Scientific Research Publishing

Advances in Internet of Things, 2017, 7, 25-45
http://www.scirp.org/journal/ait
ISSN Online: 2161-6825
ISSN Print: 2161-6817

Internet of Things Behavioral-Economic Security Design, Actors & Cyber War

Robert L. Shuler, Billy G. Smith

NASA Johnson Space Center, Houston, Texas, USA
Email: robert.l.shuler@nasa.gov, robert@mc1soft.com

How to cite this paper: Shuler, R.L. and Smith, B.G. (2017) Internet of Things Behavioral-Economic Security Design, Actors & Cyber War. *Advances in Internet of Things*, **7**, 25-45.
https://doi.org/10.4236/ait.2017.72003

Received: March 24, 2017
Accepted: April 27, 2017
Published: April 30, 2017

Abstract

Using security incident history we identify threats to and using the IoT and other ubiquitous devices emerging since 2012, gaining widespread recognition in 2016, and only lightly addressed in either IoT security literature or the press. We show the IoT has likely already been used in cyber war between major powers. The new threats, most notably "hijack," are larger than previous threats combined, but only mildly affect suppliers, and only a few clients. Using a successful behavioral-economic model we show that traditional mitigation places responsibility on un-affected parties and likely will not work. For suppliers, there are profit-conflicted motives, as the new threat rides on a profit vehicle. The new threat circumvents conventional security architecture at a behavioral level. We analyze each actor-target pair and evaluate technical strategies. More effective technical strategies are suggested where old ones are overmatched by the budgets, technical prowess or regulatory power of hostile actors, or the technical nature of the threats. Consolidated action may be needed, but regulation is difficult because of conflicts of interest within the national security community.

Keywords

Security, Reliability, Privacy, Distributed Denial of Service, DDOS, Botnet, Cyberwar

1. Introduction

This paper addresses actors which have stakes, gains and losses with respect to the Internet of Things (IoT), showing through an analysis of prior IoT security literature that supplier and customer/user are not necessarily the largest stakeholders. This may lead to incorrect assessment of threats and consequences, and omission of effective choices from device and system security and reliability design. To properly evaluate the effectiveness of security measures it is necessary to

correctly identify actors causing risk, and compare the economic factors motivating those actors, manufacturers and users as they make tradeoffs among security, features and applications. We apply a formal theory of behavioral-economic risk vs. innovation [1] which was developed initially for spaceflight software and avionics, expanded to full spacecraft systems and operations [2], and has been successfully used to explain differences in motorway death rates between the U.S. and Germany as a function of detailed wealth demographics in which low stakeholders were making unexpectedly large contributions [3]. We expect to find similar stakeholder-related effects in the case of the IoT, with consumer stakeholdings over-estimated and higher value stakeholders overlooked, leaving the IoT vulnerable even if technical targets are met.

Security concerns for the IoT were raised in 2011 based on an IoT defined primarily as sensors, but also "objects," connected by RFID, non-Internet wireless, WiFi and hardwired Internet. Primary issues identified were with regard to wireless connectivity. Vulnerabilities were identified as: 1) control of gateway nodes, 2) denial of service (DOS) attacks on sensor nodes, 3) interference with the network signal, and 4) identification, certification and control of large numbers of sensor nodes [4]. Early mitigation strategies identified were: a) gateway node encryption and key control, b) defense against DOS attacks on sensors through non-internet remote control, c) by-hop or end-to-end encryption with recognition that some businesses will have low security requirements while high-security may conflict with some national policies for lawful interception, d) varying levels of business authentication depending on whether the business is a high-value target (e.g. financial) and trusts the network layer, and e) other unique IoT requirements such as the need to manage a very large number of devices and the possibility some of them are performing dangerous mechanical work or safety-critical monitoring [Ibid. 4].

Later analyses (in 2012) continued with the same problem identifications, adding recognition that as the IoT is a very large composite system existing and changing over time, "the security mechanism of each logical layer cannot implement the defense-in-depth of [the] system," that there was as yet "no technology standard about the IoT," and that legislation was needed (reference author's country was China). Significantly, key management was identified as the most important basis of the security mechanism. It was noted that researchers had been unable to find ideal solutions, and that effective key management was seldom put into practice in real large-scale systems [5].

By the next year it was recognized that the IoT might be used in medical treatment, public health, intelligent transportation, the smart grid and for other critical purposes, with a "comprehensive intellectualized object" appearing after 2020 and gradually covering "every aspect of social life". Protecting the application became the unconscious emphasis of researchers and designers. Protecting gateway nodes, preventing key leak, and providing resilience to DOS attacks on the application continued to be primary mitigation strategies, with the new recognition that conflict between security and cost was more acute than with older

network terminals (PCs and POS terminals). The IoT was dependent on a large number of low-cost nodes in which security comparable to previous network terminals was infeasible to provide or administer. It was realized that the type of application, not the cost of the node, should determine security [6], but security concerns unrelated to the application, business or supplier were still unrecognized by the research community at the end of 2013.

The year 2014 saw discussion of connected vehicles, including vehicle to vehicle, Internet and road infrastructure [7], as a subset of a larger IoT architecture which was described as "smart cities" with environment, safety, food, logistics, traffic [8] and even crowd sensing [9]. By 2015 the potential consolidation of data from IoT systems in "the cloud" had raised 20 specific security issues, many of them new. Five of them were identified as "relatively unexplored," including 1) inter-application sharing, 2) re-identification of private data as a result of sharing, 3) lack of demonstrated audit and compliance methods for cloud service providers, 4) legal and privacy implications of 3rd party services, and 5) the impact of cloud decentralization. Additionally the "goals" of cloud-supported IoT were found to conflict with earlier security protocols. For example, encryption techniques are mature, but use of them prevents application sharing and synergistic cooperation among IoT systems [10].

By 2016 six billion connected "things" were in use, with an annual growth rate of 30% and projections of 20.8 billion "things" by 2020. The value of this business to suppliers was too compelling to slow down and address any lagging issues, with $235 billion in spending in 2016 at a 22% annual growth rate. A new category of customer/user, mostly not addressed in prior security surveys, was the "consumer," accounting for 39% of revenue but 63% of the installed base, and growing four times as fast as all other categories combined [11].

2. Method

To apply our behavioral-economics method, we must identify actors and gains/losses from system failure or compromise. Since we suspect omissions from the academic literature, the first step is a survey of news sources to provide empirical data on failures that have actually occurred within the prior four to five years. Once this is identified, a set of cases will be established for various combinations of actors, and our model applied to each case. The results will then be discussed relative to whether the mitigation methods so far proposed would be effective, and some suggestions made for areas in which to look for new methods to mitigate the unprotected vulnerabilities we find.

2.1. Data Collection

The large army of 6.6 billion waiting robots, known as the IoT, too numerous and low-cost to protect but as capable of generating internet traffic as other devices, did not go unnoticed either by criminal hackers or the national security apparatuses of nations. Figure 1 shows the general scheme of a Distributed Denial of Service (DDOS) attack. A much larger number of slaves or "bots" are used than in a traditional DOS attack

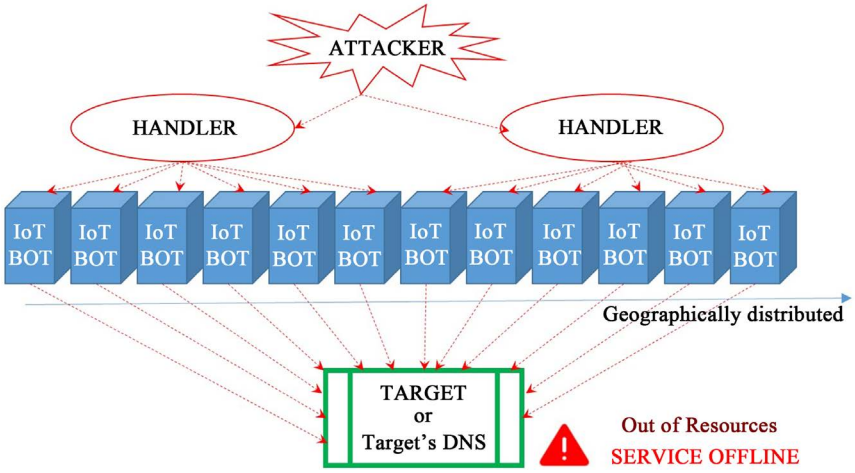

Figure 1. Diagram of DDOS attack.

(non-distributed). The impact on each bot need not be great, and they are generally distributed worldwide, making many blocking strategies ineffective. The target simply runs out of resources. Attacks include a variety of strategies:

- Connection attacks (occupying connections)
- Volume attacks (using up bandwidth)
- Fragmentation attacks (taxing victim's ability to re-assemble streams)
- Application attacks (overwhelming an application service)
- DNS reflection (obtaining reply up to 70x larger than request)
- Chargen reflection (similar to DNS reflection using outdated chargen service)

The first massive DDOS attack using a botnet of "things" instead of PC's seems to have occurred on and following Sept. 19, 2012 in Hong Kong. The "things" themselves were not targeted or harmed and only mildly impacted. They were organized to attack major banks: Bank of America, JPMorgan Chase, Wells Fargo, U.S. Bank, and PNC Bank. The incident was described as "20 times" larger than previous ordinary DOS attacks, a record by a "wide margin." Banks have some of the most sophisticated defenses against such attacks available, and the attack itself was described as "unsophisticated" [12].

One might expect such a high-value target, naively attacked, would immediately protect itself. What actually happened was (partial list):

- Mar. 18, 2013 Spamhaus was hit with a new record attack volume fluctuating between 30 and 90 Gpbs, then after resting a day hitting 120 Gbps. Spamhaus contacted Cloud Flare which successfully mitigated the attack [13].
- In 2013 in China and 2014 in Hong Kong an international organization of hackers began launching DDOS attacks on dual banks, demanding Bitcoin blackmail payment from one to cease attacks on the other. Banks lose as much as $100,000 per hour in such attacks. The organization was thought to be DD4BC, but the investigation has not been successful. The first attack is thought to be retaliation for regulation preventing Chinese financial institutions from holding or trading digital currency [14].
- In June 2014 massive DDOS attacks continued in Hong Kong, now against a pro-democracy news site (different from the attacks on banks that were

blamed on an activist). The nature of the target makes state actors suspect in the attack. The response was to extend a civil referendum for a week [15]. This time the attack overwhelmed even Cloud Flare's defenses with a 300 Gbps attack [16].

- On May 9, 2015 attacks against Hong Kong banks continued to demand Bitcoin payouts, and overwhelmed the same banks which had had by then three years to protect themselves [17].
- Shorter more intense attacks cause banks to lose $100 k per hour, and may not be an end in themselves but distractions to cover other transactions the culprits are making [18].

The attack tools, marshalling the growing IoT, were still outpacing the growth in defensive capability in 2016. Neither the attackers nor targets were stakeholders in the IoT behavioral-economic dynamic that drove development of products, features and security analysis. Further, the IoT soldiers didn't die in the battles. It was never reported that any consumer was ever aware their device was being so utilized, or suffered any inconvenience for it. No companies were penalized for offering insecure products. None of the technologies were regulated or banned. But the subject was broached indirectly by the ban on institutional Bitcoin holdings in China.

Just how does a bank lose $100k per hour in a DDOS attack? The bank's customers are effectively captive, and will postpone their transactions and payments. But commercial payment processing may be forever lost. Consumers will switch to another payment, or just forego purchases. If the bank's lose $100 k of small fees on transactions, of the order of less than 2%, the loss to the economy could be at the low end $100k / 2% = $5 million an hour. That is up to $100 million dollars a day. If the attack is carried out on multiple banks, the loss is much greater, perhaps half a billion dollars a day. It is really an attack on an economy. The banks are used by the attack to deny economic services to the economy, much like the IoT is used to deny the banks access to their transaction processing technology. If the IoT continues to scale up unchecked into the 2020s, it could be used to cause losses approaching a hundred billion dollars annually, perhaps a tenth of the annual IoT sales revenue in that time frame. That compares with the loss-production efficiency of expensive automated military systems, but the entity using the IoT for this purpose isn't even the one paying for it. This makes a vast difference in behavioral-economic models.

In July 2014 an HP study found that 70% of IoT were vulnerable to attack [19]. The listed causes both compare and contrast with the issues identified earlier by IoT security researchers:

1) Privacy concerns–8 of 10 devices collected name, email, home address, date of birth, credit card credentials, health information, and other details that might be used for identity or credit theft, extortion or blackmail. So far we do not have reports of large scale use of such information.

2) Insufficient authorization–80% of devices failed to require passwords, or were configured with weak and universal passwords such as "1234." This was an

enabler of later attacks. However, they are probably configured this way because consumers would not be able to use them and would not buy them if every IoT device required a unique password. Imagine if the owner of a house died and all of the home automation from climate control to the refrigerator was rendered long-term inoperative because the passwords died with him. Recall that the owners of the IoT botnets used in the bank attacks perceived no loss, and have no motive to act. IoT security researchers, recall, had been concerned with encryption key leakage, not password loss. Passwords are much harder to protect than keys because they must be remembered by users and are more likely to be used for multiple devices, or available from collateral sources through data mining.

3) Lack of transport encryption–70% of devices have this problem even though we report it was well-studied and in the literature since at least 2011.

4) Insecure web interface–this topic was also well studied, yet 60% of devices were found deficient.

5) Inadequate software protection–60% of devices did not use encryption when downloading updates. "Some downloads could even be intercepted, extracted and mounted as a file system in Linux where the software could be viewed or modified." The need to secure updates has only recently appeared in the research literature, so this deficiency is not surprising and might be corrected, though the lack of progress on other long-reported items does not bode well.

The reason items 2 - 5 have not been corrected is to be found in behavioral-economic factors. Neither the suppliers nor their customers are actually losing anything because of these deficiencies. Currently IoT devices only control limited financial transactions–such as Amazon Dash Buttons–tied to a physical delivery address and not easily used for ordinary theft.

The reason item 1 has not been corrected is probably because criminals have not figured out how to use the information for high value attacks yet, and governments already have the information. High value attacks with personal information are either of the old identify-theft variety, or require one's contact list for social engineering. The 2014 Yahoo account hack data is just now being used to launch spam-fraud attacks of the "I'm stranded, send money" variety to all the contact entries in one's Yahoo account. Perhaps no strategy will be devised to use IoT information in a large scale way since it is not yet correlated with an address book, but that is not to be counted on. If data pirates are not already using data mining, they will be.

We will make the case from 2016 events that the largest nation-state actors are now involved, if they weren't already, that the IoT army has been used in a major attack between the world's two most powerful nations, and that future consequences will result in an impact on the IoT industry, if only the loss of economic activity on which it depends, but probably worse:

- In May or June of 2016 the US Democratic Party presidential campaign manager lost his password to a simple phishing attack, even after consulting with

his I.T. manager. This was used to eventually get a variety of emails from Democratic National Committee (DNC) officials that resulted in charges of bias and in resignations, embarrassing and allegedly harming the leading Democratic candidate [20]. While not an IoT event in itself, this provided the context which appeared to trigger eventually a full scale nation-state use of the IoT in an on-going not-entirely-secret but no-field-reporters-invited cyber war.

- In mid-September 2016 Krebs On Security was hit with a DDOS attack double the size of any their protection service Akamai had previously seen, and among the largest ever seen on the internet. It was over 600 Gbps, 20 times larger than the average and 6 times larger than peak attacks of 2012-14, and was executed using "(IoT) devices - routers, IP cameras and digital video recorders (DVRs) that are exposed to the Internet and protected with weak or hard-coded passwords." It operated by overwhelming the DNS servers which resolve Internet addresses [21]. Brian Krebs later said he thought the attack was retaliation for a blog post about two Israeli hackers selling attack software. No loss was reported. One commenter observed "The funny thing is that you got free advertisements for your website." *But Two Israeli men were arrested at the request of the Federal Bureau of Investigation (FBI).*
- At the end of September experts said the attack was probably the Mirai Botnet, and that similar attacks may be a nation assessing the strength of core internet defenses [22]. If so, the intent may not have been to inflict damage, but a test.
- Less than a week later the code for the Mirai Botnet was anonymously released on the internet so that anyone could use it [23]. Notice that at this time who released the code is uncertain. If the two young Israeli's had it as Krebs alleged, then the FBI presumably also had it since their arrest just after mid-September. If the code were released in retaliation, whoever released it would have to be sure the FBI could not identify and apprehend them. It could have been released as a cover, so that the identity of future attackers using the Mirai Botnet would be deniable.
- On Oct. 8, U.S. President Barak Obama announced that Russia was responsible for the DNC hack (the phishing and follow-on activities; there was no DDOS against the DNC).
- On Oct. 11, U.S. President Barak Obama announced that there would be retaliation, something that would have an impact, and that Russia would know what it was (but not necessarily that anyone else would know, *i.e.* he wasn't going to announce the event) [24].
- On Oct. 21 the largest DDOS attack ever released hit the US east and west coasts, Texas and to a lesser extent other areas, using the Mirai Botnet. No permanent damage or theft of information was noted. Some suggested it was a test by a nation. No one claimed responsibility [25].
- On Oct. 26th the FBI released a bulletin advising users of IoT devices to change their security credentials (passwords) to protect against future DDOS attacks using the released code [26]. No one that the authors know changed

any passwords. Again, the impact and actors are not aligned. It almost seems as if the FBI is purposely keeping the botnet, of which they are now in control, available for future use by recommending ineffective countermeasures. One day later experts in private industry reluctantly called for "government regulation" [27]. However, we already know that the National Security Agency (NSA) under the Patriot Act requires back doors into most information systems, and China has similar requirements as noted earlier, so it seems to the authors it is unlikely to be in a major government's interest to shut down the botnet.

- There were attacks outside the United States on Nov. 2 using the same botnet, including one non-bank financial company [28].
- There were concerns that the US election on November 8th might be disrupted, but no problems were reported. However…
- On November 10th it was revealed that at least 5 Russian banks had been under a massive DDOS attack for two days [29].

It is highly unlikely the President of the United States would make a definite declaration of action, with no conditionals in the statement, and not follow through without some public notice that the situation had changed. Therefore the promised retaliation either has happened or is still pending. It would not have happened before Nov. 8th because of concerns of a counter-retaliation. Every day afterward the current president became less able to direct national policy. His term was shortly ending. That argues that the retaliation must have happened shortly after the election. Only one event was announced by the Russians. It was confirmed to have used a weapon that we know the United States had control of, and experts suspect some nation had conducted two or more tests of it.

2.2. Discussion of Data

To understand trends in information and communication technology, we have available a history more than ten times as long as the period of existence of the IoT. What it shows is that governments have always been a major player when they need to be, beginning with the cracking (hacking) of the German Enigma in World War II and construction of the Bombe as a tool to that end. From then until the creation of the first computer worm in 1989 most hacks were lucky observations and most damage was intended as pranks or at worst malicious mischief. In the 1990s the development of polymorphic code allowed rapid virus spread and made detection difficult. In the late 2000s corporate servers came under attack with the hack of TJX by organized criminals costing the company $256 million. In 2013 there was the Target breach and the Snowden leak, both apparently inside jobs [30]. In 2014 the largest Bitcoin exchange Mt. Gox, lost $460 million to hackers and went bankrupt [31].

We can identify five shifts over time:

1. TARGETS: individual users → corporations & government
2. GOALS: damage & disrupt → theft & extortion

3. TOOLS: self-destructive malware → borrowed botnets
4. ACTORS: pranksters → criminal gangs → governments
5. MITIGATION: none → pattern scanning → real time updating

Perhaps the most significant change is that the target of infection is no longer the target of theft or damage. Though this evolution may have occurred somewhat by accident, its advantage surely has been recognized by perpetrators.

Online or near-real-time updating became popular as a mitigation for PCs beginning in about 1999 after a disastrously buggy and insecure release of Microsoft Windows 98, and a large percentage of users beginning to be connected to the internet by higher speed connections, many full time. In 1995 CompuServe, AOL and Prodigy had opened for business, and by 1999 Shawn Fanning invented the music sharing application Napster and there were 150 million internet users worldwide. The stage was set to spread both malware and anti-malware. There was essentially no alternative to updating, as the malware could change. At first, there was no concept of updating anything but the malware patterns.

Operating system changes came fast—Windows Millennium, 2000, XP, Vista, 7–at only 2 or 3 year intervals. By Windows 10 it was hard to find software on physical media anymore, and newer computers did not even have CDROM drives. By the late 2000s nearly everyone had a WiFi router, and not just computers and phones were connected to it. The IoT had already begun. In the hectic competition for features and market share, updating became more than a way to combat threats. Its other uses came to include:

- FIXING BUGS AFTER RELEASE in firmware and installation software. In the rush to be first to market, it was impossible to test every possible user configuration. Let the user's test it, then patch it. If you have beta users for this purpose, most of the bugs can be fixed before users who would just return items to the store encounter the bugs.
- ADDING FEATURES AFTER RELEASE to match competing features or pre-empt possible competitors.
- MAKING CHANGES TO BREAK old hardware or competing software. New system releases had always broken some percentage of old software, but consumers could no longer be counted on to buy new OS releases, so with Windows 10 updating became mandatory.
- MAKING CHANGES TO THINGS BROKEN BY COMPETITORS' updates. The update war quickly escalated.
- ADDING FEATURES TO BETTER CAPTURE one's user base and collect user data. Auto-updated releases of Skype, Android and Gmail import each other's contact address books. Entering a friend's email in your phone automatically calls up a photo of them from Google+. It's really clever. One of the authors even likes it. But it means Google has total knowledge of everywhere the author goes, what he buys, who his friends are, and what to sell him next.

For example, Microsoft Edge and Firefox used predatory updates to take over viewing PDF documents without asking permission (malware if done by lesser

companies), but with inferior font resolution and no annotation ability. Adobe relies on updating to keep its products competitive, and to answer accusations (true or not) regarding Flash, which controls most video advertising to the envy of all internet media players. In short, the primary function of updating is no longer protection. It has a much more valuable profit-competition role. The case for updating to protect PCs can still be made. A PC is a platform for loading and running various software packages from unknown sources (even web pages), which might be infected. If a person uses a phone like a computer, one might make the argument for updating. But it does not hold because the apps are mostly loaded from a tightly controlled "store" managed by Apple, Microsoft or Google, and the operating systems are more restrictive. The real purpose of updating is so that if the vendor of your child's baby-alphabet-music app wants to add more free sounds–or wants to limit the capability of an app they previously gave away free–they can do so even though you installed it months ago. Most people do not even want their webcam or printer to do anything new. HP used updating of WiFi printers, for example, to improve detection of non-HP branded ink cartridges and to refuse to print if they were installed. At the time (about 2014) a new printer contained a full ink cartridge and was cheaper than just buying the ink cartridge.

We conclude that the largest threats to IoT security come from nations fighting wars with trillion dollar military budgets, undisclosed budgets for clandestine and information technology apparatuses, and a history of having used malware in the past (in 2009, the Stuxnet virus attack on Iranian centrifuges [32]). The designer of a $10 device to be used by a consumer to solve a $10 problem does not stand a chance of avoiding "conscription" if relying on a purely technological defense. Even if a successful defense was constructed, it would likely be legislated out of existence to preserve national security access. And even against ordinary hackers and peeping Toms, the odds are not actually very good.

We further conclude that the paradigm for protection that prevailed from 1999 through the introduction of updates-mandatory Windows 10 no longer has the actual purpose of protection. It has become a "feature" and a "means to add features continuously" which in the next section we will learn is disastrous for security (or reliability of any kind). The function of protection may now be better served in other ways.

2.3. Analysis

A crash rate equation has been derived and verified in previous literature [Ibid. 1] which takes into account economic factors, assuming that economics governs behavior. In the short run, other factors may prevail, but the model assumes a mechanism similar to natural selection operates due to commercial competition, so that those who employ economically optimal behavior come to control most resources. Competitors either go out of business or are absorbed. Briefly, so that readers do not have to consult papers with lengthy background arguments, the variables in the crash rate equation (which are defined somewhat loosely as we

shall see) are as follows:

R_O operational crash rate;

V_F value of the features (or function) per unit time;

M cost of manufacturing, marketing, distribution per unit time;

C_d (engineering) costs per development problem;

D defect ratio;

C_C cost per crash.

Rather than predicting a crash rate, often we have a measured crash rate and wish to find other parameters that will help influence the crash rate. We can take from our above summary of events that for the final half of 2016 there has been an average of something like one massive newsworthy attack per month, although most of them were grouped in a three month period. So there is an uncertainty as to whether we should be using R_0 = 1/month or 2/month. This is about as good as it gets for setting this parameter. Unless one has years of data, it is not known better than that, and even not with years of data if systems are perfected so that crashes are infrequent.

The equation (1), shown below, was originally intended to be used to determine the optimum setting of a few production and testing parameters. It was assumed that the company was competing on features, and had little choice but to add them if they were practical, due to competitive pressure. However, in this case we are talking about security features which some users anyway disregard as of little value to them. In fact, the value of the security features depends on whether it is a personal device, an industrial or medical device, and whether we are taking the point of view of the actual user, or some 3rd party who may hijack the device and use it for a very different purpose. So instead of just one set of values for the crash rate equation, we will have a set of cases with different values according to the *actors* selected. Without further ado:

$$R_O = \frac{V_F - M}{C_C + C_d / D} \tag{1}$$

For a full explanation please see the literature. A brief overview follows. The value of the features V_F in the numerator implies that crashes will be more frequent for valuable services. People will be willing to risk more for something very valuable. People's time is valuable, so they risk talking on the cell phone while driving, for example. This causes more crashes. Sometimes there is a user education problem, but after a while everyone knows smoking causes cancer, and some people do it anyway. Adding features increases crash rate. Anyone who has used a computer fully comprehends this. The Internet is full of programs to wipe out things you have installed and accumulated that are causing crashes. Someday it is likely there will be robots to find and kill all the old IoT in your house. But the features of current concern, security-related, while very valuable to an end user on a PC, are largely ignored on DVRs, cameras, etc. Indeed, the lack of any reportage of consumer complaints of their IoT devices being used in botnets illustrates that in this regard the equation is correct. Consumers are experiencing zero impactful crashes of IoT devices due to security features–or

the lack of them.

It is particularly important to understand that the ability to update products in the field, as noted earlier, is a *feature*. It is not necessarily a security feature, as its primary purpose is something else–to ensure customer loyalty, enforce lease agreements, outpace competition, fix bugs in hastily tested products, realize revenue for added functions, or to claim leadership in a field, perhaps even to encourage field testing. In 2015, after a year of selling cars with sensors, Tesla "…sent its cars a software update that suddenly made autonomous driving a reality" [33]. There were predictably a few crashes, but the real risk will come when millions of them are at the disposal of an autobotnet.

The M in the numerator, subtracted as manufacturing costs, indicates that if an item has a high profit margin, the company will be anxious to get it into stores and careless in ensuring its safety. Crashes of new self-driving cars are an example. As M approaches V_P, the item is said to become a commodity, like rice, beef, coffee, oil or pork livers. Generally the crash rate of commodities is near zero. Who ever heard of people dying from bad rice? Governments regulate and inspect commodities. Sometimes people get sick from beef, but there is an outcry over even one death, companies are shut down and sometimes go out of business. Oil transportation may cause issues, but oil itself, even though there is a tank full of it at every house and self-service gasoline stations in every neighborhood, rarely blows up killing anyone. Even air travel is a commodity. It is much safer than driving, and companies that crash go out of business.

This is important because the IoT, and electronics generally, is on a trajectory to become a commodity. Transistors are already much cheaper than grains of rice. Except for one or two high profile companies that have a brand premium, the margin has been taken out of electronics for most companies. That is why we said earlier that impacts from IoT security issues will be felt eventually. If IoT devices become a commodity, then when they cause a car or train crash, or the loss of millions of depositor dollars, or other great social losses, companies will go out of business, and regulators will appear at the doors of the rest of them. The crash rate equation has limited ability to change the destiny of a product category. But it enables its producer to choose the winning endpoint and go there immediately by design instead of by trial and error. Mostly, in the case of the IoT, it seems to have been error, as a major military weapon has inadvertently been created and handed over to nations without even compensation to its developers.

The cost of each crash C_C in the denominator is the "feedback" which keeps crash rate down. The cost of going out of business is the total amount invested. Other costs include damage to reputation, toxic waste cleanup, product returns, lost product cycles as in the case of the Samsung Galaxy 7 (possibly due to adding the "feature" of rapid battery charging [34]), refunds, and staff for customer support, etc. So far the cost to suppliers and users of hijacked IoT devices is near zero, so the crash rate is very high.

The remaining terms C_d/D are intended for design and manufacturing quality

control, to determine how much development testing and product inspection to do. C_d is the cost of design and development, including testing. D is the defect ratio, the percentage of products that are defective in a screening. In this paper, we assume every device works as designed, at least initially. In the manufacturer point of view in the next section, we'll briefly consider them in regard to the update problem. Updates allow the introduction of problems, either by the manufacturer or hostile parties, after deployment. Testing and screening are important in quality control and the crash rate equation makes it clear why. The defect ratio leverages development cost, providing the only way to gain an advantage in crash rate disproportional to the actual amount of money spent.

The crash rate equation was developed to take into account manufacturers, designers and customers. But it can take into account different actors, such as threat organizations, because certain of the terms, such as V_F, are actor specific. For a threat actor, the term C_d could represent the budget and effort of the organization posing the threat. In this paper we will analyze the possible crash rate for these cases separately, rather than trying to write some kind of composite equation.

3. Results

3.1. The Consumer Point of View

Suppose a consumer pays $300 for a DVR. (They currently range in price from $200-$500). What is the value of that? If it has to be replaced, say, every two years, then it is at least $150 per year, $12.50 month. What is the cost of a crash? If the DVR loses a movie it was programmed to record because it was hijacked by a botnet, and the movie cost, say, $5, then the consumer will be annoyed if it happens once a month and make complaints. At twice a month the consumer will replace or return the DVR. That is the point at which the crash rate approaches the utility value of the device. If the consumer hears of a more reliable device, the DVR will be returned and replaced before it reaches that point. Thus we suggest the appropriate crash rate equation is approximately:

$$R_O = \frac{V_F}{C_C} \Leftrightarrow \frac{\$12.50/\text{month}}{\$5/\text{movie} + \text{annoyance}} \le 2.5\,\text{hijacks/month} \quad (2)$$

It may seem a coincidence to the reader that our example just happens to match the roughly once a month actual botnet attack rate in the last half of 2016. But is it? We did not make up any of those numbers. The test would be to make a change to the equation, and see of the crash rate changes. For example:

1. Make the DVR so cheap it is essentially disposable. Lots of devices are already approaching this. Then the DVR would be quickly disposed, and the consumer would run through different brands until finding one that "works," *i.e.* that was not susceptible to the botnet.
2. Make movies and cable TV so expensive that the value of a movie is ten times greater. This is not competitively feasible, but you can see it has the same effect as #1. The consumer will dispose of DVRs until finding one that works.

3.2. The Supplier-Manufacturer Point of View

How did the botnet get control of the DVR in the first place? The usual method is to induce it to load a software update which contains the botnet client malware. Who is the update feature for? It can be for the manufacturer who does not take the time to completely test and certify the product before shipping it, or for the cable company who might want to control it, though the authors are not familiar with cable company motives.

In the case of the manufacturer, simply remove the update capability altogether. Take an extra month to fully test and certify the device. By that time the consumer will be disposing of his first DVR, and will buy the one that works.

From the consumer point of view, the no-update IoT device is a feature they will pay money for. Both authors would pay money to turn off automatic updating on Windows 10, but Microsoft does not offer the option. Updating is too valuable to Microsoft to push not only new features, but for advertising. Hardware manufacturers do not object because it quickly obsoletes devices. A recent purchase of a Windows 10 tablet had all of its built-in memory used by about the second update. Most likely it will be returned to the store shortly and replaced with another.

But when, not if, the competitive situation develops with the IoT such that crashes are noticed by consumers and causing problems, the company with a non-updatable and thus immune product will likely win market share.

To analyze this quantitatively we must make a number of assumptions, so the analysis is illustrative rather than definite for a particular design. If the DVR function is provided by a general purpose computer, as is becoming the case when services like Netflix allow content to be downloaded for offline playback, the option to make it not-updatable does not exist. However, the focus of this paper is on more specialized IoT devices which could be made updateable or not. We must compare the cases:

1. Lost business due to high end-user crash rate
2. Additional testing to verify non-updatable DVR
3. Some method of screening for deployed DVRs

In the literature on crash rate theory [Ibid. 2] post deployment collection of test data is emphasized as an effective way to extend the advantage of the C_d/D leverage term without overly delaying product rollout. This corresponds to option 3, which is often used by software which provides deployed diagnostic information to the manufacturer to support product improvement. So far as the authors could determine, this does not yet extend to screening for whether a device has been infected or hijacked, nor do we assess herein the technical feasibility of such screening, only the economic effect it might have if possible.

We assume the manufacturing cost is less than a third of the retail consumer value, so $V_F - M = \$300 - \$100 = \$200$ which over the 2 year life cycle is *$4.16 per month* amortized net proceeds. We take the observed and consumer crash rates as a given, and estimate a "warranty return" rate (C_c) that is consistent, as well as leveraged development costs (C_d/D), to obtain the denominator terms.

This gives a monthly amortized $C_C + C_d/D$ of \$1.66 (\$4.16/2.5 crashes per month maximum). Suppose the manufacturer returns the full retail price paid (not applicable in a rental situation, which adds complexity). Suppose further that the manufacturer diligently pursues reducing whichever denominator cost is greater, C_c or C_d/D. Then perhaps they are of similar magnitudes, and for simplicity we assume they are equal. Then that corresponds to an eventual return of $(1.66/2)/12.50 = 0.83/12.50 = 6.5\%$ of devices. This is the lost business due to end-user crash rates in case 1.

In consequence we have surmised a consequential leveraged development cost of \$.83 per month or about \$20 per unit without having to determine how many units were produced and sold. The literal development cost per unit might be much lower, for example only \$2 if D = 10%, and much less for a higher quality operation. Some manufacturers, though, will deliver products with multiple defects and rates as high as 100%, depending on updates to fix them. We do not specifically analyze consumer reaction to this situation. However, we consider the device to be delivered to the consumer when it is delivered and updated if necessary, assuming returns would be 100% otherwise, and that D = 10% refers to this entire process. The manufacturer has effectively utilized "free" consumer labor in its screening (something which the authors believe consumers resent, even when they have few alternatives but to accept it).

To analyze case 2, we roughly replace the "free" consumer screening labor and time with manufacturer effort. If D = 5% can be achieved, then the same crash rate is obtained but with C_c = 0, *i.e.* crashes are reduced to an annoyance which does not result in warranty returns or disturb brand loyalty. Pinning down exactly how much it would cost to do that is not a suitable subject for this general paper, nor is analysis of whether a manufacturer might want to go further in pursuit of reputation or brand loyalty objectives. Suppose, however, that the manufacturer was willing to spend 20% more on development (C_d) to reduce C_d/D. Then we only have to reduce D to 6% instead of 5% to achieve the same result. C_d/D is a very powerful operator often overlooked since the heyday of 6-sigma quality emphasis which allowed Japan to successfully penetrate world markets. (Perhaps to some extent the markets are different, as penetration of low-end economies and volume of low-cost products play a larger role?)

Case 3 requires a possibly substantial increase in C_d to achieve screening for security and hijack events after deployment, but allows updating to be retained (possibly). Updating has other functions and value. For example, the DVR function might need to change to accommodate new capability offerings by existing or new content providers. Users might ignore new offerings if they have to take action to obtain the capability, resulting in loss of revenue or dropping support for this type of DVR by content providers, or users might choose a different brand if they have to take action.

The manufacturer does not have to spend all of the potential value gains from updating to keep the equation balanced. It is only necessary to spend a combination of Cd in conjunction with a reduction in the defect ratio D (in this case, as

deployed, reflecting increased immunity to attack) to keep C_d/D commensurate with case 2.

However, the analysis cannot be conclusively done in this case without considering the efforts and expenditures of possibly governmental organizations determined to preserve the ability to monitor or control the deployed devices. That is so difficult to estimate that we decline even to present guesstimates.

3.3. The DDOS Victim Point of View

The banks have no point of leverage since they neither buy nor use the particular IoT devices being used in the attacks. Apparently they have already decided they are doing all they can. A catch-22 can arise with too much security. The legitimate customer cannot access their account. An example happened recently to one of the authors. A login was blocked because a computer was not recognized. The override asked for a "security code" that was supposed to already be present in a Google-registered phone. But it was not present on the phone in question. It appeared no new computer could ever be added to this account! So the author in question spent two hours on the phone with various support personnel and eventually the problem was solved by increasing what Google considered to be the level of security in the account, so that a text message containing a code was sent.

In other words, if banks become too secure, no one would remember the combination to the vault (so to speak). They are already losing customers to Paypal, newer services, even in some cases the troublesome Bitcoins.

3.4. The National Security Point of View

The NSA has already demonstrated that no matter how clever your encryption, they will demand a back door. They consider the value of access to your systems to be, well, a "matter of national security," of the life or death of a nation, many trillions of dollars. Encrypted message services with no back doors and refusing to cooperate have simply gone out of business (e.g. Lavabit and Silent Circle [35]). And we have made a strong case that these are the most troublesome actors who will interfere with your business, and already have.

Can we apply the crash rate model to what appear to be non-stakeholders? Yes, by recognizing that they do hold stake in what is consequentially affected. The national security agencies are responsible not just for physical but also for economic security. They also wish to preserve the effectiveness of their weapons. Earlier we observed that in the near future the total economic loss from a major IoT DDOS attack on financial transactions could approach a tenth the value of the annual sales of IoT devices. This suggests that a rate of something less than ten such attacks per year would provoke a response from the IoT industry, or damage it so severely as to harm the world economy. If we average our 2nd half botnet attack data over the entire 2016 year, which we probably should have done, then indeed the NSA/FBI and other unknown parties may already be intuitively operating at what will be the maximum sustainable attack (crash) rate

in a few years.

There is a way around the problem. Most engineers won't like it, but it is essentially bulletproof. There are two steps. First, the product must actually become a commodity, with extremely low profit margins. This does not mean it will not be very profitable. Most commodities are. But it means the national security interests have less leverage. If the product has a high profit margin and a lot of complexity, and does not have the feature they want, they can demand it be added. But if the product omits the capability to be updated, other than by physical replacement of the product or some part of it, and the cost is so low that changes are unreasonable, then court cases will uphold the supplier's position. However, there is a risk that if this is not the path taken early on, the national security interests will argue that it is contrived. In fact, the author suspects the market will eventually force it. The idea is to anticipate, using the crash rate model and principles, and be at the eventual optimal operating point.

3.5. Privacy Issues

That completes the analysis of all the actor positions (except actual hackers, who are a kind of weaker version of national security, and often become employed by national security when they are caught) for DDOS and similar hijack related crashes, or vulnerabilities. One could do the same sort of analysis for privacy issues. The national security apparatus also believes personal information to be a high value target, such as for terrorist surveillance. One can only imagine what they will surveil once we have mental enhancement implants. In fact, the IoT is probably going to set a precedent for future more personal technology, as some of the devices will be more connected to the body than a phone. They are the next step in intimacy with our electronic spawn. Wired physical security works, of course. There are attempts to create physically unclonable functions to, in a way, emulate wired physical security. The easier ones to fabricate have turned out to actually be clonable [36].

3.6. Clever, Partitioned and Hands-On Updating

The main point with regard to the various "other" security issues for purposes of this paper is that their solutions will not necessarily address the hijack problem. Even though the hackers (possibly working at government agencies) are always coming up with new exploits, the way to fight a trillion dollar military and security apparatus may not to make auto-updating more clever. There is not enough money in the business to make it a trillion dollars clever. Engineers tend to think ideas will triumph over money. But actually, ideas will work for either side, and the other side has more money.

Partitioned updating is an old technique, not necessarily "clever", and not easily defeated. Updating the BIOS on a general purpose computer can be designed to require user intervention, or to require hands-on access. Samsung TVs from 2012 and 2013 were the target of a hands-on firmware hack by the CIA to make the devices appear to be off when actually on [37]. It would seem impractical to

marshal large numbers of devices for a DDOS attack in this manner, and it is limited to targeted intelligence gathering like old-fashioned wiretapping. A TV is somewhat unusual as an IoT device in that it does have a power off mode which is frequently used, whereas many IoT devices are always on. Partitioning the power control and battery charging into a separate processor, adding some small cost, and making it non-updatable, would add to security, but is generally less applicable for IoT.

Perhaps a larger issue is the use of cloud services such as voice recognition [38], not only for smart TVs but general home automation devices such as Amazon Echo and Google Home. Some devices send all conversation to cloud servers for analysis, even to find activation phrases, and some companies record all conversation, presumably as a base for improving voice recognition. Such data is not only obtainable by secret court order, but may be searchable by intelligence agency data consolidation tools. The data is not deemed to have been "collected" if merely searchable, and if a search hit to a legitimate intelligence question is registered, then use of the data may be considered legitimate. Or a foreign intelligence agency searching the data may not be obligated to follow any law that the user or manufacturer is expecting. The uses of such data are far broader than theft of financial information for direct gain, or gathering of evidence of a crime. An eavesdropper, whether government or private, may simply wish to exploit a target to some other end, such as to gain knowledge of the targets customer facilities or accounts, or find a weakness that may be exploited to gain the target's cooperation.

IoT sensors are varied and new types not currently anticipated will be added. The impact of making this information searchable and analyzable is not yet known. Partitioned update might be useful to monitor something like checksums of updatable code to detect intrusion, but simple off/on functions may not apply, and intelligent analysis of the gathered data to see if it contains unauthorized information and is being handled properly may require the use of sophisticated artificial intelligence functions and cloud resources that are themselves sources of vulnerability.

4. Conclusions

We conclude that the greatest threats to the industry long term health may not be threats targeting IoT devices and users directly as researchers have expected, but threats which use the IoT as a tool to attack other things. DDOS attacks against IoT systems are not the ones that have caused great damage, but the ones using IoT systems. It is not as important how robust IoT is to DDOS as is its susceptibility to being used as a tool. IoT has been the first weapon of mass digital destruction ("digital nuclear attack" [39]) in the new age of cyber war, because the updating feature, used primarily to load new security features to guard against the non-existent threat of DDOS against the IoT itself, made it vulnerable to hijacking. The best way to prevent hijacking is not to allow updating.

Secure updating is a red herring, because the national security apparatus that

will demand a back door is also one of the worst if not the worst actor in both the hijack scenario and the data privacy scenario. This paper has addressed only the hijack scenario, not data privacy, but the principles and analysis introduced can be used for a wide variety of problems.

Two new techniques have been identified (though not extensively analyzed) as a result of our analysis. These are manufacturer (or other service provider) remote monitoring of updated IoT devices in some secure manner, and partitioned update where security critical functions are not vulnerable. These do not completely solve the problem and further work and/or new techniques are needed.

A larger objective was to familiarize the consumer and industrial electronics community with the expanding world of crash rate theory, so that economic-driven behavior can be analyzed and designed-around like an engineering problem. However, additional work is needed identifying classes of actors and quantifying their investments and rewards.

References

[1] Shuler, R.L. (2015) Optimization of Innovation and Calamity. *International Journal of Engineering Innovations and Research*, **4**, 50-56.

[2] Shuler, R.L. (2016) Economic Optimization of Innovation and Risk. Robert Shuler, Vergne, TN.

[3] Shuler, R.L. (2015) Wealth Inhomogeneity Applied to Crash Rate Theory. *Heliyon,* **1**, e00041. https://doi.org/10.1016/j.heliyon.2015.e00041

[4] Gan, G., Lu, Z. and Jiang, J. (2011) Internet of Things Security Analysis. *IEEE International Conference on Internet Technology and Applications*, Chengdu, 16-18 August 2011, 1-4. https://doi.org/10.1109/itap.2011.6006307

[5] Suo, H., Wan, J., Zou, C. and Liu, J. (2012) Security in the Internet of Things: A Review. *Computer Science and Electronics Engineering* (*ICCSEE*), Hangzhou, 23-25 March 2012, 648-651.

[6] Zhao, K. and Ge, L. (2013) A Survey on the "Internet of Things Security". 9*th International Conference on Computational Intelligence and Security*, Leshan, 14-15 December 2013, 663-667.

[7] Lu, N. and Shen, X. (2014) Connected Vehicles: Solutions and Challenges. *IEEE Internet of Things Journal*, **1**, 289-299. https://doi.org/10.1109/JIOT.2014.2327587

[8] Chen, S., Xu, H., Liu, D., Hu, B. and Wang, H. (2014) A Vision of IoT: Applications, Challenges, and Opportunities with China Perspective. *IEEE Internet of Things Journal*, **1**, 349-359. https://doi.org/10.1109/JIOT.2014.2337336

[9] Gisdakis, S., Giannetsos, T. and Papadimitratos, P. (2016) Security, Privacy, and Incentive Provision for Mobile Crowd Sensing Systems. *IEEE Internet of Things Journal*, **3**, 839-853. https://doi.org/10.1109/JIOT.2016.2560768

[10] Singh, J., Pasquier, T., Bacon, J., Ko, H. and Eyers, D. (2015) Twenty Security Considerations for Cloud-Supported Internet of Things. *IEEE Internet of Things Journal*, **3**, 269-284. https://doi.org/10.1109/JIOT.2015.2460333

[11] Staff (2015) Gartner Says 6.4 Billion Connected 'Things' Will Be in Use in 2016, Up 30 Percent from 2015. Gartner, Inc., ITxpo, Barcelona. http://www.gartner.com/newsroom/id/3165317

[12] Goldman, D. (2012) Major Banks Hit with Biggest Cyber Attacks in History. CNN

Money. http://money.cnn.com/2012/09/27/technology/bank-cyberattacks/

[13] Prince, M. (2013) The DDoS That Almost Broke the Internet. *CloudFlare Blog*. https://blog.cloudflare.com/the-ddos-that-almost-broke-the-internet/

[14] Staff (2015) Hong Kong Banks Targeted by DDoS Attacks, Bitcoin Payout Demanded. *DDoS Attacks*. http://www.ddosattacks.net/hong-kong-banks-targeted-by-ddos-attacks-bitcoin-payout-demanded/

[15] Staff (2014) Hong Kong: Massive DDoS Attacks Continue, Targeting Pro-Democracy News Site. https://advox.globalvoices.org/2014/06/20/hong-kong-massive-ddos-attacks-continue-targeting-pro-democracy-news-site/

[16] Paganini, P. (2014) Largest DDoS Attack Hit PopVote, Hong Kong Democracy Voting Site. http://securityaffairs.co/wordpress/26030/cyber-crime/popvote-largest-ddos-attack.html

[17] Young, J. (2015) Hong Kong Banks Targeted By DDOS Attacks, Bitcoin Payout Demanded. *Bitcoin Magazine*. https://bitcoinmagazine.com/articles/hong-kong-banks-targeted-ddos-attacks-bitcoin-payout-demanded-1431985107

[18] Crosman, P. (2015) Banks Lose Up to $100K/Hour to Shorter, More Intense DDoS Attacks. *American Banker*. http://www.americanbanker.com/news/bank-technology/banks-lose-up-to-100khour-to-shorter-more-intense-ddos-attacks-1073966-1.html

[19] Rawlinson, K. (2014) HP Study Reveals 70 Percent of Internet of Things Devices Vulnerable to Attack. *HP Advisory*. http://www8.hp.com/us/en/hp-news/press-release.html?id=1744676#.WEZYAtRU5_k

[20] Shaw, A. (2016) How Podesta Got Hacked: 'Password' Email Revealed in WikiLeaks Dump. *Fox News*. http://www.foxnews.com/politics/2016/10/29/how-podesta-got-hacked-password-email-revealed-in-wikileaks-dump.html

[21] Krebs, B. (2016) KrebsOnSecurity Hit With Record DDoS. *KrebsOnSecurity*. https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/

[22] Greene, T. (2016) Security Blogger Krebs Says IoT DDoS Attack Was Payback for a Blog. *InfoWorld*. http://www.infoworld.com/article/3124753/security/security-blogger-krebs-says-iot-ddos-attack-was-payback-for-a-blog.html

[23] Vijayan, J. (2016) IoT DDoS Attack Code Released. *Information Week Darkreading*. http://www.darkreading.com/denial-of-service-attacks/iot-ddos-attack-code-released-/d/d-id/1327086

[24] Lee, C.E. and Paletta, D. (2016) White House Vows 'Proportional' Response for Russian DNC Hack. *The Wall Street Journal*. http://www.wsj.com/articles/white-house-vows-proportional-response-for-russian-dnc-hack-1476220192

[25] Woolf, N. (2016) DDoS Attack That Disrupted Internet Was Largest of Its Kind in History, Experts Say. *The Guardian*. https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet

[26] Staff (2016) Distributed Denial of Service Attack against Domain Name Service Host Highlights Vulnerability of "Internet of Things" Devices. https://publicintelligence.net/fbi-iot-ddos/

[27] Holmes, D. (2016) What's the Fix for IoT DDoS Attacks? *SecurityWeek*. http://www.securityweek.com/whats-fix-iot-ddos-attacks

[28] Martin, A.J. (2016) Bookmakers William Hill under Siege from DDoS Internet Flood. *The Register*. http://www.theregister.co.uk/2016/11/02/william_hill_ddos/

[29] Staff (2016) Russian Banks Hit by Cyber-Attack. *BBC News*. http://www.bbc.com/news/technology-37941216

[30] Julian, T. (2014) Defining Moments in the History of Cyber-Security and the Rise of Incident Response. *Infosecurity Magazine*. http://www.infosecurity-magazine.com/opinions/the-history-of-cybersecurity/

[31] McMillan, R. (2014) The Inside Story of Mt. Gox, Bitcoin's $460 Million Disaster. *Wired*. https://www.wired.com/2014/03/bitcoin-exchange/

[32] Zetter, K. (2014) An Unprecedented Look at Stuxnet, the World's First Digital Weapon. *Wired*. https://www.wired.com/2014/11/countdown-to-zero-day-stuxnet/

[33] Bradley, R. (2016) Tesla Autopilot. *MIT Technology Review*. https://www.technologyreview.com/s/600772/10-breakthrough-technologies-2016-tesla-autopilot/

[34] Helfmeier, C., Boit, C., Nedospasov, D. and Seifert, J. (2013) Cloning Physically Unclonable Functions. 2013 *IEEE International Symposium on Hardware-Oriented Security and Trust* (*HOST*), Austin, 2-3 June 2013, 176-179. https://doi.org/10.1109/HST.2013.6581556

[35] Francheschi-Bicchierai, L. (2013) 2 Encrypted Email Services Shut Down to Avoid NSA Snooping. *Mashable*. http://mashable.com/2013/08/09/silent-circle-lavabit-shut-down-to-avoid-nsa-snooping/#40nQ0R2jgkqZ

[36] Triggs, R. (2016) What Caused the Great Galaxy Note 7 Defect? Here Are the Leading Theories. *Android Authority*. http://www.androidauthority.com/galaxy-note-7-defect-causes-721528/

[37] Calore, M. (2017) Worried the CIA Hacked Your Samsung TV? Here's How to Tell. *Wired*. https://www.wired.com/2017/03/worried-cia-hacked-samsung-tv-heres-tell/

[38] Young, C. (2016) Smart TVs Pose Huge Security Risks. *BetaNews*. https://betanews.com/2016/01/20/smart-tvs-pose-huge-security-risks/

[39] Kall, R. (2016) DDOS Attack Using Internet of Things on Major Sites Is a Digital Nuclear Attack. *The Huffington Post*. http://www.huffingtonpost.com/rob-kall/ddos-attack-using-internet_b_12600828.html