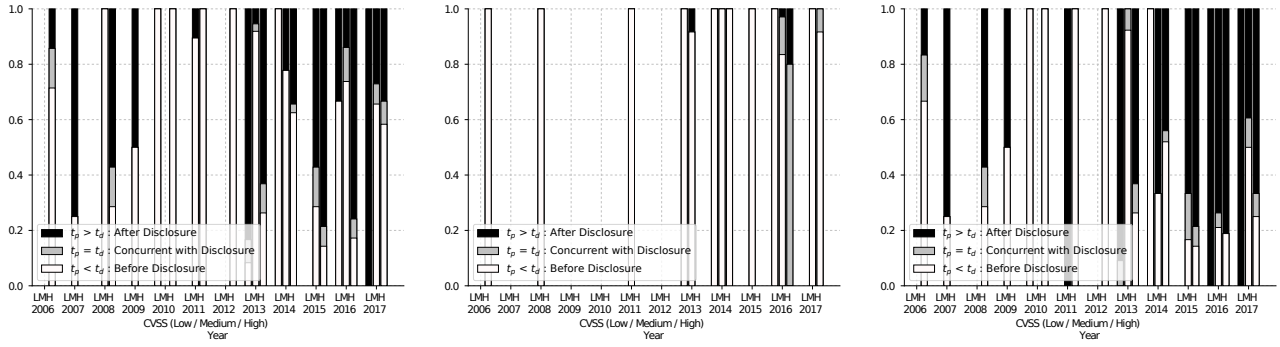
[27] Muhammad Shahzad, Muhammad Zubair Shafiq, and Alex X. Liu. 2012. A Large Scale Exploratory Analysis of Software Vulnerability Life Cycles. In *Proceedings of the 34th International Conference on Software Engineering*. IEEE Press, 771–781.

[28] Ben Stock, Giancarlo Pellegrino, Frank Li, Michael Backes, and Christian Rossow. 2018. Didn't You Hear Me?—Towards More Successful Web Vulnerability Notifications. In *Proceedings of the 2018 Network and Distributed System Security Symposium*. Internet Society. https://doi.org/10.14722/ndss.2018.23171

[29] Ben Stock, Giancarlo Pellegrino, Christian Rossow, Martin Johns, and Michael Backes. 2016. Hey, You Have a Problem: On the Feasibility of Large-Scale Web Vulnerability Notification. In *Proceedings of the 25th USENIX Security Symposium*. USENIX Association, 1015–1032. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/stock

[30] Shahed Zaman, Bram Adams, and Ahmed E. Hassan. 2011. Security Versus Performance Bugs: A Case Study on Firefox. In *Proceeding of the 8th Working Conference on Mining Software Repositories*. ACM Press, 93–102. https://doi.org/10.1145/1985441.1985457

[31] Jia Zhang, Haixin Duan, Wu Liu, and Xingkun Yao. 2017. How to Notify a Vulnerability to the Right Person? Case Study: In an ISP Scope. In *Proceedings of the 2017 IEEE Global Communications Conference*. IEEE, 1–7. https://doi.org/10.1109/GLOCOM.2017.8253993



(a) All (JP + US) (b) Japan (c) United States

Figure 5: Timeliness of patch release by all included vendors over 2006–2017, partitioned by vulnerability severity. A darker color corresponds to a worse event.

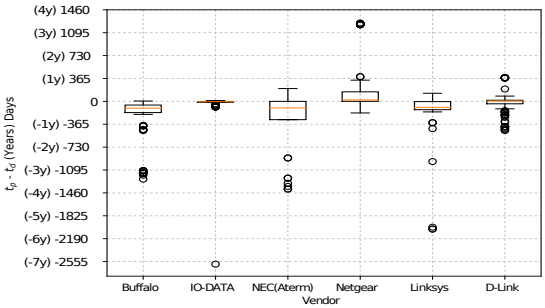


Figure 6: Box-plots of patch availability delay ($t_p - t_d$) for each included vendor.

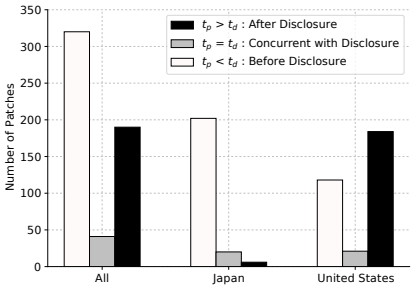


Figure 9: Number of patch released before, concurrent with, and after disclosure, along with breakdowns of JP and US.

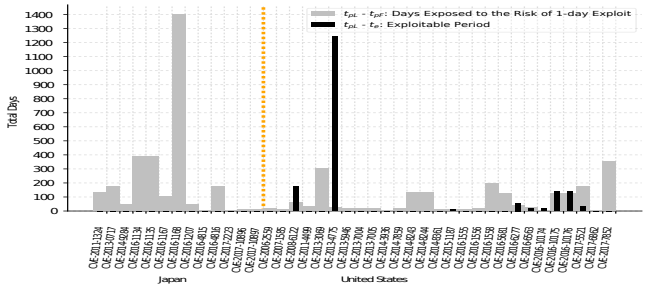


Figure 7: Number of days where a device was exposed to risk due to incremental patch release. This risk materialized in two cases: `CVE-2016-6563` and `CVE-2017-5521`.

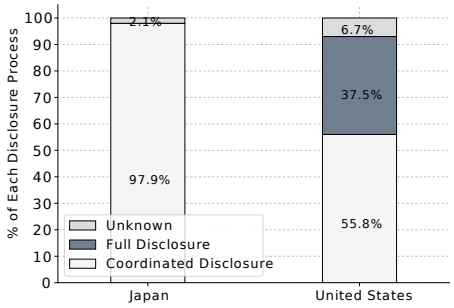
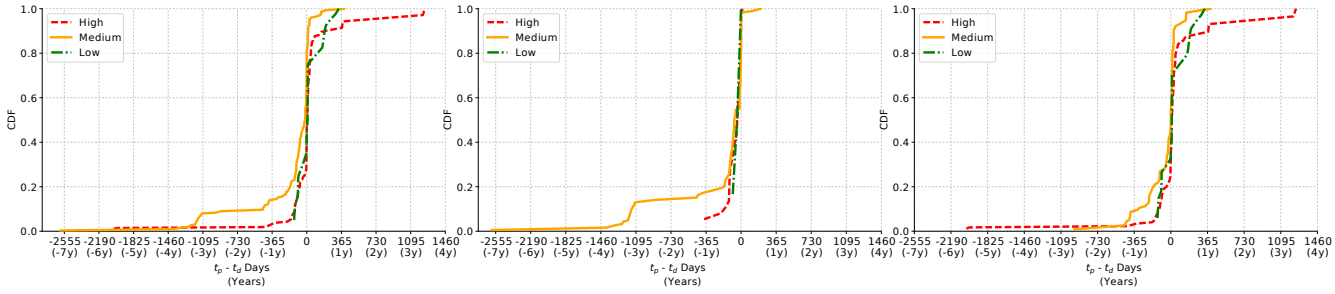


Figure 10: Category percentages of different vulnerability disclosure processes in our JP and US datasets.



(a) All (b) Japan (c) United States

Figure 8: CDFs of patch availability delay (t_p - t_d) for each severity level of vulnerability.

Vendor	CVE ID	Public Disclosure Date (t_d)	Affected Product	Patch Release Date (t_p)			$t_p - t_d$		Exploit Release Date (t_e)	$t_p - t_e$	
				US	DE	Δ	US	DE		US	DE
D-Link	CVE-2017-7852	2017-02-22	DCS-2132L RevA	2016-11-02	2016-07-21	104	112	216	2017-02-22	112	216
			DCS-2136L RevA	2016-11-04	2016-07-21	106	110	216		110	216
			DCS-2310L RevA	2016-11-04	2016-07-21	106	110	216		110	216
			DCS-2332L RevA	2016-11-04	2016-07-21	106	110	216		110	216
			DCS-5009L RevA	2016-01-05	2015-11-18	48	414	462		414	462
			DCS-5222L RevB	2016-01-14	2015-12-11	34	405	439		405	439
			DCS-6010L RevA	2016-11-04	2016-07-21	106	110	216		110	216
			DCS-932L RevA	2016-07-19	2015-11-18	244	218	462		218	462
			DCS-933L RevA	2016-01-18	2015-11-18	61	401	462		401	462
	CVE-2016-6563	2016-11-07	DIR-890L RevA	2016-11-09	2016-09-07	63	-2	61	2016-11-21	12	75
			DIR-868L RevA	2016-12-29	2016-09-22	98	-52	46		-38	60
			DIR-869 RevA	2016-12-07	2016-06-22	168	-30	138		-16	152
	CVE-2016-5681	2016-08-11	DIR-868L RevB1	2016-04-29	2015-04-29	366	104	470	-	-	-
			DIR-868L RevC1	2016-07-01	2015-07-01	366	41	407			
	CVE-2016-1558	2016-03-16	DAP-3662 RevA	2016-09-29	2017-01-26	-119	-197	-316	-	-	-
	CVE-2015-1187	2015-03-02	DIR-626L RevA	2015-03-09	2015-04-16	-38	-7	-45	2015-02-26	-11	-49
	CVE-2014-100005	2014-03-07	DIR-600 RevB	2014-03-17	2013-06-25	265	-10	255	-	-	-
	CVE-2013-6027	2013-10-17	DIR-100	2013-12-02	2013-11-01	31	-46	-15	2013-10-14	-49	-18
	CVE-2006-6055	2006-11-21	DWL-G132	2006-11-16	2006-07-07	132	5	137	2006-11-13	-3	129
				US	AU	Δ	US	AU		US	AU
	CVE-2016-6563	2016-11-07	DIR-895L	2016-11-10	2016-09-22	49	-3	46	2016-11-21	11	60
	CVE-2016-5681	2016-08-11	DIR-850L RevB1	2016-05-17	2016-06-21	-35	86	51	-	-	-
			DIR-868L RevB1	2016-04-29	2015-09-24	218	104	322			
				US	CN	Δ	US	CN		US	CN
Netgear	CVE-2017-5521	2017-01-16	R6300v2	2016-09-07	2016-07-04	65	131	196	2017-01-30	145	210
	CVE-2013-3069	2013-07-16	R6200	2013-12-26	2013-01-14	346	-163	183	-	-	-
			WNDR3400v3	2014-05-13	2013-08-07	279	-301	-22			
			WNR1000v3	2014-06-19	2014-02-26	113	-338	-225			
			WNDR3700v2	2014-01-20	2013-12-01	50	-188	-138			
			WNDR37AVv2	2014-01-20	2013-12-01	50	-188	-138			
			WNDR4500v2	2014-02-03	2013-11-21	74	-202	-128			
	CVE-2011-1674	2011-04-09	WNAP210v1	2012-05-06	2011-11-30	158	-393	-235	-	-	-
	CVE-2011-1673	2011-04-09	WNAP210v1	2012-05-06	2011-11-30	158	-393	-235	-	-	-
	CVE-2006-6125	2004-01-30	WG311v1	2004-01-30	2003-12-19	42	0	42	2006-11-22	1027	1069

Table 2: Vulnerabilities and corresponding products where the patch release date by the US subsidiary is at least 30 days behind.

CVE ID	Affected Product	Patched Version (Global)		Latest Version at Subsidiary Shown	
		version	release date	version	release date
Buffalo US					
CVE-2016-4816	WHR-300HP	1.98	2016-01-25	1.93	2013-03-07
D-Link Australia					
CVE-2017-7852	DCS-2310L RevA	1.08.03	2016-11-04	1.07.00	2015-10-01
	DCS-2332L RevA	1.08.03	2016-11-04	1.07.00	2015-10-01
	DCS-6010L RevA	1.15.03	2016-11-04	1.14.00	2015-10-01
CVE-2016-6563	DIR-880L	1.08b04	2016-11-10	1.07b08	2016-03-20
	DIR-868L RevB1	2.05b02	2016-12-07	2.03b01	2015-09-24
CVE-2015-2052	DIR-645 RevA	1.05b01	2015-04-24	1.03b11	2012-10-12
CVE-2015-2051	DIR-645 RevA	1.05b01	2015-04-24	1.03b11	2012-10-12
CVE-2014-100005	DIR-600 RevB	2.17b01	2014-03-17	2.15b02	2013-03-11
CVE-2014-9518	DIR-655 RevB	2.12b01	2014-11-01	2.05b06	2012-01-17
CVE-2013-7389	DIR-645 RevA1	1.04b11	2013-12-19	1.03b11	2012-10-12
Netgear China					
CVE-2017-6862	WNR2000v4	1.0.0.66	2017-01-17	1.0.0.44	2015-03-20
CVE-2017-5521	R6700	1.0.1.16	2017-01-16	1.0.0.26	2016-03-31
	WNDR3400v2	1.0.0.54	2017-01-19	1.0.0.48	2013-06-13
CVE-2016-10176	WNR2000v4	1.0.0.66	2017-01-17	1.0.0.44	2015-03-20
CVE-2016-10175	WNR2000v4	1.0.0.66	2017-01-17	1.0.0.44	2015-03-20
CVE-2016-10174	WNR2000v4	1.0.0.66	2017-01-17	1.0.0.44	2015-03-20
CVE-2016-10106	FVS336Gv3	4.3.3-8	2016-05-27	4.3.3-6	2015-10-29
	FVS318Gv2	4.3.3-8	2016-05-27	4.3.3-6	2015-10-29
	SRX5308	4.3.3-8	2016-05-27	4.3.3-6	2015-10-29
CVE-2016-6277	R6700	1.0.1.16	2017-01-16	1.0.0.26	2016-03-31
CVE-2016-1556	WN604	3.3.3	2016-03-03	3.0.2	2012-12-19
CVE-2016-1555	WN604	3.3.3	2016-03-03	3.0.2	2012-12-19
CVE-2013-4775	GS724Tv3	5.4.2.27	2017-01-09	5.4.2.19	2015-06-25
	GS716Tv2	5.4.2.27	2017-01-09	5.4.2.19	2015-06-25
	GS108Tv2	5.4.2.27	2016-12-26	5.4.2.19	2015-07-01
	GS110TP	5.4.2.27	2017-01-09	5.4.2.19	2015-07-01
	GS510TP	5.4.2.27	2017-01-09	5.4.2.19	2015-07-01
	GS752TPS	5.3.0.29	2016-12-26	5.3.0.26	2015-01-20
	GS728TPS	5.3.0.29	2016-12-26	5.3.0.26	2015-01-20
	GS728TS	5.3.0.29	2016-12-26	5.3.0.26	2015-01-20
	GS752TS	5.3.0.29	2016-12-26	5.3.0.26	2015-01-20

Table 3: Vulnerabilities whose patches are not provided by a regional subsidiary, which we believe is likely due to implicit End-of-Support.