Broken Authentication Vulnerability Mitigation Lab

INstructor: Douglas Lundin

Password Change Interaction

Prepared By: Brooklynn Silva & Jonathan Rumley

2020

# Abstract

Insecure software attacks are executed in a variety of methods, but according to the OWASP Top 10, the second most commonly known attack is Broken Authentication. This report will be focused on the importance of having a strong password, which supports the importance of having secure software. Applying strong algorithms and/or methods aid in avoiding password weakness vulnerabilities. Using HaveIBeenPwned to scan passwords for security will ensure that the user’s password will not likely be exploited. This report outlines a mitigation for broken authentication from the detection, to the mitigation, and the timeline of the password vulnerability.

Table of Contents

[Abstract 1](#_Toc57710951)

[Broken Authentication Vulnerability Detection 3](#_Toc57710952)

[Pre-Mitigation Code 3](#_Toc57710953)

[Impacts to Company 4](#_Toc57710954)

[Confirmation of Vulnerability Mitigation 4](#_Toc57710955)

[Post-Mitigation Code 5](#_Toc57710956)

[Post-Mitigation Results 6](#_Toc57710957)

[Timeline of Events 7](#_Toc57710958)

[References 8](#_Toc57710959)

# Broken Authentication Vulnerability Detection

The cyber-team detected a broken authentication vulnerability in the source code of HaveIBeenPwnedApiClient.java. With a lack of a main method that scans the users’ potential passwords, it was determined that it needed to be added in order to ensure secure password authentication.

# Pre-Mitigation Code

Below displays a segment of the pre mitigation-code, HaveIBeenPwnedApiClient.java

Text

Description automatically generated

# Impacts to Company

Possible impacts to the company from broken authentication attacks range from compromised passwords, keys, or session tokens by exploiting flaws in the software to acquire other user information (2020). Depending on the attacker’s intentions, admin user credentials could be more valuable than non-admin users’ credentials, so the attacker could potentially take over just one admin account to get ahold of the system compared to multiple non-admin user accounts. Data breaches cost companies millions of dollars from possibly just one exploit, so any data spill is considered very serious for how it could impact the company, although different types of data breaches could lead to laundering, identity theft, social secuirty fraud, or highly sensitive information (2020).

# Confirmation of Vulnerability Mitigation

Confirmation of the broken authentication vulnerability mitigation starts with successfully mitigating the vulnerability in the java file, HaveIBeenPwnedApiClient.java. This vulnerability was mitigated by developing a main method that would give the client “df47212e0f47445db4a8c9e9e810038d” the opportunity to create a new password, as long as it has not been “pwned”. Our team developed a system that will give the user up to 5 attempts to choose a password that is secure, according to the HaveIBeenPwned password scanner. If the new password is secure, it will be saved and updated to the user account.

# Post-Mitigation Code

The DEV cyber-team modified the following code in the java file, HaveIBeenPwnedApiClient.java, then sent to PROD cyber-team to test the mitigated code.

Text

Description automatically generated

# Post-Mitigation Results

Below is a demonstration of how our mitigation works. Image 1 demonstrates what will occur if the user attempts to create 5 vulnerable passwords. After each scan determined the password was not secure, the user was logged off of the system and no longer allowed any further attempts at creating a new password. Image 2 demonstrates what will occur if the user uses a secure password in one of the attempts. The password was deemed secure as it has not been “pwned”, and the password was changed.

Text

Description automatically generatedImage 1: Post-mitigation code output (attempting to change password 5 different times with non-secure passwords – after 5 attempts of non-secure passwords the system automatically logs on)

Text

Description automatically generatedImage 2: Post-mitigation code output (attempting to change password once with non-secure password then with a secure password – successful secure password allows for password change)

# Timeline of Events

* Vulnerability scan was performed by PROD cyber-team
* Broken authentication vulnerability identified in HaveIBeenPwnedApiClient.java
* Broken authentication vulnerability mitigation process was completed by DEV cyber-team to add depth to the java file by adding a feature that gives the user the opportunity to create a new, SECURE password
* Mitigation works as expected, user is granted 5 attempts to create a new, SECURE password
* Broken authentication vulnerability report written and presented to management in order to bring attention to the vulnerability and prevent future data spills and/or data manipulation

# References

OWASP Top 10. (2020). *OWASP.* Retrieved November 30, 2020, from

https://www.owasp.org/www-project-top-ten/