Reaction: The Myth of Cybersecurity

Divya T Aradhya

Table of Contents

Abstract

The April 8, 2017 U.S. print issue of The Economist revolved around exploring “Why computers will never be safe”. Two of the published articles are titled “The myth of cyber-security” (under their “Leaders” section of the magazine) and “Why everything is hackable” (under the “Science and technology” section). This paper is study of these articles and a reaction to the points they raise.

The myth of cyber-security

Summary

The article builds a case around the premise that computers are inherently not secure and the risks posed by them need to be managed through "economics rather than technology" ("The Myth of Cyber-Security").

Response

In an attempt to highlight that "computers will never be secure" the article draws focus to recent data breach and cybercrimes, including the one with Yahoo! and the Bangladesh bank. It also recognizes that the reasons for software to never be fail safe vary from the fact the many organizations do not security seriously, that software is increasing in complexity, and that security wasn't built into software ground-up.

I agree with this analysis, as software is mostly developed to build something and to do something, and rarely, if ever, to ensure that it is not broken, and that it doesn't do what it should not.

The article further states that the Government can bring in regulations which mandate, regulate, and oblige companies to follow certain minimum safety measure and guidelines for the "public health" of computing ("The Myth of Cyber-Security"). It also draws conclusions from the developments in the 1960s when the public demand for safer cars led to the government enforcing rules for safety belts and headrests. The article makes a doomsday prediction that the same clamor for computing safety will follow with the "first child fatality involving self-driving cars" ("The Myth of Cyber-Security").

While this is a hard-hitting, and perhaps even a calculatedly provocative statement, aimed at gaining attention, it is, I think, unfortunately and alarmingly true. With the looming reality of the

Internet of Things, of their shocking lack of security, the prediction may as well be tragically prophetic.

The article emphasis that the solution to accepting the risks that software pose, even while allowing software makers the freedom of creativity and innovation, lies in the realm of economics. Cybersecurity insurance is stated to be a way of protecting consumers. An organization whose products are "repeatedly hacked" ("The Myth of Cyber-Security") will be subjected to increasing premiums, while a firm that "takes reasonable steps to make things safe" ("The Myth of Cyber-Security") but still compromised, may have a recourse to an insurance payout that prevents it from going bankrupt.

I agree with the idea that bringing economics is the picture can help alleviate the issues of the grave risks of inadequate software security. A firm who has profits at stake and the possibility of bankruptcy will be forced to take security seriously and weave it into their software right from scratch, and not as an afterthought. It also holds companies accountable for the products they are creating and selling, and the possible consequences of their failure to be secure.

The Sarbanes-Oxley Act is one such example. Non-compliance results in steep fines and criminal penalties, and these in turn act as effective deterrents. In the fifteen years since its enactment, lawyers and accountants vouch for its success in making the business space secure and accountable, minimizing negligence, improving reliability, and protecting the consumers and restoring their confidence ("Analysis: A decade on, is Sarbanes-Oxley working?").

Why everything is hackable

Summary

The article discusses the inevitability of every software being susceptible to being hacked, the reasons for this inherent vulnerability, possible solutions around this problem, including one from the economic world: cyber-insurance ("Why everything is hackable").

Response

The article does a good job of elucidating various hacks in the recent past, and identifying the hackers and motives behind them. It mentions the British high-school student who hacked into the POS systems because he "could", the Bangladesh bank heist by cyber criminals, the hacks into the NSA and CIA database by powerful hacking groups, possible state-sponsored, the hacktivist hacks into the National Committee e-mail servers in order to cause embarrassment. The reasons for hacking are varied, as are the actors behind the hacks – but a round-up of the different hacks emphasis the seeming inevitability that all software systems will be eventually hacked.

The article also talks about how with the growth in technology, especially in the realm of the emerging "Internet of Things", hacks can only continue to grow in intensity and number, as it did with the Mirai botnet hack attack. I agree with this statement, as the Internet of Things seem to pushed out products into the market without a matured project model and with no foresight of possible security risks, as was seen with the German talking doll, Cayla, that could potential violate a child's privacy ("Germany bans talking doll Cayla, citing security risk").

The article next addresses the reasons for software being prone to hacks. First, it recognizes that software is increasingly complex, spilling to billions of lines of code - making it hard to test, maintain, and be fail-proof.

Second, software is built by putting together disparate modules, each built of different teams, sometimes even in different countries. A flaw introduced (deliberately or accidently) is almost impossible to trace, but definitely possible to exploit.

Third, most of today's software systems are still based off legacy code, written at a time when security wasn't considered in any phase of the software development lifecycle. Any security that the system may have was added as an after-thought and not an integral part of the software itself.

Fourth, the software industry is constantly rolling out new release, new features, and sending code into production with "agile" cycles, where security is not a concern, but short-term functionality is. The organizations are focused on immediate gains and do not strategize on long-term plans, which should include risk assessments, and the consequences of security breaches.

Fifth, software is packaged with long, verbose legalese that most users don't read or understand, but effectively prevent the software firm from being liable and not take responsibility or be accountable for security lapses and flaws in the software.

Sixth, governments are suspicious of encryption, and truly secure encryptions are not encouraged either in research or implementation.

I completely agree with this detailed analysis and the points highlighting why secure software is not a reality in today's world.

In an attempt to explore possible solutions to these problems, the article next talks about a small proof of concept that was presented by Dr. Watson of the University of Cambridge, who designed a new type of chip that "attempts to bake security into hardware" ("Why Everything Is Hackable" p.71). Dr. Fisher, another security researcher, developed a flight-control software using a technique called "formal methods" and publically exposed every line of the source code to a team of hackers. It was found that the system could still not be hacked. She admits, though,

that “it will be a long time” before the methodology and result can be replicated on a full-fledged operating system. (“Why Everything Is Hackable” p.71).

While research in the security field is vital to come up with new ways of generating fail-proof design and code, some of the problems to do with security lies in the mindset and end goals. Immediate profits always seem to have higher persuasion powers than the fear of a possible loss due to security lapses.

Next, the article explores the realm of cyber-insurance, in an attempt to find a solution to the risks posed by software. It admits, however, that software development companies, who mostly have a “libertarian streak” will put up stiff resistance over any attempts to hold them accountable and impose liability. (“Why Everything Is Hackable” p.71).

The article is right in recognizing the mindset of the software development world – that in the name of innovation and complete freedom they also shirk responsibility towards the damages caused by software, released without matured security measures. The solution would perhaps lie in every stakeholder being tuned into working towards more secure systems. Researchers, developers, governments, and yes, even end users, who each take responsibility for what they propose, design, produce, regulate, and consume.

References

The Myth of Cyber-Security; Computer Security. (2017, April 8). The Economist (US), 9-9

Why Everything Is Hackable; Computer Security. (2017, April 8). The Economist (US), 69-71.

Drawbaugh, K., & Aubin, D. (2012, July 30). Analysis: A decade on, is Sarbanes-Oxley working? Retrieved from http://www.reuters.com/article/us-financial-sarbox-idUSBRE86Q1BY20120730

Germany bans talking doll Cayla, citing security risk. (2017, February 17). Retrieved from http://www.reuters.com/article/us-germany-cyber-dolls-idUSKBN15W20Q