CS-405-10861-M01 Secure Coding 2024 C-3

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

Computer Science Department

Southern New Hampshire University

Eric Farkas

6/9/2024

**Introduction:**

Equifax experienced a major security breach in 2017. This well-known credit reporting company handles sensitive information for roughly 143 million users. The breach occurred on September 7, 2017, exposing personal and financial information through unauthorized access. It affected people in the United States, Canada, and the United Kingdom. A vulnerability in Apache Struts allowed remote code execution to occur. This vulnerability became known and classified as CVE-2017-5638. The initial exploit occurred on March 6, 2017, but was not identified as a breach by Equifax until July 29, 2017 (InfoTransec, 2019). This breach was considered severe due to the sensitive nature of the data and the number of users potentially affected.

**Describe the Breach:**

This breach is caused by incorrect exception handling during file uploads within the Jakarta Multipart Parser, a component of Apache Struts. Once exploited, this incorrect exception handling and error message generation allow unauthorized remote users to execute commands from a crafted HTTP header. The NIST classified CVE-2017-5638 in the National Vulnerability Database with an initial score of 10.0. High-scoring vulnerabilities should be patched immediately due to the potential risks and impacts imposed on the environment.

The vulnerability is within the Apache web application framework, and error message generation did not work properly. These factors made it difficult for Equifax to identify the vulnerability as part of their standard scanning. Ashutosh Barot's discovery and confirmation by NIST should have prompted Equifax to perform the patch. Performing this patch in early 2017 would have prevented the breach. Equifax’s delay in patching put the personal information of 143 million people throughout three countries at risk. This is why the Equifax data breach is considered one of the most significant and largest security breaches of the 21st century (InfoTransec, 2019). Being one of three major credit reporting agencies, Equifax is constantly a target for hackers to obtain sensitive data for individuals, companies, and agencies. This data is valuable to hackers wanting to access other systems requiring the data or sell it for profit to other interested parties.

**Identify the Threat(s):**

The Equifax data breach was partially influenced by the slow reaction to fix vulnerabilities and partially by an opportunistic hacker or hacking team that identified their systems as vulnerable. The hack was carried out by remotely accessing the back-end systems of their web servers through arbitrary commands. According to the National Vulnerability Database, this is done through crafted Content-Type, Content-Disposition, or Content-Length HTTP headers. A further contributing factor was a vulnerability test, which resulted in a false negative result. These factors resulted in a breach on March 6, 2017, where an unauthorized user accessed a large amount of sensitive information without detection (InfoTransec, 2019).

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

Many Apache products are developed in the community as open-source collaborative projects that evolve over time. This may have contributed to the vulnerability that was exploited. The developer and test team should have been more thorough with input validation and error reporting. The developer likely did not anticipate misuse of the framework, and the testers did not define a procedure to include this type of misuse. Equifax was running a system that utilized this framework and likely relied on it to handle input validation instead of creating a custom mechanism. This reliance and likely lack of testing further enabled the vulnerability. Testing and analysis tools exist to prevent vulnerabilities by scanning environments for strange behavior and dangerous constructs. Using this type of tool would have accelerated the detection of the breach.

**Summarize the case by explaining the role of best practices, Triple A and defense in depth in preventing future attacks.**

Based on the information available, no aspects of Triple-A were overcome to accomplish the breach. The hacker would have been denied access if Equifax had required credentials to perform functions on this web service. The practices of authentication and authorization would have been beneficial in this case. Properly logged-in users are also accounted for by tracking activities in the system. Equifax had no knowledge of the hacking activities, which was later explained by a failure in error reporting. There is little information available regarding defense strategies for Equifax. This lack of documentation is likely a defense mechanism itself. It does not appear that the hacker had to circumvent multiple layers of security to perform this hack. It is possible that defense in depth would have prevented or detected the hacker in this exploit by having multiple layers of defense and detection. In this data breach, routinely patching systems, disabling default accounts and credentials, and using string passwords with multi-factor authentication were basic methods of prevention (Dice Staff, 2020).

**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>