Department of Information Systems & Technology Management – ISTM 6206

Assignment #02 (15% of Total Grade)

Introduction

Apple is an American corporation that develops and sells computer electronics, software, personal computers, and portable devices internationally. Founders Steve Jobs, Steve Wozniak, and Ronald Wayne established Apple in 1976, with its incorporation in 1977. Apple has a history that spans over 30 years, and during that time the company has experienced its ups and downs in financial performance. It was after 2007 when Apple finally achieved widespread success with the launches of the iPhone, the iPod touch, and the iPad. Recently, Apple began incorporating the use of open-source software into its product lines for a variety of reasons, including accelerating the development of its product lines and enhancing its cyber security through the combined efforts of knowledgeable programmers.

One of its business concepts, which underpins its firm conviction in the communal work of the open-source software community, is “Open-source software is at the heart of Apple platforms and developer tools. Apple works with developers around the world to create, contribute, and release open-source code.” (*Apple Open Source*, n.d.) Apple’s featured open-source projects are Swift, Kubernetes, and Web Kits.

This paper will look at the potential risks it faces, and the risk management options that might materially aid the business in preventing system compromise.

Risk Assessment and Analysis

The Apple Company is a high-tech business that specializes in creating software, electronics, and smart phones. They have released a large number of product lines that are regarded as open-source software programs. I will look at the threats that could expose the Apple company to some risk associated with the use of open-source software in its product lines in this section. These dangers have a range of repercussions that could affect the company's standing in the marketplace, competitiveness, and financial health. So, in the end, I will suggest some possible strategies to deal with these risks.

Risk 1: Exploitable code

Swift is a general-purpose programming language built using a modern approach to safety, performance, and software design patterns.

(*Swift*, n.d.)The Swift project seeks to develop the best language currently obtainable for a variety of applications, including system programming, mobile and desktop apps, and cloud services. Swift was designed primarily to make it easier for developers to write and maintain correct programs. We think that the most obvious way to write Swift code must also be safe, quick, and expressive in order to accomplish this. Because of this, Swift became open source in 2015. Swift is more likely to be targeted with exploitable vulnerability code due to the large number of contributors, which include thousands of developers and hundreds of companies, who are dedicated to developing and writing millions of lines of code for modification. The overwrite vulnerability resulted in a potential security breach at the Apple company.

Risk 2: Publicity of Exploits

Open-source flaws are made accessible to the public on sites like Foundation DB, which is open to all Apple users. Criminals might exploit that information and use it to their advantage. For instance, the significant Equifax hack in 2017—during which the credit reporting firm exposed the personal data of 143 million people—is a well-known illustration of attacks caused by open-source vulnerabilities that are readily accessible to the public. This incident happened as a result of Equifax utilizing a high-risk, vulnerable version of the open-source Apache Struts framework, which was exploited by the attackers. Such attacks on open source software not only result in data loss or leakage but also have an effect on a company's market standing, stock price, and clientele. This could consequently have an effect on your client churn rate, retention rate, sales, and revenue. Dealing with the consequences of a breach Dealing with the impact of a breach caused by open source vulnerabilities can be a lengthy and painful process. (Cure, 2015)

Risk 3: License violation

Developers may violate license terms. Because open-source software is subject to more than 200 different types of licenses, for example, the GPL, Apache, MIT, BSD, and Unlicensed, when using open-source software, developers have to comply with all the terms of the licenses applicable to the open-source software components they are using. The issue with this is that many of these licenses are incompatible with each other. The more open-source software components a developer uses, the more difficult it is for them to track and compare all of the license requirements. Developers may violate Apple's license requirements. (Fichtner, 2022)

Risk 4: Lack of security

(*Contrast Security*, n.d.) For two reasons, it is challenging to comprehend OSS security in its entirety: OSS is dispersed in nature by design. As a result, there is no central organization in charge of ensuring quality and upkeep. What kinds of OSS are used the most frequently is unknown because they can be freely copied and updated.

In essence, OSS and custom code are equally secure. However, it may contain bugs that cause security problems, just like any other piece of software. Security researchers can manually review the code to find these vulnerabilities because OSS is freely accessible. As a result, Apple may find thousands of new vulnerabilities each year and publicly publicize them frequently, with exploits being used to demonstrate the vulnerability's existence.

Risk 5: Software Security Risks

When open-source vulnerabilities are found, attackers may be tempted to use them as targets. Usually, information regarding these open-source flaws and how to exploit them is made available to the public. This gives hackers access to all the information they need to launch an attack. When you consider this and the widespread use of open-source software, you can see the mayhem that can result when an open-source vulnerability is discovered. (Cure, 2015)

The fact that tracking open-source vulnerabilities and their solutions is more difficult than one might think is one of the biggest problems that Apple encounters. It only takes a short while for attackers to use an open-source vulnerability and its path of exploitation to break into the Apple company. Apple must implement the required procedures and technologies to address open-source vulnerabilities.

Risk 6: Distribution

The distribution channel is another typical open-source software security flaw. Most IT teams won't perform any additional verification if the code is published as a binary, other than seeing if the provided hashes match the binary. But since the binary and hashes frequently originate from the same place, an attacker can compromise either one of them. This occurred recently when a number of Linux distribution repositories were compromised. (Compact Magazine, 2020) So, this can happen with Apple's open-source project like Swift.

Risk management for open-source software

The aforementioned risks for open-source software can be reduced using a number of risk management techniques. A cloud-based security solution that regularly does vulnerability penetrating testing on databases, software applications, and source code files to look for any potentially exploitable insecure code is one of the scanning tools that are efficient to carry out such scanning. It will also save time and aid in spotting harmful code injections in open-source code files to stop them from changing the deployment environment if a plan is made to do a regular penetration test through cloud services. The top scanning tools are the open-source risk engines SimpleRisk, Eramba, etc.

Conclusion:

After researching numerous OSS risks, I believe that Apple should use OSS for the benefit of the business, owing to its faster rate of innovation, strong community support, and lack of vendor lock-in. For many, open source involves more than just code. It is symbolic of a way of life. A community that uses open source software celebrates and promotes the notion that openness, reciprocity, sharing, and collaboration result in the creation of better software—and perhaps even a better society. (Tennant, 2022)

According to an article, "Why Apple is wooing open source developers with Swift," by Cliff Saran (Managing Director) wrote this. “Apple is beginning to embrace open source, according to Chris Wanstrath, CEO and co-founder of the open source code repository GitHub, who was speaking to attendees at the European GitHub conference in Amsterdam. He notably cited Apple's work on GitHub as a shining example of its innovative working methods. "Open source is much more than just a philosophy." It is an excellent method for creating software. Companies profit from open source, he claimed.

With its new programming language, Swift, Apple has taken a fresh attitude toward the open source community, according to Wanstrath: "Apple's Swift is a new open source initiative." It is an excellent open-source project that supports an excellent community and establishes excellent open-source standards.

References:

http://www.sfu.ca/~sheppard/478/syn/1123/Synopsis5.pdf

Apple Open Source. (n.d.). https://opensource.apple.com

Cure, A. (2015, February 5). *C#/.NET/Core Training in Denver, CO – May 2019*. https://www.cypressdatadefense.com/blog/open-source-security-risk/

Swift. (n.d.). Swift.org. https://www.swift.org/about/

Fichtner, E. (2022, May 3). *7 Risks Posed by Open-Source Software and How to Defend Yourself*. Datto. https://www.datto.com/uk/blog/7-risks-posed-by-open-source-software-and-how-to-defend-yourself

Tennant, D. (2022, August 17). *How Open Source Software Benefits Businesses*. The Agile Content Platform WordPress VIP. https://wpvip.com/2022/05/12/how-open-source-software-benefits-businesses/

Contrast security. (n.d.). Understanding the Risk of Open Source Software. https://www.contrastsecurity.com/hubfs/Understanding-the-Risks_WhitePaper_042020_Final.pdf?hsLang=en

Saran, C. (2016, May 20). *Why Apple is wooing open source developers with Swift*.ComputerWeekly.com. https://www.computerweekly.com/news/450296755/Why-Apple-is-wooing-open-source-developers-with-Swift

Compact Magazine. (2020, April 23). *The risks of open-source software for corporate use*. Compact. https://www.compact.nl/en/articles/the-risks-of-open-source-software-for-corporate-use/

https://www.goodfirms.co/risk-management-software/blog/best-free-open-source-risk-management-software