**Module 5: Case Study**

Zane Milo Deso

Southern New Hampshire University

CS-405: Secure Coding

**Case Study: T-Mobile Data Breach**

T-Mobile September 2023 Data Breach  
<https://firewalltimes.com/t-mobile-data-breaches/>The breach was disclosed on September 21, 2023.

**Introduction**  
 This case gained attention because it marked the ninth publicly known data breach at T-Mobile since 2018, showing a pattern of systemic failure in securing user data. Although smaller in scale compared to earlier incidents, this breach affected 37 million customers, exposing their account PINs, phone numbers, and plan details. The recurrence of breaches, particularly with the same types of exposed data, indicates that T-Mobile’s internal controls, secure coding practices, and layered defenses remained inadequate (Firewall Times, 2024). Given the telecom industry's role in identity verification and SIM management, such leaks create real downstream risks like SIM-swapping, phishing, and identity fraud.

**Breach Explanation**

The breach stemmed from API abuse, in which attackers exploited a poorly secured application programming interface to repeatedly extract sensitive customer data over a period of months. This type of breach involves a combination of secure coding flaws (lack of input validation, insufficient rate limiting) and architectural oversights (exposing internal APIs to external actors). There was no proper authentication required for accessing certain API functions, allowing attackers to programmatically harvest data without triggering alerts or being blocked (Firewall Times, 2024).

**Why was this company a target?**  
 T-Mobile is a valuable target due to its access to sensitive customer data such as mobile numbers, device information, and account credentials; all useful for identity fraud and social engineering. The fact that attackers were able to exploit the same class of vulnerability displays that T-Mobile was a target of opportunity and a repeat target due to weak infrastructure coupled with historical negligence in fixing the root causes of prior breaches (Firewall Times, 2024).

**Immediate Threat**  
 The primary immediate threat was the exposure of account PINs and personal mobile data, which can be used to hijack accounts or perform SIM-swapping attacks. This positioned any affected users at risk of losing access to bank accounts, email, or multi-factor authentication codes linked to their phone numbers.

**Potential Threat Without Resolution**  
 If left unaddressed, the exposed API could allow even further unauthorized scraping of customer data or unauthorized changes to customer accounts. Attackers could use this access to impersonate users, redirect communication flows, or even target enterprise accounts tied to mobile credentials. The fact that the breach sustained for months signifies that T-Mobile lacked proper logging, alerting, and automated anomaly detection; all of which are necessary to mitigate persistent exploitation.

### **How Could It Have Been Prevented** This breach could have been prevented by enforcing API security best practices during development. All public-facing APIs should be protected using authentication tokens, rate limiting, IP filtering, and scope-based authorization. Furthermore, development teams should integrate static application security testing (SAST) and dynamic analysis (DAST) into the CI/CD pipeline to catch issues like missing authentication headers. From a policy perspective, adherence to OWASP API Security Top 10 and NIST 800-53 guidelines would mandate such controls. This attack clearly bypassed multiple layers of Defense in Depth. The specific layers bypassed were the application, identity, and monitoring layers; accentuating the requirement for overlapping safeguards that assume any one layer may fail.

### **Application of Triple A**

Authentication  
 The core vulnerability was that the targeted API lacked proper authentication controls, leaving it vulnerable to repeated data access without requiring user identity verification. A basic bearer token or OAuth2 implementation could have blocked this attack vector.

Authorization  
 Once access was gained, the system lacked fine-grained authorization, meaning the API did not restrict what data could be pulled or by whom. Role-based access controls (RBAC) and resource-scoped tokens would have helped enforce user permissions.

Accounting  
 The breach went undetected for months, showing a lack of proper logging and monitoring. Accounting mechanisms, such as request rate logging, anomaly detection, and alert thresholds for abnormal access patterns, were either misconfigured or absent; allowing any attacker to scrape data without interruption (Firewall Times, 2024).

### **Application of Defense in Depth**

The September 2023 breach depicts a layered security failure. At the application layer, APIs were exposed without requiring authentication. At the identity layer, no access tokens or identity checks were enforced. At the monitoring and response layer, the attack went unnoticed for weeks due to poor request tracking and no behavioral alerting. Had a properly configured Defense in Depth strategy been in place, with API gateways, WAF filtering, IAM policies, and a centralized SIEM, the attacker’s behavior would have raised red flags early. Defense in Depth is designed precisely for this scenario: to ensure that when one control fails (like authentication), others (like anomaly detection or authorization scopes) can contain the damage.

### **Conclusion**

The T-Mobile breach disclosed on September 21, 2023, is not just another isolated incident; it reflects systemic failure in adopting secure coding practices, enforcing Triple A principles, and implementing a true Defense in Depth model. The breach was caused by an unsecured API that lacked authentication and rate limiting, and it went unnoticed for months due to insufficient monitoring. With the right coding standards and layered security architecture, proper token management, logging, anomaly detection, and privileged access controls, this attack would have been either blocked or quickly mitigated. For a company with T-Mobile’s resources and risk profile, continuing to miss these fundamentals is dangerous.

**Reference**

Firewall Times. (2024). *T-Mobile Data Breaches*. Retrieved from [https://firewalltimes.com/t-](https://firewalltimes.com/t-mobile-data-breaches/) mobile-data-breaches/