# How we're helping to reshape the software supply chain ecosystem securely

cloud.google.com/blog/products/identity-security/how-were-helping-reshape-software-supply-chain-ecosystem-securely


As we start the new year, we see ongoing revelations about an attack involving SolarWinds and others, that in turn led to the compromise of numerous other organizations. Software supply chain attacks like this pose a serious threat to governments, companies, non-profits, and individuals alike. At Google, we work around the clock to protect our users and customers. Based on what is known about the attack today, we are confident that no Google systems were affected by the SolarWinds event. We make very limited use of the affected software and services, and our approach to mitigating supply chain security risks meant that any incidental use was limited and contained. These controls were bolstered by sophisticated monitoring of our networks and systems.

Beyond this specific attack, we remain focused on defending against all forms of supply chain risk and feel a deep responsibility to collaborate on solutions that benefit our customers and the common good of the industry. That's why today we want to share some of the security best practices we employ and investments we make in secure software development and supply chain risk management. These key elements of our security and risk programs include our efforts to develop and deploy software safely at Google, design and build a trusted cloud environment to deliver defense-in-depth at scale, advocate for modern security architectures, and advance industry-wide security initiatives.

To protect the software products and solutions we provide our cloud customers, we have to mitigate potential security risks, no matter how small, for our own employees and systems. To do this, we have modernized the technology stack to provide a more defensible environment that we can protect at scale. For example, modern security architectures like BeyondCorp allow our employees to work securely from anywhere, security keys have effectively eliminated password phishing attacks against our employees, and Chrome OS was built by design to be more resilient against malware. By building a strong foundation for our employees to work from, we are well-prepared to address key issues, such as software supply chain security. Many of these topics are covered more extensively in our book *Building Secure and Reliable Systems*.

## How we develop and deploy software and hardware safely at Google

Developing software safely starts with providing secure infrastructure and requires the right tools and processes to help our developers avoid predictable security mistakes. For example, we make use of secure development and continuous testing frameworks to detect and avoid common programming mistakes. Our embedded security-by-default approach also considers a wide variety of attack vectors on the development process itself, including supply chain risks.

A few examples of how we tackle the challenge of developing software safely:

- **Trusted Cloud Computing**: Google Cloud's infrastructure is designed to deliver defense-in-depth at scale, which means that we don't rely on any one thing to keep us secure, but instead build layers of checks and controls that includes proprietary Google-designed hardware, Google-controlled firmware, Google-curated OS images, a Google-hardened hypervisor, as well as data center physical security and services. We provide assurances in these security layers through roots of trust, such as Titan Chips for Google host machines and Shielded Virtual Machines. Controlling the hardware and security stack allows us to maintain the underpinnings of our security posture in a way that many other providers cannot. We believe that this level of control results in reduced exposure to supply chain risk for us and our customers. More on our measures to mitigate hardware supply chain risk can be found in this blog post.

- **Binary Authorization**: As we describe in our Binary Authorization whitepaper, we verify, for example, that software is built and signed in an approved isolated build environment from properly checked-in code that has been reviewed and tested. These controls are enforced during deployment by policy, depending on the sensitivity of the code. Binaries are only permitted to run if they pass such control checks, and we continuously verify policy compliance for the lifetime of the job. This is a critical control used to limit the ability of a potentially malicious insider, or other threat actor using their account, to insert malicious software into our production environment. Google Cloud customers can use the Binary Authorization service to define and automatically enforce production deployment policy based on the provenance and integrity of their code.

**Change Verification**: Code and configuration changes submitted by our developers are provably reviewed by at least one person other than the author. Sensitive administrative actions typically require additional human approvals. We do this to prevent unexpected changes, whether they're mistakes or malicious insertions.

## Reshaping the ecosystem

We also believe the broader ecosystem will need to reshape its approach to layered defense to address supply chain attacks long-term. For example, software development teams should adopt tamper-evident practices paired with transparency techniques that allow for third-party validation and discoverability. We have published an architectural guide to adding tamper checking to a package manager, and this is implemented for Golang. Developers can make use of our open-source verifiable Trillian log, which powers the world's largest, most used and respected production crypto ledger-based ecosystem, certificate transparency.

Another area for consideration is limiting the effects of attacks by using modern computing architectures that isolate potentially compromised software components. Examples of such architectures are Android OS's application sandbox, gVisor (an application sandbox for containers), and Google's BeyondProd where microservice containerization can limit the effects of malicious software. Should any of the upstream supply-chain components in these environments become compromised, such isolation mechanisms can act as a final layer of defense to deny attackers their goals.

## Our industry commitment and responsibility

The software supply chain represents the links across organizations—an individual company can only do so much on their own. We need to work together as an industry to change the way software components are built, distributed and tracked throughout their lifecycle.

One example of collaboration is the Open Source Security Foundation, which Google co-founded last year to help the industry tackle issues like software supply chain security in open source dependencies and promote security awareness and best practices. We also work with industry partners to improve supply chain policies and reduce supply chain risk, and publish information for users and customers on how they can use our technology to manage supply chain risk.

## Pushing the software ecosystem forward

Although the history of software supply chain attacks is well-documented, each new attack reveals new challenges. The seriousness of the SolarWinds event is deeply concerning but it also highlights the opportunities for government, industry, and other stakeholders to collaborate on best practices and build effective technology that can fundamentally improve the software ecosystem. We will continue to work with a range of stakeholders to address these issues and help lay the foundation for a more secure future.