Edwin Jones

CS 405: Secure Coding

June 8, 2025

Case Study

In October 2023, 23andMe, a major biotechnology and personal genomics company, experienced a significant data breach that compromised the sensitive genetic and personal information of millions of users. The case, known as the 23andMe Data Breach, was reported in several major news outlets, including Wired, whose article can be found at <https://www.wired.com/story/23andme-breach-sec-update>. This incident made headlines because it involved genetic data—information that is deeply personal and cannot be changed—and affected approximately 6.9 million people. The breach occurred after attackers used a credential-stuffing technique, which involves using leaked passwords from unrelated breaches to access user accounts on 23andMe’s platform.

Once inside the breached accounts, attackers were able to exploit 23andMe’s “DNA Relatives” feature to collect additional data from linked family profiles. The exposed data included display names, ancestry information, predicted relationships, and other personal identifiers. This breach qualifies as both a data breach and a security breach, involving unauthorized access to and exfiltration of sensitive personal data. The attackers likely targeted 23andMe because of the high value of its stored genetic data, which can be monetized, misused for identity profiling, or sold on black markets. The immediate threat posed by this breach was unauthorized access to sensitive user data, while the potential long-term threats included genetic identity theft, targeted scams, and the misuse of data in discriminatory or unethical ways.

There are several preventative measures that developers could have implemented to reduce the risk of such a breach. Most critically, 23andMe did not require multi-factor authentication (MFA) at the time of the breach. Enforcing MFA could have prevented the success of credential-stuffing attacks, even when valid credentials were used. Developers could have also implemented rate limiting, IP blacklisting, and anomaly detection to flag and stop suspicious login attempts. Additionally, better access controls for the DNA Relatives feature would have limited the scope of the breach by preventing one compromised account from being used to scrape data from thousands of others.

Policies that would help prevent this type of attack include a mandatory MFA policy for all user accounts, a secure authentication and password policy that flags reused credentials, and an access control policy that enforces the principle of least privilege. A logging and monitoring policy would ensure that any abnormal behavior is detected in real time, allowing developers or security personnel to respond quickly. Regular audits and security assessments of publicly accessible features would also help identify vulnerabilities before they are exploited.

In summary, the 23andMe breach highlights the critical importance of applying secure coding best practices and aligning software development with the Triple A framework—authentication, authorization, and accounting—supported by a defense-in-depth strategy. Authentication failed because user accounts were only protected by passwords, many of which had been reused from other breaches. Authorization controls allowed a single compromised account to view far more data than necessary, violating the principle of least privilege. Accounting was insufficient because the company did not detect the attack until after the data had been widely shared online. A layered defense-in-depth approach, including MFA, rate limiting, access controls, behavior-based monitoring, and rapid response systems, could have drastically reduced the impact of this breach. This case demonstrates that without proper implementation of security principles, even companies with sensitive, high-value data are vulnerable to avoidable attacks.