ENCRYPTION CONSULTING

CASE STUDY

CODE SIGNING SECURITY

The 2023 GitHub code signing certificate theft and how CodeSign Secure prevents it

In January 2023, attackers stole three encrypted code signing certificates from GitHub. No malware was signed, but the incident exposed how fragile software trust can be, and why private keys belong in an HSM.

SUBJECT	INCIDENT	FOCUS
GitHub (Microsoft)	Jan 2023 cert theft	Code signing keys

3
Code signing certificates stolen

~1 day
Undetected attacker access

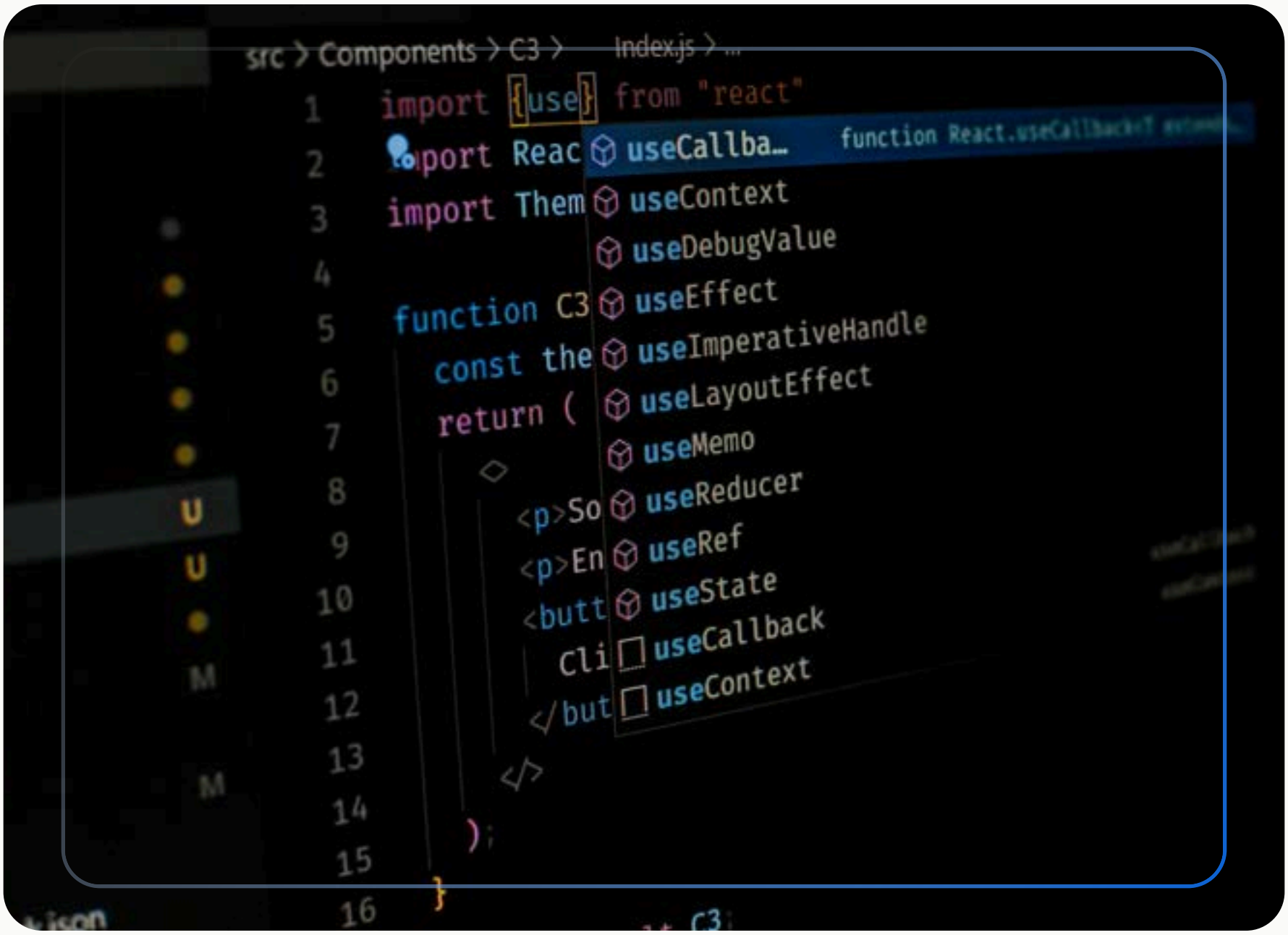
0
Confirmed malicious signatures

GitHub, at the center of the world's software

GitHub, owned by Microsoft, is the world's largest platform for version control and collaborative software development. As of 2023 it hosted more than 100 million repositories and served over 94 million developers, with code hosting, version control, and CI/CD pipelines powering millions of open-source and enterprise projects.

That reach makes GitHub a crucial link in the software trust chain. Its certificates vouch for the software distributed through it, which is exactly what makes it such a high-value target. Compromise those certificates, and an attacker could sign malicious code that millions of systems would trust.

When a platform sits at the center of the software supply chain, its signing keys are the crown jewels. Whoever controls them controls trust.



A breach in the software trust chain

No malware was signed, but the theft exposed real weaknesses in how signing certificates were stored, accessed, and watched.

1

A compromised access token

Attackers used a stolen Personal Access Token, tied to a machine account, to reach GitHub repositories and clone them, including ones holding code signing certificates.

2

Three certificates stolen

They took three encrypted code signing certificates: one Apple Developer ID valid until 2027, and two issued by DigiCert.

3

A day of undetected access

The intrusion went unnoticed for about a day before GitHub spotted it, exposing a gap in real-time monitoring for machine accounts and tokens.

4

Keys in repos, not HSMs

The certificates sat in accessible repositories rather than a hardware security module, and the token's broad permissions pointed to weak role-based access.

5

Slow, complex revocation

Revoking the stolen certificates meant coordinating with DigiCert and Apple, a time-consuming process that disrupted operations, and the 2027 Apple certificate had to be watched for misuse long after.

> “Enterprises don't lose trust when a certificate is stolen, they lose it when they can't prove the stolen key was never usable. HSM-backed signing is what makes that proof possible.
>
> **ENCRYPTION CONSULTING, CODESIGN SECURE TEAM**
> CODESIGN SECURE · ENCRYPTION CONSULTING · CODE SIGNING SECURITY

How CodeSign Secure would have stopped it?

CodeSign Secure is built so that even if certificates are stolen, the private keys behind them stay locked away and unusable, with every signature authorized, logged, and monitored.

HSM-backed key storage

Private keys are generated and used inside FIPS 140-2 Level 3 HSMs and never leave the device, so stolen certificates have no usable key behind them.

Client-side hashing (KSP)

Our proprietary Key Storage Provider hashes code locally, so signing happens inside the HSM and the private key is never exposed to the user or application.

Automated signing in CI/CD

Signing is automated directly inside the CI/CD pipeline, so only authorized code is signed, consistently, with no manual handling of keys.

Tamper-proof audit trails

Every signing operation is logged in detail, the kind of real-time visibility that lets a team catch unauthorized access before it becomes a theft.

Strict role-based access

Granular RBAC ensures only authorized people and processes can start a signing operation, closing the over-broad permissions that opened the GitHub breach.

SIEM monitoring and compliance

Integration with SIEM tools like Splunk gives centralized, real-time alerting, while the platform helps meet CA/B Forum and GDPR requirements.

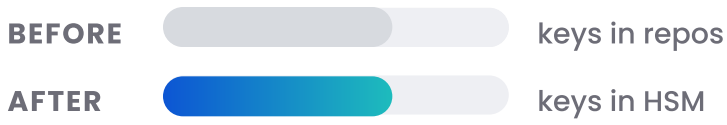
04 THE IMPACT

No malware signed, but the risk ran deep

GitHub confirmed the certificates were never used to sign malware. Even so, the theft showed how much damage a stolen signing key could do.

$4.9M

What a breach can cost, when keys aren't HSM-protected



Malware distribution risk

Decrypted, the certificates could have signed malware that slips past security checks on Windows and macOS systems.

Reputation on the line

A breach like this risks the trust of GitHub's 94 million developers, the kind of confidence that takes years to rebuild.

Operational disruption

Revoking and reissuing certificates invalidated signed releases of GitHub Desktop for Mac and Atom, forcing users to update.

An industry-wide warning

With software supply chain attacks up 742% in recent years, the theft renewed focus on protecting signing keys and meeting CA/B Forum and GDPR rules.

IN SUMMARY

Secure the keys behind your software

The 2023 GitHub theft is a reminder of how fragile software trust can be. Three encrypted certificates were stolen, and only their encryption kept them from being misused. CodeSign Secure removes that gamble: with FIPS 140-2 Level 3 HSMs, client-side hashing, automated signing inside your CI/CD pipeline, full audit trails, strict access control, and real-time monitoring, stolen certificates stay useless and every signature stays accountable. Securing the software supply chain is a shared responsibility, and it starts with protecting the keys.

SEE IT IN ACTION

Make stolen signing keys useless.

Book a CodeSign Secure Demo →