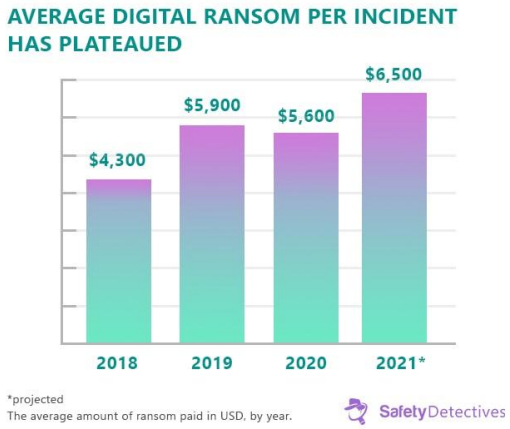
# Software Security

Software Engineering II
CSC424
April 22, 2021

Corey Welborn

Today's society is becoming more and more reliant on software than ever in history. Every type of business in today's market utilizes software in some way or another. These technologies improve the efficiency and longevity of every industry in today's environment. With all these improvements, developers have learned many lessons since the dawn of the Technological Revolution. Malicious attackers have invaded and exploited the products we use every day. With the momentum of innovation, security is an important aspect of developing any product to be used on a mass scale. With the sensitivity of certain types of applications, there is a need to mandate compliance to certain regulations in the realm of software.
[1]

Secure software is defined by Study.com as "software developed or engineered in such a way that its operations and functionalities continue as normal even when subjected to malicious attacks. (1)" The reliability of an infrastructure is paramount when a project is released to the public. Attackers should be identified and dealt with all while the product continues to process user interactions. When a service goes down, users are left to look elsewhere for the service you provided. In order to retain customers, developers should continue to review and update the packages involved in the implementation of your products. Secure software should be built with security in mind, as we develop new products, the best practices should be used in order to prevent malicious attacks in the future.

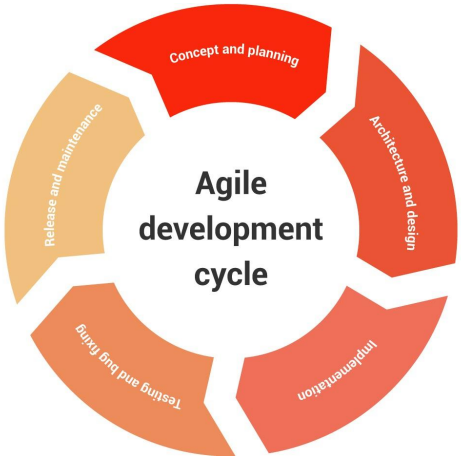



The severity of a security breach has become a driver in the innovation of tools available to test for holes in your design. Many companies have made it their duty to develop a set of easy to use tools to find said vulnerabilities. A few of the most known tools that are open source are Zed Attack Proxy, Wfuzz, Wapiti, and W3af. Zed Attack Proxy (ZAP) was developed by the Open Web Application Security Project. The Zed Attack Proxy has a graphical user interface that allows developers to test a web application during and after project completion. Wfuzz is a tool that only has a command line interface. Developed in Python, it is used to brute-force attack web applications. Wapiti was developed by SourceForge and DevLoop, and uses both GET and POST requests to find vulnerabilities in web applications. W3af is another very popular tool that is open source and developed in Python. This tool tests for over 200 types of security flaws and output can be logged in a console, or sent as a file or email.

According to Positive Technologies (2), a security plan should be involved in all the stages of application development. Planning in advance can have a dramatic effect on time to

[1] Image from SafetyDetectives shows ransomware paid for the past 3 years (1)

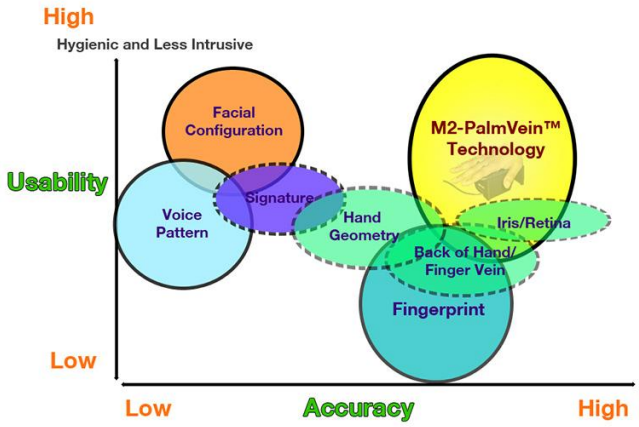
deploy in product development. The Agile development cycle repeats the same stages many times. During these iterations, concept and planning, architecture and design, implementation, testing and bug fixing, release and maintenance, and end of life should have an aspect of security. Thinking about security throughout the process leads to a better execution of secure software.




During the concept and planning phase, the application is evaluated in order to find out the viability of the software. Developers should make a list of security measures that should have special considerations considering the type of application being developed. The architecture and design stage allows programmers to design technologies in a way to complete the requirements of the application at hand. During this stage, it is pivotal to search for ways your application could be infiltrated in the deployment phase. The implementation phase of the Agile development cycle is where the product is created. After coding the product, security tools should be used to find vulnerabilities not thought of in the architecture and design phase. Testing and bug fixing gives a team the time to correct any vulnerabilities found in the previous stages. During release and maintenance, updates are created and deployed in order to maintain security and add new functionality. Also, an incident response plan should be established in order to prepare for any attack that may occur. This step is critical in that in the event of a breach, the services rendered have their best chance at surviving the attack. Lastly, the end of life phase is when a developer no longer supports a software application. Certain types of data have a protocol in place in order to secure sensitive information.

According to the Department of Defense (3), just because an application isn't open source doesn't mean it is more secure. Many attacks do not utilize source or binary code to perform attacks. Hackers observe output from input to probe for vulnerabilities. If the source code is needed, there are decompilers that do a pretty good job at replicating the original source code. This is often enough to allow attackers to understand the underlying infrastructure of the target application. When a project is open source, the code is available for other developers to review and find vulnerabilities. This is an advantage in that said vulnerabilities can be found by people reviewing the code that would have been otherwise overlooked.

Given the major developments in mobile technology in the past 20 years, newer features of these devices have given a new outlook for security of modern technological platforms. Fingerprint scanners, Facial Recognition, and Two Factor Authentication are a few examples of innovative ways to secure resources in an app. Requiring one of these three features leads to a great increase in application security. Users can not simply discover a password and login to another persons accounts. Two Factor Authentication requires the user logging in to also have access to either a phone number or an email address already on file. Fingerprint scanners and facial recognition add to applications a level of security that a password simply cannot. According to Security Magazine, passwords have given users a false sense of security. The implementation of Biometrics not only makes applications with sensitive data more secure by requiring the intended users presence, it also adds to the flow of the application. Users no longer have to remember passwords every time they want to use a service, they simply look at the screen or place their finger on the scanner.[2]




While there is much to learn about the field of security when it pertains to software and application development, we have learned to overcome and adapt when we have fallen short of expectations. The advent of specialized tools to uncover vulnerabilities that would have otherwise been released has given us a second chance in our applications reputation. All the while newer technologies are being developed in order to keep up with the malicious intents of hackers across the globe. While any time is a good time to think about security, it has been shown that security should be a vital part of the design process as well as the maintenance phase. While there is much power in the development of new technologies, there is also a great responsiblility.

[2] Image from Google Biometrics (5)

Works Cited


1. SafetyDetectives. “Ransomware Facts, Trends & Statistics for 2021.” SafetyDetectives. https://www.safetydetectives.com/blog/ransomware-statistics/. Visited on April 22, 2021.
2. Positive Technologies. “How to approach secure software development.” Positive Technologies. https://www.ptsecurity.com/ww-en/analytics/knowledge-base/how-to-approach-secure-software-development/. Accessed on April 22, 2021.
3. Department of Defense. “Open Source Software FAQ.” Department of Defense. https://dodcio.defense.gov/open-source-software-faq/#Q:_Doesn.27t_hiding_source_code_automatically_make_software_more_secure.3F. Accessed April 22, 2021.
4. Security Magazine. “AI and biometrics in 2021.” Security Magazine. https://www.securitymagazine.com/articles/94548-ai-and-biometrics-in-2021-predictions-trends-and-insights-for-what-might-lie-ahead. Accessed on April 22, 2021.
5. Google. “Biometrics” Google. https://sites.google.com/site/biometrics2/graphs-and-data. Accessed April 22, 2021.