Effects of software security vulnerabilities on the software development industry

Jensen Sebast-Epiktos, PhD

Abstract: The security and reliability of the software development process and products have been under considerable scrutiny for some time now. Technically, software needs to behave as expected with minimal software failures even in the presence of a malicious attack regardless of whether it is produced in a competitive market or not. Regrettably, some data breaches and cyberattacks have been attributed to software vulnerabilities, which in most cases can be consequential if proactive measures are not taken by information systems (IS) leaders during the development process. It is therefore incumbent upon IS executives to understand potential threats to software development prior to considering enterprise software adoption.

A review of "The impact of malicious agents on the enterprise software industry". By Galbreth, M. R., & Shor, M. (2010). *Mis Quarterly*, 595-612.

Summary: "In this paper, a competitive software market that includes horizontal and quality differentiation, as well as a negative network effect driven by the presence of malicious agents, is modeled. Software products with larger installed bases, and therefore more potential computers to attack, present more appealing targets for malicious agents. One finding is that software firms may profit from increased malicious activity. Software products in a more competitive market are less likely to invest in security, while monopolistic or niche products are likely to be more secure from malicious attack. The results provide insights for IS managers considering enterprise software adoption."

***Keywords**: Software development; software security vulnerabilities; software reliability; security tools and testing; security baselines*

The rapid growth of the software industry continues to pose security threats and attract targeted attacks by cybercriminals. The security and reliability of software products have been under considerable scrutiny for some time now as a result of the prevailing threats. Some common types of software security vulnerabilities include injection flaws, broken authentication, cross-site scripting (XSS), cross-site request forgery (CSRF), bugs, buffer overflow, security misconfiguration, and broken access control among others. It is therefore imperative that the software industry produce tamper-proof software by employing secure practices and techniques such as a code signing certificate and enforcing security standards during the development process to prevent software security vulnerabilities that pose threats to users. Technically speaking, Galbreth and Shor [1] argue that software needs to behave as expected with minimal software failures even in the presence of a malicious attack regardless of whether it is produced in a competitive market or not. Unfortunately, some data breaches and cyberattacks have been attributed to software vulnerabilities, which in most cases can be consequential if proactive measures are not taken by information systems (IS) leaders during the development process. It is therefore incumbent upon IS executives to understand potential threats to their software and the bottom line prior to considering enterprise software adoption.

While the dynamics of technology innovations today continue to evolve at a rapid pace with an accelerating rate of change, they have also created unprecedented opportunities for attackers. According to Galbreth and Shor [1], by "breaching the security of enterprise systems, malicious agents can cause significant financial loss and other negative consequences". The author believes that unsecured products would attract lower prices and hence provide incentives for software firms to minimize software vulnerabilities. Thus, the main focus of their study was to propose a model of competition surrounding malicious agents in terms of the relationship between market share and security. Stated differently, the authors were motivated to model the enterprise software selection process as a means to address these challenges.

Table 1 of the paper summarizes the relationship between the competitive environment and the incentives of software firms to minimize the vulnerabilities of their products. In addition, their model provides an understanding of the impact of malicious agents on the industries they target, though they acknowledged some limitations. The authors concluded that firms in competitive markets are likely to have less incentive to encourage patching compliance for more secure software products. However, if the impact of the market on software security would streamline the behavioral study of malicious agent incentives, what role does the level of competition play in this case?

The primary motivation of threat actors and malicious agents has seen a significant trend toward attacks with prestige, financial, political, or military goals [1]. Chess and Arkin [4] noted, over the past decade, software security has matured from a niche topic studied by academics to a vital part of how software is built. It makes sense to argue that security in the software development life cycle is necessary but not sufficient as noted by Chess and Arkin.

Recent research [2]-[4], [5], [6]-[7] has addressed these growing software security threats in a variety of ways, including mitigating the complexity of threats, sophisticated attacks, and third-party flaws. In this regard, a few studies have examined the impact of baselines and internal benchmarks. For example, Rotella believed that without historical baselines and internal, it becomes a challenge to reliably assess how well the software development teams can control the level of security vulnerabilities in the organization's commercial software products [2]. Of particular interest is understanding how the extent of software vulnerabilities emanates from the open-source software that is used in commercial products. From the perspective of sophisticated attacks, Parizi et al [3], noted that producing reliable and secure software is difficult because of its growing complexity and the increasing number of sophisticated attacks. The author presented an initial set of requirements for proper benchmarking used in the assessment of security vulnerability testing (SVT) and detection tools. In addition, Shin and Williams [7] performed statistical analysis on nine code complexity metrics from the JavaScript engine in the Mozilla application framework to investigate the validity of this hypothesis. Of course, among the various aspects of cybersecurity, software security testing plays a critical role. Third-party management flaws can also be counter-productive to software development. Known security vulnerabilities can be introduced in software systems due to third-party involvement [5].

Overall, these findings clearly impact the development, marketability, and usability of the software products produced by vendors. While the study [1] has made important contributions to addressing the threat of malicious attacks in terms of not only safeguarding software firms but also benefiting their clients and consumers, more areas remain to be explored. Upon examining the impact of malicious activity on the software industry, the authors asserted that the outcome is an important consideration for IS executives. The reason was attributed to the fact that as software firms continue to invest in minimizing

software vulnerabilities, the value of their products increases, which in turn improves customer satisfaction and hence makes organizations more profitable.

The following questions are worth considering in our quest for lasting solutions:

- The question is how do malicious agents impact the market, as demand and supply, in general, are inversely related?
- How does enabling a firm with inferior software products to operate profitably impact the software development industry?
- Can software firms profit from increased malicious activity of their products?
- Why the level of vulnerability to malicious activity is preferred by software firms in highly competitive industries?

To unpack these questions, Galbreth and Shor [1] argued that the intense competition among software vendors promotes less incentive to address vulnerability to malicious attacks unlike the more monopolistic environments in a traditional sense. However, how do firms that operate in competitive environments tend to weaken price competition? One would be tempted to believe that the risk of these practices raises serious privacy and security concerns for individual users and customers of their products. The fact that malicious software updates can raise considerable cybersecurity concerns that could have serious effects on final software projects, security policy, and compliance enforcement cannot be ignored, at least not now. Certainly, addressing software vulnerabilities within the confines of the competitive market requires effective policy and compliance requirements, as well as the public trust of software vendors. Out of the numerous solutions proposed, two of the most effective ways to make software tamper-proof are by employing a code signing certificate and enforcing security standards during the development process to prevent software security vulnerabilities that pose threats to users.

References:

[1] Galbreth, M. R., & Shor, M. (2010). The impact of malicious agents on the enterprise software industry. *Mis Quarterly*, 595-612.

[2] Rotella, P. (2018, May). Software security vulnerabilities: baselining and benchmarking. In *Proceedings of the 1st International Workshop on Security Awareness from Design to Deployment* (pp. 3-10).

[3] Parizi, R. M., Qian, K., Shahriar, H., Wu, F., & Tao, L. (2018). Benchmark requirements for assessing software security vulnerability testing tools. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)* (Vol. 1, pp. 825-826). IEEE.

[4] Chess, B., & Arkin, B. (2011). Software security in practice. *IEEE Security & Privacy*, *9*(2), 89-92.

[5] Cadariu, M., Bouwers, E., Visser, J., & van Deursen, A. (2015). Tracking known security vulnerabilities in proprietary software systems. In *2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER)* (pp. 516-519). IEEE.

[6] Richet, J. L. (2022). How cybercriminal communities grow and change: An investigation of ad-fraud communities. *Technological Forecasting and Social Change*, *174*, 121282.

[7] Shin, Y., & Williams, L. (2008). Is complexity really the enemy of software security? In *Proceedings of the 4th ACM workshop on Quality of protection* (pp. 47-50).