Equifax Data Breach

Kenneth J Gollaher

CS-405 Secure Coding, Southern New Hampshire

Module 5-1 Case Study: Triple A and Defense in Depth

Professor Ivan Gappy

Oct 2nd, 2022

**Introduction**

The case study analysis for my chosen security breech is dealing with a large well-known company called Equifax. Even though you may think their security is high and untouchable, unfortunately on September 7th, 2017, the company released a public announcement of a massive data breach of their system. The vulnerability allowed unauthorized access of personal and financial information to roughly 143 million users, which is roughly 44% of the U.S. population. This breach affected residents of Canada, the UK, and the USA. The vulnerability, an Apache Struts RCE [CVE-2017-5638](https://nvd.nist.gov/vuln/detail/CVE-2017-5638#vulnCurrentDescriptionTitle), was identified and actively exploited on March 6th, 2017, but Equifax didn’t identify they were the victim of this vulnerability until July 29th, 2017 (InfoTransec, 2019).

The case made headlines for the number of people affected, the personal and financial information exploited, the root cause, and the preventative measures that was ignored by the security executives that could have eliminated the breach, which will be explained below. Although there have been bigger breaches, the severity of this data breach and the sensitive nature of the data exploited has created a substantial problem for too many people.

**Describe the breach**

According to the National Vulnerability Database (2017), the vulnerability called the Jakarta Multipart parser in Apache Struts [CVE-2017-5638](https://nvd.nist.gov/vuln/detail/CVE-2017-5638#vulnCurrentDescriptionTitle), mismanages during file uploads with incorrect exception handling and error message generation. This allows remote unauthorized access to have the ability to execute arbitrary commands via a CMD prompt. The critical severity of this vulnerability was scored a 10.0, which should be patched as soon as possible due to its impact and risk they pose to the environment.

Since this vulnerability is found within the framework for Apache web-applications, Equifax would have had a difficult time to identify any vulnerability instances. Unfortunately, Equifax didn’t patch this instance when the vulnerability was released back in the beginning of March 2017. The vulnerability allowed for a remote threat to the back-end systems and were highly compromised. This vulnerability led to a series of events that is being considered one of the most significant and largest security breaches of the 21st century (InfoTransec, 2019).

Equifax is one of the three largest credit reporting agencies in the United States and has been a target for many hackers for many years. Unfortunately, it will always be a target. Since Equifax collects personal data from individuals, multiple agencies, and businesses, the data is highly valuable to those on the black market to exploit and sell that data for a large profit.

**Identify the threat(s)**

In Equifax’s case, the immediate threat involved an actor to remotely access the back-end systems to their webservers through arbitrary commands. According to the National Vulnerability Database, this is done through crafted Content-Type, Content-Disposition, or Content-Length HTTP headers. When the vulnerability was released and announced by the appropriate authorities, Equifax ran tests to see if they were vulnerable. The test came back negative for risk, and they were unable to recognize the vulnerability, even after multiple tests. A few days later, on March 13th, 2017, threat actors gained access to a large amount of sensitive information without being detected by the security team (InfoTransec, 2019).

As you can see from this event, Equifax left a potential threat unresolved and caused a remarkable issue that is now an example of what an unresolved potential threat can create for many trusting individuals and business. The reputation of the company decreased significantly and has taken a long time to regain its previous status.

**What could a developer have done to prevent this breach?**

The developer in this case could have likely prevented this breach with the correct vulnerability management tool to detect bugs, or by detecting undefined behavior and dangerous coding constructs, such as the tool [Nodeware](https://nodeware.com/). It is designed to prevent these types of vulnerabilities by scanning the environment and compiling a report to prioritize risks and actions.

All in all, the breach didn’t have to happen, and Equifax’s IT and security team failed to secure the data. Even though both the U.S. Computer Emergency Readiness Team and the Department of Homeland Security urged the security team to patch the software flaw within 48 hours, Equifax failed to make it a priority and patch the vulnerability. The competitors, TransUnion and Experian, both were able to patch the vulnerability within the advised timeline and prevented the breach (Dice Staff, 2020).

**Role of best practices, Triple A and defense in depth in preventing future attacks.**

There are multiple policies and procedures that can be implemented by a developer to prevent many types of breaches. Following those security protocols are crucial to the environment from being accessed by unauthorized users. If there were system and data breach management processes incorporated, the vulnerability could have been discovered much sooner. The first and foremost issue at Equifax was a lack of patching. The Equifax case shows the importance of incorporating the triple-A and defense-in-depth strategy, and how crucial it is to have multiple security layers and detection preventatives in place to prevent this type of breach. In this case, there is a lack of security awareness as well as the lack of leadership to push and apply best practices of security. The triple-A concept, also known as authentication, authorization, and accounting play a vital role in security.

In the data breach in this case, routinely patching systems, disabling default accounts and credentials, and using string passwords with multi-factor authentication are basic ways of preventing breaches, and can go a long way to reduce an organization’s risk. In the article from Dice Staff (2020), they said, “The way we develop code must also be secure throughout the lifecycle; oftentimes security is bolted on at the end, which inevitably will leave gaps. No code is perfect, but we should strive for it.”

References

Dice Staff. (2020, February 18). *What equifax is still teaching us about security - dice insights*. Dice. Retrieved from <https://insights.dice.com/2020/02/18/what-equifax-still-teaching-us-cybersecurity/>

InfoTransec. (2019, February 11). *Analysis of the 2017 Equifax data breach*. InfoTransec. Retrieved from <https://infotransec.com/news/analysis-of-the-2017-equifax-data-breach/>

National Vulnerability Database. (2017, March 10). *CVE-2017-5638 Detail*. The National Institute of Standards and Technology. Retrieved from <https://nvd.nist.gov/vuln/detail/CVE-2017-5638#vulnCurrentDescriptionTitle>