An Everything-as-Code Approach to Securing the Software Supply Chain

How a Secure Development Ecosystem Supports Zero Trust Principles

leidos

The directive to enhance software supply chain security has been a north star for government technologists over the past couple of years, but a June 2023 Office of Management and Budget (OMB) memo suggests federal agencies still have a long road ahead. The document clarifies action items from National Institute of Standards and Technology (NIST) guidance and extends timelines laid out in an earlier memo, "Enhancing the Security of the Software Supply Chain Through Secure Software Development Practices."

The June OMB memo also emphasizes that "when software producers responsibly implement industry-leading development practices, which include minimizing risk from third-party code, the burden of accounting for secure software development practices is appropriately placed on the producer of the end product rather than Federal agencies."

In other words, while agencies must collect attestations for their critical software, the onus is on software producers to actually create those attestations, at a bare minimum. To truly secure the most critical asset at the heart of software, the data, software producers must go beyond basic compliance to take responsibility for the security of every element — open-source and proprietary code, development tools and processes, execution layers and runtime environments, all the way through deployment and monitoring. Complete security down to the cellular level of software: mission-critical data.

"Everything is essentially becoming software, even in what used to be just the hardware space, or just the IT space," says Paul Burnette, vice president and director of software at Leidos. "Elements like infrastructure, networking, compute resources, they're all becoming more and more dictated and controlled by software or as software."

This metamorphosis is a driving force behind Leidos' "Everything as Code," or EaC, approach to secure software development. In a technology landscape that's constantly shifting, blocks of code create a sense of order. A block of code is discrete and repeatable, it executes the same way every time. An EaC philosophy allows developers to bring some of that certainty to the entire development lifecycle.

"When you describe everything as code, and you manage it all as code — infrastructure, security, policy, configurations — you are essentially creating a known state of execution that you can maintain," Burnette says. "You know when something changes, and you also know when something goes wrong, and how to find where it went wrong."

"Everything is essentially becoming software, even in what used to be just the hardware space, or just the IT space. Elements like infrastructure, networking, compute resources, they're all becoming more and more dictated and controlled by software or as software."

—

Paul Burnette

Vice President and Director of Software, Leidos

SecDevOps Puts Security First

As the software development and delivery pipeline becomes more automated, the speed to deployment increases — as do attack surfaces. To bolster security throughout development, Leidos created portable SecDevOps (pSDO), its solution for rapid automated production of trusted software. As the name indicates, security comes first, a necessity in a zero trust environment.

"Zero trust is a change in mindset. You used to secure at the perimeter, but the mindset now is the enemy is already inside the gates," says Jeremy Burton, chief software architect at Leidos. "How do you limit what they're able to do once they're inside the perimeter? And how do you limit the blast radius for the actions that they take?"

The speed of modern development comes with concessions, and it's impossible to build completely impervious, ironclad software. Leidos' EaC and pSDO philosophies leverage zero trust principles by pulling security from the perimeter to integrate it throughout the development process and the deployed operational software. Risk will never be eliminated, but the goal is to identify and mitigate threats before they cause damage throughout the development process and the deployed operational software.

pSDO automates deployment, configuration and management for on-premise, disconnected and cloud development environments, while its built-in security controls decrease risk. It employs a continuous integration/continuous delivery (CI/CD) toolchain that automates testing and increases the speed and agility of development.

"pSDO is essentially our approach to managing, through everything as code, the entire development lifecycle," Burnette says, "including the software bill of materials, how we manage the software bill of materials, and what gets handed over into the operational runtime environment."

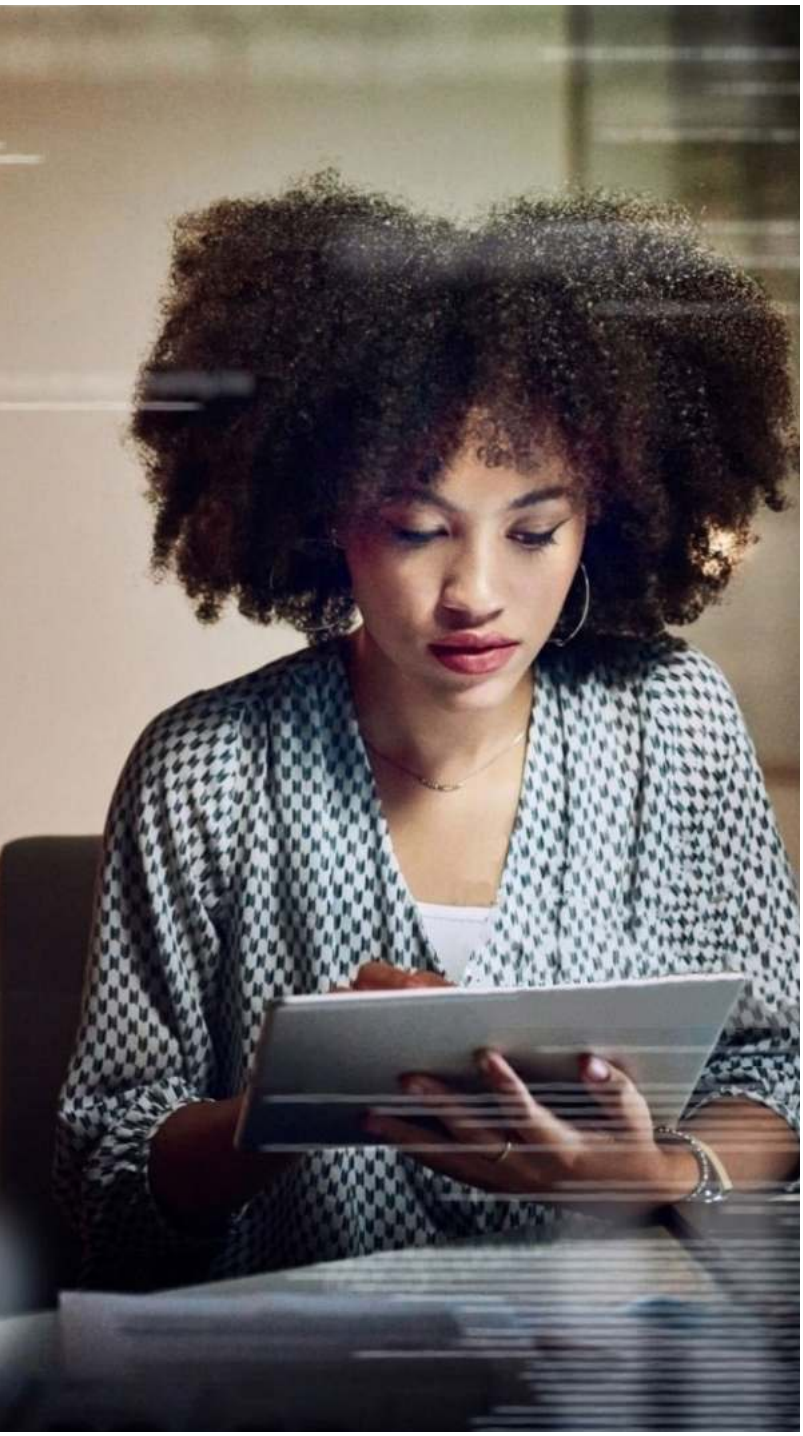
Enhancing the Software Bill of Materials

The software bill of materials, or SBOM, is an essential building block in a zero-trust environment. Rarely is software built from scratch. Implementing open-source components leads to more efficient development but, like many timesavers, it also increases risk. The key to safely using someone else's code is understanding its provenance.

"You can't have a zero trust approach to the way that you design your networks, the way you design your systems, unless you have some degree of assurance that the software that you're bringing in isn't itself riddled with security holes."

—

Jeremy Burton
Chief Software Architect, Leidos



"You can't have a zero trust approach to the way that you design your networks, the way you design your systems, unless you have some degree of assurance that the software that you're bringing in isn't itself riddled with security holes," says Burton. "[The SBOM] is a foundational piece."

Consider software as a recipe. Many software producers view the SBOM as an ingredients list. Leidos takes a deeper approach, turning it into a complete nutrition label that highlights the "health," or overall risk posture of the software, not just the ingredients. Software is more than the sum of its parts, and its security depends on how much risk each of those parts introduces.

This approach offers multiple benefits:

- **Detailed provenance.** What components are included in the software, and where did each component come from? Are there any known or potential risks associated with them?

  "The developer is saying, 'I've given you this SBOM. This is the software I'm using, and I am also making an attestation that this software is of known provenance. As far as I'm aware, it doesn't have any significant risks,'" Burton says. "It's not merely a case of generating an SBOM, it's a case of, 'I generated an SBOM and I'm making a statement about the goodness of the third-party software that I'm using.'"

- **Vulnerability identification.** The Log4j zero-day vulnerability provides a real-life use case for SBOMs. When the vulnerability was discovered in the heavily used logging tool, organizations scrambled to assess the impact.

  "The faster that you can identify that system A has this vulnerability because its SBOM tells you that this part, this ingredient, is running in system A and not in system B, you know which systems you need to patch faster," says Drew Formica, lead software engineer at Leidos. "You're not searching for a needle in the haystack."

- **Insider threat detection.** The SBOM also serves an internal checks and balances function when insider threats can be difficult to detect.

  "An SBOM … allows me to answer questions like, 'Is what I'm running now what I was actually supposed to be delivered?'" Burton says. "It provides a vehicle for adding additional checks in that chain that we can use to detect changes." Any deviation from the SBOM would indicate a potential red flag to explore further.

At Leidos, however, basic provenance isn't enough. Leidos goes beyond surface-level compliance to take responsibility for the full software supply chain.

This includes not only the ingredients of the software it creates but also the tools it uses throughout the process. Leidos' pSDO automates both SBOM generation and security analysis of third-party applications.

"We have build tools, we have test tools — those pieces of software are also having an effect on the software we're building and deploying, and we have the ability to put security guards around those tools to make sure we're implementing them in a secure way," Burnette says. "You are essentially providing the tools access to the code that you're deploying … We'd be remiss if we didn't include that complete supply chain, the chain of custody of not only all the software that we're pulling from open source, but also all of the software that we're using to build, deploy, manage and even run our software."

Automating a Secure, Consistent Runtime Environment

When everything is code, software runs everywhere — from the cloud, to on-premise, to edge devices. While common runtime environments have emerged to offer greater flexibility in deploying software anywhere, these new solutions have also created unintended security gaps.

In response, Leidos created the Leidos Secure Runtime Environment, or LSRE, to offer security and consistency. LSRE creates repeatable, hardened Kubernetes clusters to enable rapid development, testing, deployment and maintenance in any environment — public cloud, private cloud, on-premise, air-gapped and edge.

"What we do is inherently tech agnostic — we can adapt to the customer environment," Burnette says. "You can still use the tools that you want, we just have ways of bringing a level of security to them that helps you implement zero trust without being boxed in."

While many organizations are innovating around Kubernetes, for example, Leidos goes above and beyond with LSRE, securing all pieces around Kubernetes clusters, in addition to how it stands up and utilizes the clusters. Ultimately, LSRE is creating an architecture to leverage and secure its Kubernetes clusters without being bound to a single implementation or vendor.

"Given our portfolio of different operating environments and restrictions in which we have to build, deploy and maintain our software, this is where we really shine," Formica says. "Everywhere from fully connected systems to air gapped to embedded systems, we're doing secure software across a large variety — and it's not one size fits all."

LSRE can stand up production environments in 15 minutes, and components can be configured to comply with industry standards, regulatory requirements and security frameworks. Along with continuous scanning, these features enhance the security posture of the solution and reduce the time to accreditation and authority to operate.

"What we do is inherently tech agnostic — we can adapt to the customer environment. You can still use the tools that you want, we just have ways of bringing a level of security to them that helps you implement zero trust without being boxed in."

—

Paul Burnette

Vice President and Director of Software, Leidos

"We can quickly provide a runtime environment that has met the security technical implementation guides that are being produced and revisioned quite frequently, so that the images that we produce can run with security enforced from the beginning," Formica says.

Protecting the Most Important Asset

Ultimately, the objective of every tool and solution in the software supply chain, of the entire ecosystem supporting a zero trust architecture, is to protect data. Mission-critical data is the most important asset and the most difficult to secure, but that's the goal of Leidos' secure software development.

"How you access data, how you move data around, how you secure that data, how you build a software architecture that ensures the lineage, the pedigree and the provenance of that data is critical, because that data is the most valuable thing inside the software," Burnette says. "At its core, that's what software is trying to do, transform data."

Leidos is tackling its data responsibility head-on, and success depends on striking a delicate balance of protecting data without limiting its potential. The key is developing the tools for understanding and applying security at every level, from the perimeter to the individual unit of data, Burnette says.

"You could essentially provide security from all the way within the software and build an inherently zero trust architecture, only allowing the right people to have access at the right time, and knowing if there's any out-of-the-ordinary behavior within your software," Burnette says. "That's something Leidos specializes in — understanding the risk in the system and what's going on across all of the access controls, all the way down to the data attribute level."

Learn more about how Leidos can help your agency secure development processes and advance toward zero trust.

leidos

#23-Leidos-0816-26780