Software Security

CSC 424 Software Engineering II

Cody Ezell

3/23/2021

Introduction

Software surrounds us in our everyday life. From our iPhones, to our computers, to the Alexa or Google assists scattered across our homes. Software is used everyday by many people to enhance our productivity and lifestyles. When critics state "There's an app for that," nine times out of ten they are correct. So why should we be scared of it? What could possibly go wrong? Sadly, software is not only an assistant to our everyday lives, but is also a doorway to our personal lives and sensitive data. Hackers can track our every move and know every minute detail about our lives, from our name and birthdate to our social security numbers. Hackers can track our location and spy on us in our private lives. If this is the case, why do we rely on software to begin with? Luckily, this is where software security comes into play. Yes, hackers can track us and dig into our very personal data, but through software security we can alleviate the possibility of hackers accessing this data. Software security is a measure that is put into the development process to seal back doors and block hackers from accessing data they have no business accessing. Not only that, but software security also provides data integrity and helps to prevent companies from releasing data or products that have not been debuted to the public quite yet. Mainly, software security protects developers from malicious attacks and secures code from being leaked or stolen, but in terms of a user's point of view, software security extends to protecting the data that is inputted and stored to the software as well. So in detail, what exactly is software security and how does it work? What are some testing tools that developers use to determine if the security implementations work? What can developers do to make code more secure? These questions, and more, are what software security is all about.

What is Secure Software?

What is secure software? To understand what secure software is, we must first understand what software security is and how it makes software secure. According to Techopedia, software security is, "an idea implemented to protect software against malicious attack and other hacker risks so that the software continues to function correctly under such potential risks. Security is necessary to provide integrity, authentication and availability" [1]. Software security extends to a vast spectrum. Software security benefits the developers of a particular software from being maliciously accessed, stolen, or released to the public. Also, it protects the integrity of the software by ensuring that the functionality and data requirements are up to par and are performing optimally. Software security extends to benefitting a company by securing its assets and reputation. Companies also typically have certain levels of requirements that their software must meet, and software security provides Similarly, companies who develop software for their own internal usage require security of their software to protect the sensitive and private data stored within their systems. Extending to the user's standpoint, software security protects their inputted personal data from being spread to undesired locations or being accessed maliciously. To sum up the spectrum, software security contains the factor of application security, or securing data within the app from malicious attacks and hacking, but it also extends to securing the way a particular app is ran and maintained. Developers understand that it is not a matter of if a hack will occur, but rather when a hack will be attempted, and developers use software security to ensure all systems continue to operate as intended and block unwelcome activity. In doing so, we are able to then create secure software. Secure software is software that meets a particular objective of security. If the objective is met, meaning minimal risks to the integrity of the software objective, then the software can be deemed secure.

Testing Tools

When developing software security, it is crucial to be aware of the various types of testing tools available and what tools satisfy the developer's software security objectives. In general, there are 6 critical areas of testing. Those include, authentication, authorization, availability, confidentiality, integrity, and non-repudiation. To test these criteria, companies have designed specialized tools to ensure maximum security in software development. To name a few, Cigniti, a software quality company, have highlighted a few recommendations. One popular tool Cigniti recommends is called Google Nogotofail, and they describe as, "a network traffic security testing tool. It checks applications for known TLS/SSL vulnerabilities and misconfigurations. It scans SSL/TLS encrypted connections and checks whether they are vulnerable to man-in-the-middle (MiTM) attacks. It can be set up as a router, VPN server or proxy server" [2]. Another recommended tool is Iron Wasp. Iron Wasp is a web application vulnerability tester used by many companies worldwide and is described as, "A GUI-based powerful scanning tool which can check over 25 kinds of web vulnerabilities. It can detect false positives and false negatives. It is built on Python and Ruby and generates HTML and RTF reports" [2]. One of the more popular tools used by many companies is ZED Attack Proxy, also known as ZAP. Described by Cigniti, ZAP was, "Developed by AWASP and is available for Windows, Unix/Linux and Macintosh platforms. It has high ease of use. It can be used as a scanner or to intercept a proxy to manually test a webpage. Its key features are traditional and AJAX spiders, Fuzzer, Web socket support and a REST-based API" [2]. With hundreds of available testing tools to take advantage of, it becomes almost impossible to not find a reliable source for testing needs to ensure software security objectives are met. Failure to implement such important testing procedures during development not only creates reputation issues for a particular developer or company but is also extremely dangerous.

Making Code More Secure

It comes as no surprise that companies invest thousands if not millions of dollars in securing software code. Vulnerabilities are inevitable if there is a lack of software security, and this creates a domino effect of issues from the developer, to the company, to the user. But how can you make code more secure using software security methodologies? Rob Lemos, a writer and analyst for TechBeacon, outlines four ways to secure code regardless of the programming language used. Lemos states that to make code more secure, developers should use an environment that suggests secure patterns and reinforces security best practices through notifications in the environment, educate yourself on secure coding by knowing the general design patters to avoid and which functions produce vulnerabilities, use available testing tools to catch vulnerabilities, and automate the security process to speed up the testing process and also ensure optimal testing is performed [3]. Choosing a language and framework that corresponds to the needs of the project will ensure a developer has all the right resources to implement security features in the software, which also helps the developer to adopt secure coding practices. With this in mind, most languages are security-neutral as developers can definitely create code that is secure using most modern languages. Furthermore, education on security should always be on the top of a developer's priority list. Technology evolves rapidly year to year, and with evolving technology comes evolving tactics for data breaches and vulnerabilities. It is just as important to continue education on security of software as it is just as important for doctors and lawyers to continue their education on medicine or updated laws. Finally, using testing tools and automating the testing process

simplifies the process of securing the code and also provides more effective strategies in software security tactics.

Conclusion

Software security has become one of the most important topics in recent years, and has even evolved into its own respective field of study. Ensuring code is secure not only protects the code itself but eliminates the users of the code from being vulnerable to data breaches and tracking. It goes without saying that software is a powerful tool that is used by millions of people every day. With great power, there is also great dangers that can come into play if the software is not properly developed. Ensuring software security is implemented into every piece of work will create less vulnerabilities and will help to not only maintain the users of the code, but also the companies and the developers who designed and released the software. This cycle of benefit continues to empower our technical industry, and future implementations of software security only tightens the gap between attacks.

Sources Cited

[1]. Techopedia. (2011, July 30). What is software security? - definition from Techopedia. Retrieved March 23, 2021, from https://www.techopedia.com/definition/24866/software-security

[2]. Cigniti (2021, April 12). Security testing tools you need to know about. Retrieved March 23, 2021, from https://www.cigniti.com/blog/security-testing-tools-need-know/

[3]. Lemos, R. (2019, November 12). 4 ways to secure your code regardless of programming language. Retrieved May 23, 2021, from https://techbeacon.com/security/4-ways-secure-your-code-regardless-programming-language