Firmware Update Trend in the Internet of Things  
-An Empirical Survey of Japanese HGW Vendors-

Ichiro Mizukoshi,  
Institute of Information Security, Japan  
2-14-1 Tsuruya-cho, Kanagawa-ku, Yokohama-city, Kanagawa, 221-0835, Japan  
mizukoshiichiro@gmail.com

Aki Nakanishi,  
Meiji University, Japan  
1-1 Kanda-Surugadai, Chiyoda-Ku, Tokyo, 101-8301 Japan  
aki.nakanishi@nifty.com

Atsuhiro Goto,  
Institute of Information Security, Japan  
2-14-1 Tsuruya-cho, Kanagawa-ku, Yokohama-city, Kanagawa, 221-0835, Japan  
goto@iisec.ad.jp

ABSTRACT
Firmware vulnerability is a serious concern in the Internet of Things (IoT) environment. Home Gateway (HGW) is a small router that connects the home network to the Internet. Malicious hackers attack HGW utilizing its vulnerability. HGW can be considered as a standard example of IoT as it is always connected to the Internet and many HGWs exit. In Japan, there are more than 40 million units. In this paper, we first report the current situation regarding vulnerability management by the HGW vendors in Japan. There are two types of business models. The first one is called “SELL,” and the other is “SUBSCRIPTION” (hereafter SUB). The SELL model is simple. The vendor sells HGWs to the end user and the vendor cannot access the HGWs without the end user’s permission. The SUB model is slightly complicated. HGWs are leased to the end user from the vendor and the vendor also acts as a service operator providing Internet connectivity to the user. The vendor is required to maintain the functioning of the HGWs. Next, we describe our findings as follows:

1) Aggressiveness with regard to the security update varies for each vendor. 2) SELL vendors have an average of 4.5 times of updates during the lifetime and the final update is provided 46.7 days before the end of sales. 3) SUB has 16.45 times of updates and the final update is provided 1069.7 days after the end of sales.

Finally, we discuss some issues as follows: The devices that are not updated against vulnerability become dangerous debris and the mass of debris will become a serious risk. There are several ways to regulate them. Based on Lessig’s code, we classified them into four categories: Law, Norms, Market, and Architecture. Our classification is as follows: Law - Product Liability act., Norms - Open source, Market – Subscription, and Architecture - programed to die.

KEYWORDS
IoT (the Internet of Things), security, vulnerability, end-users, vendors

1. INTRODUCTION
Although the definition of Internet of Things (IoT) remains ambiguous [1], we can assume
that Home Gateway (HGW) is an example of IoT as a large number of such devices are connected to the Internet at all times. HGW is a small router that connects the home network to the Internet and has the following features:

- HGWs are always connected to the Internet making them a popular target for hackers.
- Sometimes it is necessary to take measures for protocol vulnerability, which is not because of the product itself.
- Users are rarely aware of the presence of HGWs because they are not handled directly by users.

In Japan, there are more than 40 million units of HGWs.

The basic function of an HGW is substantially common among vendors and hence it is meaningful to compare these produce as a group. Moreover, security incidents related to HGW vulnerability are so serious and dangerous in the real world that it is important to research this topic. For example, in 2011, targeting the vulnerability of one firmware, millions of Brazilian HGWs were under attack [2]. In late 2013, Polish HGWs were attacked for the purpose of financial theft [3]. In 2014, self-replicating malware attacked popular American HGWs [4]. A hacking contest was held to test the security of HGWs at a security conference [5]. Moreover, since 2013, more than thousands of IDs and passwords have been stolen from vulnerable HGWs in Japan [6]. In 2015, the Japanese Metropolitan Police Department sent vulnerability alerts for specific HGWs [7].

To fix the vulnerability, firmware update is necessary, which the vendors usually provide. Tripwire reports that, “74 percent of Amazon’s top 50 best-selling SOHO wireless router models have security vulnerabilities. In addition, 34 percent of Amazon’s top 50 selling models have publicly documented exploits available, making it relatively simple for attackers to use this information to craft targeted attacks or simply attack all the vulnerable systems they can find.” [8].

In this paper, first we report the current situation of vulnerability remediation by the HGW vendors in Japan. For this purpose, there are two types of business models. The first one is called “SELL,” and the other is called “SUBSCRIPTION” (hereafter SUB). The SELL model is simple. Here, vendors sell HGWs to the end users; therefore, vendors cannot access the HGWs without the end users’ permission. The SUB is slightly complicated. HGWs are leased to the end users by the vendors. In this case, vendors can also act as service operators and are required to maintain the functioning of the HGWs to provide Internet connectivity to the users. We obtained the firmware update information from both types of vendor support web pages in Japan. We collected the data for 330 products from 5 vendors along with the 1837 updates and analyzed them.

Next, we describe our findings as follows: 1) The update behavior and styles varies greatly for each vendor. 2) SELL has an average of 4.5 times of update and the final update is provided 46.7 days before the end of sales. 3) On the other hand, SUB has 16.45 times of update and final update is 1069.7 days after the end of sales. This is considered to be due to the difference between both the business models.

Finally, we discuss the following issues: The devices that are not updated against vulnerability are dangerous debris and this mass of debris will become serious risk. There are several methods to avoid or mitigate these issues. Based on Lessig’s theory in his book “Code,” we classified them into four categories: Law, Norms, Market, and Architecture. Our classification is as follows: 1) Law - product liability act, 2) Norms - open source, 3) Market – subscription and 4) Architecture - programmed to die.

2. LITERATURE REVIEW

Regarding the security of IoT, FTC (Federal Trade Commission) states that, “companies should continue to monitor products...
throughout the life cycle and, to the extent feasible, patch known vulnerabilities” [9]. However, it is difficult to obtain real data of these activities. In this paper, we researched the update cycle using field data.

Rahul et al. researched on vulnerability disclosure and found that there is a relationship between vulnerability disclosure and vendor share prices. The factors that have an adverse effect on the stock price are the seriousness of the vulnerability, intensity of the competitive environment, size of the vendor’s scale, and the delay in the release of the updated firmware [10]. A delay in response is observed when there is no vulnerability disclosure by CERT/CC, etc. [11]. They focus on the relationship between the vendor’s profile and the vulnerability disclosure. However, it is important to investigate the vendor behavior in more detail. Therefore, in this paper, we analyze the vendor update behavior for each product.

Gear warned about the risk when devices are no longer updated. He showed two options to avoid the risk [12] [13]. First, IoT devices must be programmed in advance to reach their end-of-life at a certain time.

Second, when products reach their end-of-life, its vendor should release it as open-source.

Lessig argues that there are four major regulators in cyberspace, namely, **Law, Norms, Market, and Architecture** [14]. Paul summarizes them as follows [15].

**Law**—explicit mandates that can be enforced by the government

**Norms**—social conventions that one is often compelled to follow

**Market**—economic forces

**Architecture**—the physical or technical constraints on activities

### 3. FRAMEWORK AND DEFINITIONS

#### 3.1 Firmware vendor update behavior

We focus on the vendor update behavior for each product. There are three reasons for updating the firmware after the release.

- Adding a new feature
- Fixing a bug
- Vulnerability remediation

The detailed information about the update is provided in the vendor release notes. Important changes are described in these notes; however, minor ones may be omitted. Though feature additions and bug fixes are reported relatively in detail, vulnerability remediation should not be reported as it would provide the necessary information to attackers. Therefore, it is difficult to observe vulnerability remediation accurately from the outside. However, the update includes it. Using the update information published by the vendors, we can infer the status of vulnerability remediation.

We focus on two parameters.

- **Frequency**: How many times an update has been released?
- **Duration**: How long period update has been released?

#### 3.2 Vendor decision making and device life cycle

Even when bugs and vulnerabilities are found in the device firmware, firmware updates are not necessarily provided. Vendors decide on whether to provide updates considering the following factors:

- **External**
  - damage to the users, loss of the reputation of the company, etc.

- **Internal**
  - development cost, ensuring development resources, etc.

- **Constraints**
  - size of the free memory in devices, etc.

Some bug fixes and vulnerability remediation are developed all at once and some others are postponed. The decision regarding an update is taken by the vendors and the users have no option in this matter.

The life cycle of the devices is as follows.

**Sales Start -> End of Sales -> Repair Exit**

It is possible to continue the use of devices after the repair exit. As long as the users...
continue to use the devices, vulnerability remediation is required. However, it is up to the vendors whether they want to continue with vulnerability remediation and the users do not have an option. Figure 1 shows the relationship between the life cycle and the update for a device.

Fig. 1 Life cycle and update

3.3 Business models of vendors
We develop two business models for providing HGWs to the end users. One is the SELL model, which is simple. Vendors sell HGWs to end users. The other model is called SUB. In SUB, HGWs are leased to the end users from the vendors. In addition, the vendors also function as service operators providing Internet connectivity to the users.

SELL: HGWs are sold by the vendors. Users buy them.
SUB: HGWs are leased by the vendors. Users use them.

3.4 Parameters
We target the HGWs for which the sales have already ended. We check their support pages on the vendor web site. We choose JAN (Japanese Article Number) as the product identifier because each JAN has mostly unique support pages. JAN is a barcode standard that is compatible with International Article Number schemes such as EAN and UPC. For this reason, HGWs and peripherals including Wi-Fi USB adapters are uniquely identified. Therefore, HGWs are treated as separate products. Moreover, when the color of the body is different, JAN is also different. However, in this study there was no variation of color.

We have collected the following four parameters for each product.

- **SalesStart**: date on which the sales started
- **SalesEnd**: date on which the sales ended
- **UpdateCount**: total number of updates
- **FinalUpdate**: date of the final update

In case an update is not provided after the start of sale, **UpdateCount** is 0 and **FinalUpdate** is the sale start date. When the dates are reported in the early, mid, or later part of the month, we replace each of them with 1st, 10th, and 20th.

We count the number of former firmware versions on the support page to update the value of **UpdateCount**. We also considered citing the number based on the release notes; however, we did not do so in view of the presence of versions that exist only in the release notes. These versions might be the internal releases by the vendors, which the end users cannot download.

We define some parameters as follows:

- **SalesPeriod**
  Since how long have the products been sold?
  \[ \text{SalesPeriod} = \text{SalesStart} - \text{SalesEnd} \]

- **UpdatePeriod**
  Since how long have updates been released?
  \[ \text{UpdatePeriod} = \text{FinalUpdate} - \text{SalesStart} \]

- **UpdatePeriodAfterEndOfSales**
  For how long have updates been released after the end of sales?
  \[ \text{UpdatePeriodAfterEndOfSales} = \text{FinalUpdate} - \text{SalesEnd} \]

4. HYPOTHESIS

4.1 Conditions
Conditions that determine the development of the update (External, Internal, Constraints) will be different for each vendor. This leads to the following hypothesis:

H.1 **UpdateCount** and **UpdatePeriod** are different for each vendor.
4.2 Sales Period
The purpose of the SELL model is to sell the products. SELL is active in introducing new products to stimulate replacement demands. In contrast, in the SUB model, HGWs are used for providing the functions. In this model, the introduction of new products is not beneficial due to the incurred replacement cost. This discussion leads to the following hypothesis:

H.2
SalesPeriod(SELL) ⊈ SalesPeriod(SUB)

4.3 Update Frequency
Updates prolong the life of the devices; however, the development cost is required to be incurred. Updates for products are mostly provided free of charge. In the Sell model, updates are not only costly but also may reduce future sales. On the other hand, in the SUB model, funds may be obtained from the continued contracts. If prolonging through updates is not possible, the SUB model should provide a successor for service continuity, which requires more cost (developing a new product, shipping, installing, etc.) than providing an update. Hence, update is preferred in the SUB model.

From the above, we develop the hypothesis that SUB is has advantages with updates while SELL does not.

H.3a
UpdateCount(SELL) << UpdateCount(SUB)

H.3b
UpdatePeriod(SELL) << UpdatePeriod(SUB)

4.4 Update Period after the End of Sales
For the same reason, as in the previous section, SUB has advantages with providing updates after the end of sales, while SELL does not. This leads to the following hypothesis:

H.4
UpdatePeriodAfterEndOfSales(SELL)
  ⊈ UpdatePeriodAfterEndOfSales(SUB)

5. RESEARCH METHOD
5.1 Data Set
We analyzed five vendors providing 330 products and 1837 updates as follows. Vendor E provides more than 6 million units of HGWs in Eastern Japan, and its sister company develops a similar service in Western Japan. The total number of units provided by them is 10 million as of October 2015. The other four vendors are major players in the Japanese HGW market. We chose vendors listed in the top-selling list from the Japanese e-commerce web site. This data is considered to substantially cover the entire HGW market in Japan, except for CABLE TV. HGWs for CABLE TV are called STBs (Set Top Boxes). We could not find any update information on STBs from the CABLE TVs’ home page. Therefore, our research on STB will be a part of future work.

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Type</th>
<th>Products</th>
<th>Updates</th>
<th>SalesStart</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>SELL</td>
<td>73</td>
<td>263</td>
<td>2002/1/31 – 2014/9/15</td>
</tr>
<tr>
<td>L</td>
<td>SELL</td>
<td>30</td>
<td>65</td>
<td>2009/8/1 – 2013/5/1</td>
</tr>
<tr>
<td>N</td>
<td>SELL</td>
<td>77</td>
<td>447</td>
<td>2000/10/13 – 2009/10/5</td>
</tr>
<tr>
<td>B</td>
<td>SELL</td>
<td>121</td>
<td>585</td>
<td>2002/10/22 – 2012/3/20</td>
</tr>
<tr>
<td>E</td>
<td>SUB</td>
<td>29</td>
<td>477</td>
<td>2004/9/1 – 2011/3/1</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>330</td>
<td>1837</td>
<td></td>
</tr>
</tbody>
</table>

All surveys were conducted in May 2015, except for vendor B, which was conducted in October 2015. Vendor N does not report its end of sale date on its support page. From our interview, we defined the sale end date to be five years prior to the repair exit date on the support page. Vendor B does not report SalesStart and SalesEnd in its support page.
Therefore, we used the date registered in the JAN database.

An overview of the data set is shown below.

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Average</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>718.7</td>
<td>505.2</td>
</tr>
<tr>
<td>L</td>
<td>750.7</td>
<td>335.2</td>
</tr>
<tr>
<td>N</td>
<td>1055.0</td>
<td>268.3</td>
</tr>
<tr>
<td>B</td>
<td>605.0</td>
<td>344.1</td>
</tr>
<tr>
<td>E</td>
<td>927.1</td>
<td>274.7</td>
</tr>
<tr>
<td>Total</td>
<td>776.7</td>
<td>404.7</td>
</tr>
</tbody>
</table>

Fig. 2 Boxplot for SalesPeriod for all five vendors

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Average</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>3.60</td>
<td>1.99</td>
</tr>
<tr>
<td>L</td>
<td>2.17</td>
<td>2.79</td>
</tr>
<tr>
<td>N</td>
<td>5.81</td>
<td>3.98</td>
</tr>
<tr>
<td>B</td>
<td>4.83</td>
<td>3.63</td>
</tr>
<tr>
<td>E</td>
<td>16.45</td>
<td>4.94</td>
</tr>
<tr>
<td>Total</td>
<td>5.57</td>
<td>4.96</td>
</tr>
</tbody>
</table>

Fig. 3 Boxplot for UpdateCount for all five vendors

Table 4 UpdatePeriod: **FinalUpdate- SalesStart**

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Average</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>552.9</td>
<td>432.0</td>
</tr>
<tr>
<td>L</td>
<td>564.2</td>
<td>596.1</td>
</tr>
<tr>
<td>N</td>
<td>816.3</td>
<td>597.7</td>
</tr>
<tr>
<td>B</td>
<td>787.1</td>
<td>373.2</td>
</tr>
<tr>
<td>E</td>
<td>1996.7</td>
<td>578.1</td>
</tr>
<tr>
<td>Total</td>
<td>828.2</td>
<td>614.9</td>
</tr>
</tbody>
</table>

Fig. 4 Boxplot for UpdatePeriod for all five vendors

Table 5 UpdatePeriodAfterEndOfSales: **FinalUpdate-SalesEnd**

Negative value indicates that there has been no update after the End of Sales

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Average</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>-165.7</td>
<td>451.6</td>
</tr>
<tr>
<td>L</td>
<td>-186.6</td>
<td>504.2</td>
</tr>
<tr>
<td>N</td>
<td>-238.8</td>
<td>661.1</td>
</tr>
<tr>
<td>B</td>
<td>182.1</td>
<td>530.6</td>
</tr>
<tr>
<td>E</td>
<td>1069.7</td>
<td>737.7</td>
</tr>
<tr>
<td>Total</td>
<td>51.5</td>
<td>670.8</td>
</tr>
</tbody>
</table>
6. ANALYSIS

6.1 Conditions
There was a statistically significant difference between the number of updates and the period of the update for different vendors with a mean rank of 73 for products by I, 30 for products by L, 77 for products by N, 121 for products by B, and 29 for products by E.

UpdateCount (H=101.96, P < 2.2e-16)
UpdatePeriod (H=89.288, P < 2.2e-16)

We rejected the null hypothesis that all vendors have the same distribution. Therefore, H.1 invalidated.

H.1 UpdateCount and UpdatePeriod are different for each vendor.

6.2 SalesPeriod
We compare the data for the SELL and SUB models and perform the Shapilo-Wilk test for normality.

Both SELL and SUB models do not meet the normality. Then, we conducted the Wilcoxon rank sum test. It indicated that the SalesPeriod was significantly longer for SUB (Mdn = 1095.0) than for SELL (Mdn = 719.9), W=2973.5, P = 0.0046.

H.2 SalesPeriod(SELL): 719.9
< SalesPeriod(SUB): 1095.0

6.3 Update
For both UpdateCount and UpdatePeriod, we performed the same analysis as shown in the previous section.

Table 7 UpdateCount by type
<table>
<thead>
<tr>
<th>Type</th>
<th>SELL</th>
<th>SUB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>4.5</td>
<td>16.45</td>
</tr>
<tr>
<td>Standard</td>
<td>3.49</td>
<td>4.94</td>
</tr>
<tr>
<td>Deviation</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>Median</td>
<td>0.82</td>
<td>0.89</td>
</tr>
<tr>
<td>W</td>
<td>&lt;2.2e-16</td>
<td>0.006</td>
</tr>
</tbody>
</table>

Fig. 5 Boxplot for UpdatePeriodAfterEndOfSales for all five vendors

Fig. 6 Boxplot for SalesPeriod for SELL and SUB

Fig. 7 Box Plot, UpdateCount SELL and SUB
The Wilcoxon rank sum test indicated that UpdateCount was significantly greater for SUB (Mdn = 16) than for SELL (Mdn = 4), W=8729, P < 2.2e-16.

<table>
<thead>
<tr>
<th>Type</th>
<th>SELL</th>
<th>SUB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>715.6</td>
<td>1997</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>488.6</td>
<td>578.1</td>
</tr>
<tr>
<td>Median</td>
<td>639</td>
<td>1869</td>
</tr>
<tr>
<td>W</td>
<td>0.82</td>
<td>0.89</td>
</tr>
<tr>
<td>P</td>
<td>&lt;2.2e-16</td>
<td>0.006</td>
</tr>
</tbody>
</table>

Fig. 8 Boxplot for UpdatePeriod for SELL and SUB

The Wilcoxon rank sum test indicated that the UpdatePeriod was significantly longer for SUB (Mdn = 1869) than for SELL (Mdn = 639), W=238, P < 2.2e-16.

Both H.3a and H.3b are validated.

H.3a
UpdateCount(SELL): 4
\ll UpdateCount(SUB): 16

H.3b
UpdatePeriod(SELL): 639
\ll UpdatePeriod(SUB): 1869

6.4 Update Period after the End of Sales
Finally, we analyzed the UpdatePeriodAfterEndOfSales.

The Wilcoxon rank sum test indicated that UpdatePeriodAfterEndOfSales was significantly longer for SUB (Mdn = 1164) than for SELL (Mdn = -34), W=3378, P = 0.04449. H.4 is supported.

H.4
UpdatePeriodAfterEndOfSales(SELL): -34
\ll UpdatePeriodAfterEndOfSales(SUB): 1164

7. DISCUSSION AND CONCLUSION

7.1 SUB and SELL
The following table summarizes the comparison of the SELL and SUB models.

<table>
<thead>
<tr>
<th>Type</th>
<th>SELL</th>
<th>SUB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>-46.65</td>
<td>1070</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>577.0</td>
<td>737.7</td>
</tr>
<tr>
<td>Median</td>
<td>-34</td>
<td>1164</td>
</tr>
<tr>
<td>W</td>
<td>0.94</td>
<td>0.90</td>
</tr>
<tr>
<td>P</td>
<td>4.735e-10</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Fig. 8 Boxplot for UpdatePeriodAfterEndOfSales SELL and SUB
<table>
<thead>
<tr>
<th></th>
<th>Average</th>
<th>Median</th>
<th>UpdatePeriod After EndOfSales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>715.6</td>
<td>1997</td>
<td>-46.65</td>
</tr>
<tr>
<td>Median</td>
<td>639</td>
<td>1869</td>
<td>-34</td>
</tr>
</tbody>
</table>

The table shows that SUB has higher frequency and longer period of update than SELL. As we argued in the former chapter, vulnerability remediation heavily depends on the update by the vendors. Therefore, we can say that the SUB model is a better solution for security.

We focus on the frequencies of updates and the period of final update. Research on the details of each update date, size, etc. will be a part of future work.

**7.2 Business model**

A vendor with the business model SUB has two features. First, the vendor leases HGWs to end users. Second, the vendor also acts as a service operator providing Internet connectivity to the end users. It is necessary for the device to continue functioning in order to provide service. There are opportunities to develop and offer new additional services too. In addition, the reputation of the company can be damaged if users have trouble due to product vulnerability. Therefore, for the SUB model, the development costs can be considered to be essential expenses for business continuity.

The release of a new product does not always mean replacement of the existing ones. In an interview, one vendor said that “When the old type of HGW is broken after the release of new one, if old one is in our stock, we replace broken one with old one.” We show this in Fig. 9.

![Fig. 9 Subscription and repair](image)

In cases of repair, products are exchanged, as far as possible, for the same model. In other words, the vendors use the old model for as long as possible.

On the other hand, the SELL business model is simpler, i.e., just sell the products. While the products are available in the market, updates improve their commercial value, leading to an increase in the sales. Nevertheless, providing updates after the end of sales is costly and contributes only to the reputation without increasing direct revenues. A negative value of UpdatePeriodAfterEndOfSales(SELL): -46.65 is a strong evidence of this. The vendors using the SELL model cease to provide updates a month before its end of sale.

There is a need for further research on the relationship between the business model and vulnerability remediation.

**7.3 Applying an update**

Even if an update is provided, it is useless until it is applied to the devices. According to our brief survey, devices that are equipped with automatic updates are few. One of the vendor, using the SUB model, stated that “For 1st generation of HGWs, users had needed to trigger updates by themselves. However, users rarely update. So, we added automatic updates from 2nd generation.” CABLE TV vendors follow a form of the SUB model. Moreover, automatic update is common
7.4 Law, Norms, Market, and Architecture
From the previous sections, we can conclude that the SUB model is more suitable for vulnerability remediation. From a different perspective proposed by Lessig, the SUB model can be considered as regulation by the market. By including Gear’s idea, the regulation to IoT’s vulnerability remediation can be mapped as follows:

**Law**  Product liability act

**Norms**  Providing the firmware as an open source

**Market**  SUB business model

**Architecture**  IoT device must be programed to have an end-of-life

The product liability act is an easy to understand example. Nevertheless, to ask vendors to respond to vulnerability forever is not realistic. As in the case of exhaust gas regulations, it might be a good idea to ask the user to bear a part of the burden.

As stated by Lessig, these four types of regulations, namely, law, norm, market, and architecture, can influence each other. For example, law might enforce the IoT devices to be programed have an end-of-life. We hope that such taxonomy is helpful to consider the regulations for the IoTs, including HGWs, in the future.

**Acknowledgement**
This work was supported by a Grand-in-Aid for Scientific Research (B) (Kakenhi 15H03385 and 15H02696) from the Japan Society for the Promotion of Science (JSPS)

**Reference**


