Published on *The National Law Review* https://natlawreview.com

Don’t be a Dummy – FTC Warns against Inadequate Security Controls in “Dummy” (Non-Production) Environments

Article By:

Allison Trimble

The FTC recently announced a revised settlement with Uber Technologies, Inc. (“Uber”) in which the ride-sharing company has agreed to expand the proposed settlement it reached with the FTC last year over charges that Uber deceived consumers about its privacy and data security practices. The revised settlement took into account a 2016 breach of customer data that Uber failed to disclose during the August 2017 settlement. According to the complaint, Uber software engineers developed and tested software that had connectivity to cloud data using inadequate access controls. In November 2016 Uber learned that hackers exploited this vulnerability in order to gain full access to Uber’s cloud storage environment, which contained unencrypted data files with more than 25 million names and email addresses, 22 million names and mobile phone numbers, and 600,000 names and driver’s license numbers of U.S. Uber drivers and riders.

Although production environments typically receive the most attention from an information security perspective, the Uber settlement highlights the risks of failing to properly secure non-production environments utilized for development, testing, and quality assurance purposes. Neil Chilson, Chief Technologist of the FTC, explains “Insecure non-production environments leave a company open to corporate espionage, sabotage by competitors, and yes, theft of private consumer data.” In short, non-production environments traditionally have weak security controls and hackers know it. With more than 80% of companies reporting they use sensitive production data for non-production activities, there is a critical need to better understand the challenges of securing these environments and potential solutions for doing so:

Securing Non-Production Environments Presents Several Challenges:

1. Demands of the Environment

In order for non-production environments to achieve their intended purpose (provide a space that enables developers to freely write and test code), developers must have broad access to data and functionality, which frequently translates into imposing less formal security standards. In addition, because software is still in the development phase, it may not embody all of the typical security features that would otherwise be incorporated into the production code. Non-production environments instead are typically governed by access controls so only authorized developers can access the environment.

2. Use of “Dummy” Data

Many developers believe dummy data doesn’t serve as the best test data as the variances or error messages cannot be replicated to the same degree of precision as would be the result when “real” client data is used. Actual client data could contain personally identifiable information or other sensitive information that if disclosed in an unauthorized manner, could result in liability for vendors under various laws and regulations.

3. Incorporation of Security Standards

Pressure to meet tight client deadlines may result in a rushed transition from non-production to production environments. As a result, the incorporation of security elements that were not present during the development stage can be missed.

Tips to Protect Non-Production Environments:

1. Data usage within non-production environments should be governed by enterprise level data governance policies and procedures, including limiting data usage to that which is only absolutely necessary (i.e., if personally identifiable information or other sensitive data elements aren’t required, remove them from test data in order to mitigate risk).

2. Emphasize the importance of utilizing data security practices regardless of the stage of software development. When possible, try to replicate production environment security control methods in non-production environments, otherwise seek security methods customizable for purposes of the non-production environment, such as data masking (method by which one-way algorithms are applied to the data enabling de-identification of specific elements within the set).

3. Re-enforce the importance of access controls (multi-factor authentication, VPN, etc.) and ensure they are being enforced.

4. Make sure developers know that unless they are working on an open-source project, proprietary code shouldn’t go within public repositories (and if it is an open source project, code should be carefully reviewed prior to submission for purposes of removing any vulnerabilities, such as hardcoded keys and logins).

5. If security features are removed from development features, have a process for accounting for these items prior to code being placed into production.

6. Conduct protective monitoring of the development environment to determine if the environment is subject to unauthorized activity – (i.e., strange websites being accessed).

7. Include data governance and information security obligations as it relates to non-production environments as part of internal audit processes to ensure proper protocol is being followed.

National Law Review, Volume VIII, Number 136

controls-dummy-non-production