TALES FROM THE CODEFRONT

VERACODE

Preventing Revenue Loss:

How an E-commerce Giant Avoided a Trojanized Dependency Attack

Overview

An e-commerce leader was targeted by a malicious, typosquatted npm package intended to steal customer payment data. Veracode's upstream controls and automated protection blocked the threat, safeguarding both revenue and customer trust.

Problem

- Typosquatted, trojanized UI package introduced via npm to a 500+ microservices e-commerce platform.
- Malicious binary aimed to exfiltrate payment data and session cookies.
- Threat was poised to spread undetected across customer-facing services.

How We Solved It

- Veracode Package Firewall enforced registry-level policies to block the malicious package pre-download.
- Malicious package protection mapped binaries to known malware, denied entry, and guided teams to clean fixes.
- Veracode SCA identified all package references, enabled precise rollbacks, and produced audit-ready SBOMs.

Result

- **Revenue and platform availability protected.**
- **Breach and fraud averted before any customer impact.**

Learn more at www.veracode.com

THE FULL STORY

Background

A large e-commerce giant, running a complex architecture with over 500 microservices, depended on popular open-source libraries like jQuery to build its user interface. The development teams operated under the assumption that the widespread use and popularity of these npm staples were reliable indicators of their safety and integrity.

The Incident

Exploiting this trust, attackers published a typosquatted version of a core UI library. This trojanized package, mimicking a legitimate dependency, embedded a malicious binary designed to exfiltrate session cookies and payment information upon installation. The lookalike package was nearly integrated across the platform, threatening to infect the majority of the company’s customer-facing services without being detected by traditional security scans.

The Impact

A successful compromise would have resulted in a massive data breach, exposing hundreds of thousands of customer records. This could have led to significant fraud-related losses, costly legal battles, and severe revenue disruption from platform downtime. The erosion of customer trust would have caused lasting damage to the brand and its market position.

Resolution with Veracode

Veracode’s upstream controls prevented the attack before it started. At registry-level enforcement, Veracode Package Firewall halted the download of the malicious package using policy-based integrity checks and version reputation signals. Concurrently, Veracode's malicious package protection capabilities analyzed the dependency, mapping its embedded binary to known malware families in its threat intelligence database. It enforced an automatic denial of the package and provided developers with clear guidance to secure, clean versions. Additionally, Veracode SCA identified where the typosquatted package was referenced, accelerated rollback with precise fix versions, and validated clean builds by exporting SBOMs for audit and commerce partner requirements.

Outcome

This preventative action stopped the lateral spread of the trojanized package, which allowed the security team to accelerate a safe rollback and preserve platform availability and revenue.

Key Takeaway

You can preserve ecosystem integrity and protect revenue by using upstream controls to prevent the ingestion of tampered or typosquatted packages and steer development teams toward clean versions.