GitHub Security Alerts

An Easier Path to Preventing Future Hacks

Gavin H. Smith

COMP 116 Final Project

December 12, 2018

I. Abstract

GitHub is the largest host of source code in the entire world, containing over 57 million repositories. Many of the projects that are hosted on GitHub are written in Ruby or JavaScript, languages commonly used for web development. Most web-based projects require numerous dependencies to function, and these third-party dependencies require constant updating to make sure they minimize security and performance flaws. In this paper I will outline how GitHub uses their Security Alerts scanning feature to detect security vulnerabilities using MITRE's Common Vulnerabilities and Exposures (CVE) list. I will discuss how this alert system should be used by developers to help streamline their security development process, and how taking advantage of this product will be beneficial to organizations in the long-term.

II. Introduction

In today's modern age where many code-based projects are made up of thousands, if not millions, of lines of code, it is almost impossible for a human being to be able to review all of it to check its security. Not only this but many projects are dependent upon many other projects themselves, further adding to the number of ways that a project can be exploited.

Considering that GitHub is the largest host of source code in the world, it would make sense that they would implement something that could help developers keep their code safe and effective. GitHub's new Security Alerts provides an alert system for developers to more easily be able to fix and update their existing code, warning them of the issue with what the issue specifically is as well as the level of severity of the issue. Most issues that are exploited by hackers are not day-zero exploits, but rather known exploits that have already been patched. GitHub putting this security measure into their code hosting site allows developers to more easily

avoid situations where code is not updated or fixed, not requiring a third-party to also review their code as well.

In this paper I will explain how GitHub integrating this service into their code hosting site can prevent major hacks, but also how it can be improved upon in the future to make up for its shortcomings.

III. How Security Alerts Work

Starting in November of 2017 GitHub automatically checks all public repositories and private repositories that opt into the feature that use Ruby and JavaScript for vulnerabilities that are in MITRE's Common Vulnerabilities and Exposures database, which contains a list of most known specific vulnerabilities that have existed in current and previous versions of publicly available software (GitHub)(CVE). Since then they have added support for Java, Python, and .NET. Also, they have begun to install a machine learning model that helps to analyze the public commits and analyzing any vulnerable dependencies that one's code might contain, including vulnerabilities that might not be listed in the CVE database (Bathompso).

GitHub's security alerts work by giving the developer a notification on the dependency that has been flagged to be out of date or contain some found vulnerability. Included in this warning is the CVE identification number associated with the specific vulnerability that was found in that dependency, as well as how serious a threat this vulnerability poses to any project that is currently using that dependency. The intention is so that developers have a much easier time tracking what parts of their code need to be fixed and giving them a head start towards fixing the issues. Since GitHub does not need to wait until projects are finished or late in their development cycle to alerts developers of the issues in the dependencies that they are using, the

alerts can create a much easier pathway for developers to fix bugs early, avoiding having to change or remove significant portions of a projects later in the development cycle. In the first month after GitHub launched this feature in November of 2017 it claimed that 450,000 vulnerabilities had been resolved (Rashid). Even though this is only about ten percent of vulnerabilities that were caught by the alerts, this is still a great stride in the right direction towards better computer security.

IV. Security Alerts and Avoiding Hacks

Having this kind of technology be more widespread in the past could have helped to mitigate or entirely avoid some of the largest hacks that have ever occurred. In the summer of 2017 Equifax, one of the three largest credit agencies that is responsible for collecting information of over 800 million people, was hacked because of a vulnerability in an unpatched version of Apache Structs, a Java framework, that allowed for remote code execution which stole over 148 million people's private information (nixawk). Then in December of 2018 the United States House Oversight and Government Reform Committee found that this hack was completely avoidable and was only due to negligence on the part of Equifax employees (United States House).

GitHub security alerts may not have caught this exact issue, as it was found that Equifax knew about the issue from the Department of Homeland Security and they just failed to implement a fix for all of their services, but having another way for companies and developers to more easily see exactly where each of these vulnerabilities exists in their projects would have definitely decreased the chances of this happening. As well as this is a good example to show that a vulnerability that only consists of one product having a not updated dependency can lead to a very serious hack that affects hundreds of millions of people.

V. Improvements going forward

i. GitHub

While GitHub's security alerts are a step in the right direction towards removing simple errors in developers code and streamlining the process of debugging code, they are nowhere near a permanent solution to these issues. First, the way that these alerts work is that they put the full agency of fixing the bugs on the developers. While this may be a fine strategy for smaller developers who are able to fix these issues more easily and push the updated versions to production, it does almost nothing for larger corporations who more often can cause a greater issue when hacked versus the smaller companies and developers. Just in 2017 the WannaCry ransomware attack that caused millions of dollars in damages to the NHS in the United Kingdom was caused by an exploit in Windows XP, an operating system that had its extended support ended in 2014 (Wall). It is clear the companies and organizations who are at the largest risk of being hacked for severe damage have a problem with updating their code and systems. For large companies in many instances it is not feasible to update their systems because of difficulty and cost of moving all their systems to new platforms or simply just pushing an update to all their networks. In other cases, as mentioned before, it is just to due carelessness among the employees of companies among the sheer number of systems that some companies look over. This does not mean that larger companies should ignore these features though; even if only a few vulnerabilities are fixed that is better than doing nothing.

There are improvements being made for this issue of updates and bugfixes being only actionable by the developers themselves. There is research being done with machine learning that can take broken or vulnerable code and show how that code can be fixed specifically

(Warren). Using a technology like this could greatly improve the likeliness that a developer might realize that there is an actual issue in their code rather than just some phony error message. CVE id's can be long and complicated, so providing an actual change to the code gives the developer a concrete idea of exactly why their code is vulnerable, and exactly how they can fix it. If GitHub continues to improve this technology going into the future, it will become easier and easier for companies to integrate this technology into what they already do, which will help to avoid attacks like Equifax or WannaCry.

ii. Developers

Even for small developers the fact that these security alerts are only actionable by the developers themselves can promote a culture of simply ignoring these issues. Developers must deal with and endless amount of error messages when dealing with the many dependencies that their product relies on, so the fact that these CVE alerts and updates just tell you what the issue is could possible lead to legitimizing the idea that one can just ignore these error messages. What should instead be done is to integrate these alerts into the early stages of a products development cycle, so that any issues that arise are easy to remedy. Again, this may not fix all issues in the future as vulnerabilities will be found in dependencies that finished products use, but at least it will help avoid some future issues. Computer security is based around avoiding as many issues as possible rather than all issues, as there will always be issues in publicly available code.

iii. Lawmakers

Although there are strides being made for developers and organizations to more easily be able to implement security concerns into their development cycle, if it is not in the company's best interest to invest in computer security they never will. Obviously, it is hard to tell exactly when a company should be punished for being hacked as there are cases where it was out of the

company's control whether they were breached. There are cases though, such as the Equifax hack mentioned before, where it was conclusively found that the company and its employees were directly at fault for this preventable hack. Still, even knowing this, it is likely there will be no legal action by the United States toward Equifax, as Equifax continues to operate as it did before, making $3.36 billion dollars in 2017. Even though there is currently $70 billion-dollar class action lawsuit being filed against Equifax (Millis), there needs to be definite legal repercussions for organizations who choose to engage in careless and dangerous security practices. With pre-determined consequences to inattention companies would be dissuaded from acting as Equifax did in this scenario and would also be more likely to use tools like GitHub's Security Alerts in their development flow in the future.

VI. Conclusion

Computer security sometimes is seen to be not worth it as there are just so many products that are many up of so many lines of code that any effort to secure all of it is futile. This mindset it not helpful to both developers and the users of their products; even though every issue cannot be fixed, and every hack cannot be avoided, we should still strive to do everything possible to fix vulnerabilities. GitHub's Security Alerts a step in the right direction, and with GitHub pushing out updates to their product and planning improvements in the future, they are an example of how we should be trying to improve our technologies even though they can't accomplish everything we desire. Going forward computer security should continue to be invested in by both developers and third parties, where ideally preventable computer hacks will begin to become a thing of the past.

Reference Material

Bathompso. “Applying Machine Intelligence to GitHub Security Alerts.” *The GitHub Blog*, 9 Oct. 2018, blog.github.com/2018-10-09-applying-machine-intelligence-to-security-alerts/.

“Committee Releases Report Revealing New Information on Equifax Data Breach.” *United States House Committee on Oversight and Government Reform*, 10 Dec. 2018, oversight.house.gov/report/committee-releases-report-revealing-new-information-on-equifax-data-breach/.

Cress, Steven. “Four Stocks to Leverage the Cyber Security Craze.” *Forbes*, Forbes Magazine, 24 July 2018, www.forbes.com/sites/stevencress/2018/07/24/four-stocks-to-leverage-the-cyber-security-craze/#42c47f1d6c4a.

“CVE - Common Vulnerabilities and Exposures (CVE).” *CVE - Common Vulnerabilities and Exposures (CVE)*, cve.mitre.org/.

“GitHub Help.” *GitHub*, help.github.com/articles/about-security-alerts-for-vulnerable-dependencies/.

Mills, Chris. “Equifax Is Already Facing the Largest Class-Action Lawsuit in US History.” *BGR*, BGR, 8 Sept. 2017, bgr.com/2017/09/08/equifax-hack-lawsuit-class-action-how-to-join/.

nixawk. “CVE-2017-5638 - Apache Struts2 S2-045 · Issue #8064 · rapid7/Metasploit-Framework.” *GitHub*, 7 Mar. 2017, github.com/rapid7/metasploit-framework/issues/8064.

Rashid, Fahmida. “GitHub Security Alerts Make a Difference for Ruby, JavaScript Code.” *Decipher*, 26 Mar. 2018, duo.com/decipher/github-alerts-help-fix-bugs-ruby-javascript-code.

Wall, Matthew, and Mark Ward. “WannaCry: What Can You Do to Protect Your Business?” *BBC News*, BBC, 19 May 2017, www.bbc.com/news/business-39947944.

Warren, Justin. “GitHub Foreshadows Automated Security Fixes.” *Forbes*, Forbes Magazine, 5 Nov. 2018, www.forbes.com/sites/justinwarren/2018/11/02/github-foreshadows-automated-security-fixes/#38755f651092.