OWASP MEETUP

Software Supply Chain Security

Meet the Speakers

Kostadin Ivanov
CEO & CISO

- BSc with honors in Cybersecurity and Ethical Hacking
- International Cybersecurity Expert
- Specialist in digital forensics, digital evidence, risk assessment, and data analysis
- Investigator of cybercrimes for private enterprises in Tyrol
- Leader in cybersecurity teams (Threat Intelligence, Incident Response & Penetration testing)
- Auditor of Information Security Management Systems
- Auditor of Business Continuity Management Systems
- Auditor of IT Service Management Systems
- Auditor of Risk Management Systems
- Delivered hundred+ successful cybersecurity projects in Europe and abroad, utilizing a wide array of technologies and techniques for cybersecurity diagnostics and assurance
- Lecturer and Mentor in Cybersecurity

Ivan Kadiev
CTO

- BSc in Computer Science with Minor in Economics
- Full stack software engineer with years of enterprise experience building complex software solutions on Angular, HTML, CSS, Typescript, Javascript
- Lead architect in the development of complex microservice infrastructures
- Coordinator of product software engineering teams
- Specialist in building and coaching cross-functional Agile teams; integrates customer feedback loops into sprint cycles
- Lead expert in artificial intelligence development
- 2 professional certificates from Oracle for Java
- Enterprise level expertise in developing and integrating whole systems (database, API communication, front-end, back-end, third-party integrations, payment services, API workflow optimizations).

Our Agenda for Today

01 What is the problem we're fixing

02 Why do we need to take Software Supply Chain Security seriously

03 Defense strategy

A03:2025 - Software Supply Chain Failures

CWE-477
Use of Obsolete Function

CWE-1035
2017 Top 10 A9: Using Components with Known Vulnerabilities

CWE-1104
Use of Unmaintained Third Party Components

CWE-1329
Reliance on Component That is Not Updateable

CWE-1395
Dependency on Vulnerable Third-Party Component

Highest average incidence rate of 5.19% and admittedly highest concern across organizations.

Supply Chain Attacks Timeline

Attacks are shifting towards exploiting trust relationships through third-party dependencies.

2020

SolarWinds Orion

Injected malicious code directly into the build process. Affected 18,000+ customers including government agencies.

2021

MOVEit Transfer

Vulnerability in a "trusted tool" caused cascaded across 620+ organizations (BBC, British Airways, etc.). Discovered in 2023.

2021

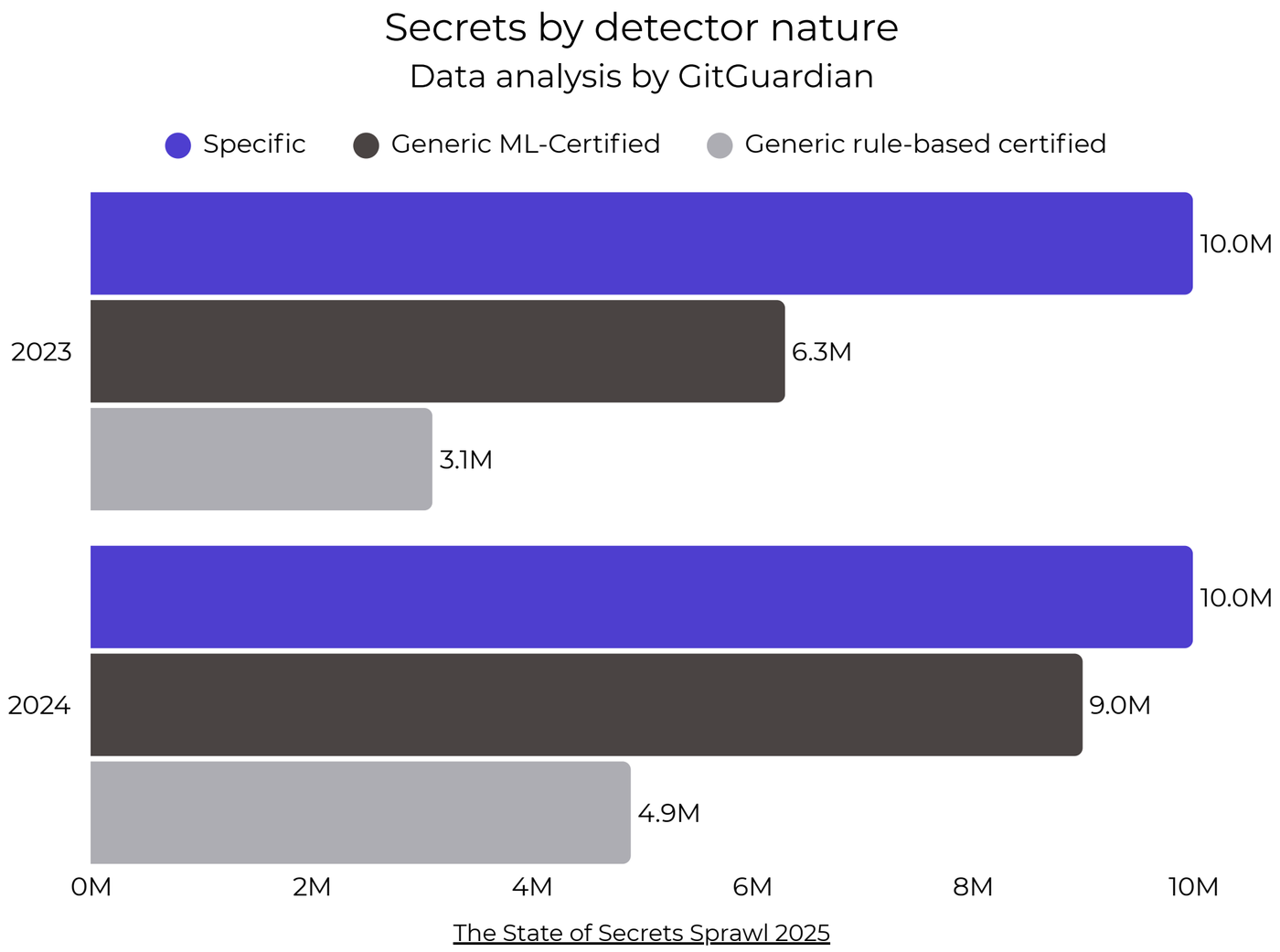
3CX

Compromised Trading Technologies library → 3CX employee downloads it → 3CX build process compromised → poisoned update pushed to all customers.
Discovered in 2023.

2024

XZ Utils Backdoor

Added malicious code hidden in tarball releases. Could have enabled RCE on millions of Linux systems through OpenSSH.

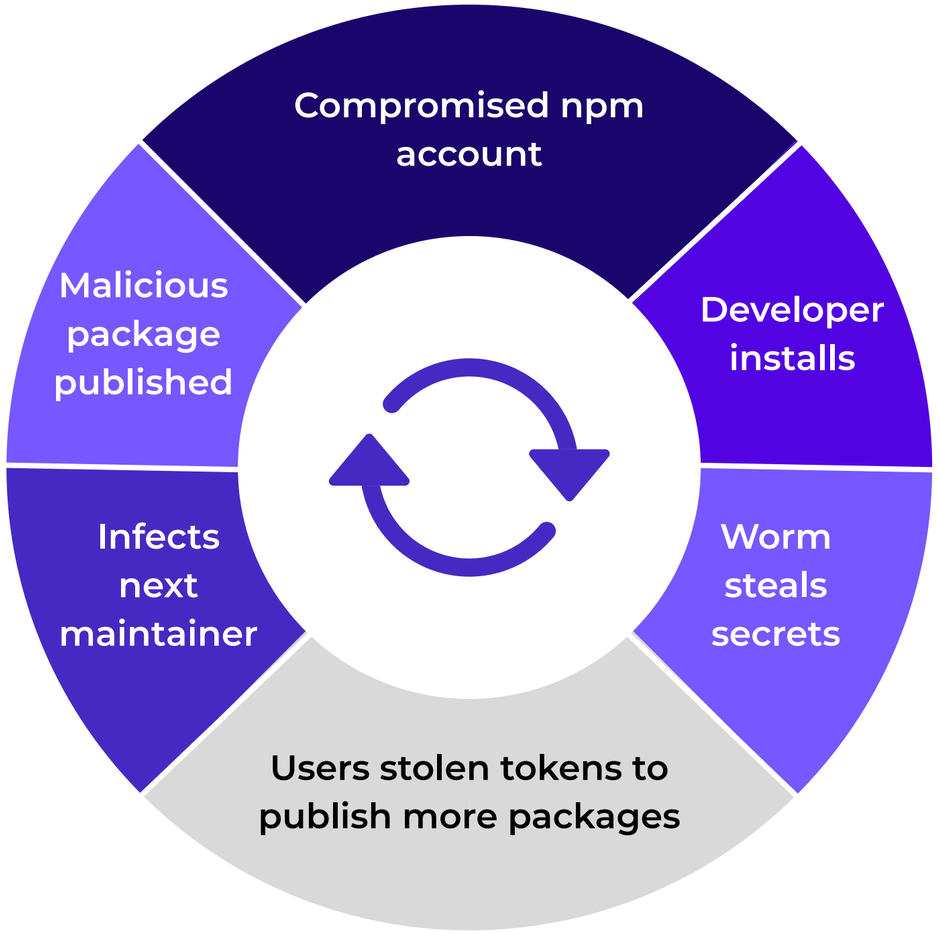


The State of Secrets Sprawl 2025

23,8M total hardcoded secrets uncovered in public GitHub commits by GitGuardian (2025 Report)

Shai-Hulud 2.0 High-level Attack Flow

from initial infection to final propagation.



In-depth overview of the attack:
https://blog.gitguardian.com/shai-hulud-2/

Results from the Shai-Hulud 2.0 Attack (November 2025)

487

organizations breached and forced to trigger incident response plans

50M

in direct crypto theft

132M

monthly downloads targeting developer environments

27K

known malicious repositories created by the malware in their own namespaces

Supply Chain Red Flags

And how to spot them

- Shadow versions
- End-of-Life software
- Blind spots
- Uncontrolled changes
- Weak supply chain
- Excessive permissions
- Risky downloads
- Delayed patching
- Untested updates
- Misconfigurations
- Insecure CI/CD

Blind spots

Log4Shell

Log4j vulnerability

Hidden flaw

A zero-day RCE vulnerability in the ubiquitous log4j Java library allowed unauthenticated remote code execution via simple text strings.

Massive Impact

The library was often buried deep in transitive dependencies, meaning companies did not know they were using it.

Trivial Exploit

Attackers compromise a server just by typing a malicious string (e.g., `${jndi:ldap://...}`) into a login box, chat window, or user agent field.

Visibility is key

You cannot patch what you cannot see. This event made the Software Bill of Materials (SBOM) a mandatory industry standard.

Supply Chain Defense Strategy

- Centralized SBOM
- Verified & Signed Sources
- Deep Dependency Trace
- Version Pinning
- Minimize Attack Surface
- Lifecycle Management
- Continuous Inventory
- Secure Toolchain
- Automated Scanning
- Harden CI/CD Pipeline

Defensive best practices currently being adopted by high-maturity enterprise software companies.

Centralized SBOM

Automated SBOM

The "living" inventory

Zero-Latency Search

Because SBOMs are generated at every build, security teams can find a vulnerable library across the entire org in seconds, not days.

Transitive Visibility

The system maps not just your dependencies, but the dependencies of your dependencies, uncovering hidden risks deep in the chain.

Machine Readable

SBOMs are exported in standard formats, allowing automated tools to ingest and analyze them without human spreadsheet work.

Compliance as Code

The process satisfies strict regulatory requirements automatically, freeing developers from manual documentation.

THANK YOU