

JUN 2025

SOFTWARE SUPPLY CHAIN SECURITY

Enhancing Trust and Resilience Across
the Software Development Lifecycle

BROUGHT TO YOU IN PARTNERSHIP WITH

digicert®

ENDOR LABS

Table of Contents

INTRODUCTION AND DZONE RESEARCH

03 Welcome Letter

Lucy Marcum | *Acquisitions Editor at DZone*

04 Key Research Findings

AN ANALYSIS OF RESULTS FROM DZONE'S 2025 SOFTWARE SUPPLY CHAIN SECURITY SURVEY

G. Ryan Spain | *Freelance Software Engineer, Former Engineer & Editor at DZone*

FROM THE COMMUNITY

25 [infographic] Software Security Treat or Threat?

LEVERAGING SBOMS TO CONTROL YOUR SUPPLY CHAIN CHAOS

Derrick Kean Auxtero | *Freelance Artist*

26 Software Supply Chain Security Regulations From a DevSecOps Perspective

Apostolos Giannakidis | *Principal Product Security Engineer at Microsoft*

33 Maximizing Return on Investment When Securing Our Supply Chains

WHERE TO FOCUS OUR LIMITED TIME TO MAXIMIZE REWARD

Justin Albano | *Software Engineer at IBM*

38 [checklist] Practical Steps to Secure the Software Supply Chain End to End

Karteeek Kotamsetty | *Lead Customer Engineer at Google Public Sector LLC*

41 AI-Powered Security for the Modern Software Supply Chain

REINFORCING SOFTWARE INTEGRITY IN AN ERA OF AUTONOMOUS CODE AND EXPANDING RISK

Akanksha Pathak | *Senior Cybersecurity Consultant at Visa Inc.*

ADDITIONAL RESOURCES

44 Solutions Directory

Meet the Team

DZone's Content and Community team can be found reviewing contributor pieces, working with authors and sponsors, and coordinating with our researcher and designer to deliver valuable, high-quality content to global DZone-ians.

Caitlin Candelmo
Carisse Dumaue
Melissa Habit
Lucy Marcum
Dominique Roller
Ging Villena

Director, Site Strategy
Content Editor
Senior Manager, Production Ops
Acquisitions Editor
Community Specialist
Content Editor

Welcome Letter

By Lucy Marcum, Acquisitions Editor at DZone

Security permeates all aspects of life. It's a feeling people crave from their relationships, partnerships, governments, and entities they trust. Everyone wants to know that they are in good hands and that the information they trust to share will be protected at all costs.

Arguably the most important connotation of the word, security is a *practice*. While people want security from the world, they must also provide security by protecting secrets, safeguarding personal information, and entrusting only those within the right positions — tactics that all lean conveniently into software supply chain security.

We live in a day and age where it can be difficult to trust what you see on the internet and where hackers are becoming brilliant at what they do. Organizations are frequently the target of attacks that result in massive data leaks, crippling the trust that their users have in their software. Different governments have varying and quickly changing mandates for secure software practices, making compliance difficult to obtain while also being of the utmost importance for successful deployments.

The rapid advancements have left many developers wondering, "How do I ensure my software supply chain is secured?"

With the *2025 Software Supply Chain Security Trend Report*, DZone is analyzing the top trends for successful supply chain security to answer this very question.

DZone's in-house research dives into adoption trends of zero-trust architecture, the threat of shadow AI and other risks posed by AI use, and managing cloud and open-source security postures. Complementing the research, the expert-written articles provide detailed reviews of modern threat detection, government regulations and compliance, the increasing importance of SBOMs to unify company culture and practices, and much more.

Through research, checklists, and analytical articles, the goal of this Trend Report is to provide you, the people whose boots are on the ground, with actionable insights to mitigate any supply chain security challenges that come your way.

You should feel secure in your work, just as your work should be secure from any bad actors. 🛡️

Safe developing,



Lucy Marcum



Lucy Marcum

🛡️ @lucymarcum

Lucy manages the acquisition process and strategy for DZone Publications, from sourcing new contributors and community members to leading them through the editorial review. She also edits publications, creates different components of Trend Reports, and collaborates with the Site Content and Community team. Outside of work, Lucy spends her time reading, writing, running, and trying to keep her cats, Olive and Tiger Lily, out of trouble.



Key Research Findings

An Analysis of Results From DZone's 2025 Software Supply Chain Security Survey

G. Ryan Spain, Freelance Software Engineer, former Engineer & Editor at DZone

Gone are the days of fragmented security checkpoints and analyzing small pieces of the larger software security puzzle. Today, we are managing our systems for security end to end. For those who are arriving late, a software supply chain refers to every existing touchpoint across the SDLC, specifically where security is concerned. Thanks to this shift, software teams have access to a more holistic view of our entire software security environment.

Software supply chains are on the rise as security continues to flourish and evolve across modern software systems. Through the increase of zero-trust architecture and AI-driven threat protection strategies, our security systems are more intelligent and resilient than ever before.

For these Key Research Findings, we asked our global audience of software professionals to join us in exploring the software supply chain — every touchpoint and security decision — with topics including software bills of materials (SBOMs), AI/ML threat defense, SDLC application security, and much more.

In April and May, we surveyed software developers, architects, and other IT professionals in order to gain insight on the current state of supply chain security in the software development space.

Methods

We created the survey with question formats that included mainly single and multiple choice, plus options for write-in responses in some instances. The survey was disseminated via email to DZone and TechnologyAdvice opt-in subscriber lists as well as promoted on dzone.com, in the DZone Core Slack workspace, and across various DZone social media channels. The data for this report were gathered from responses submitted between April 28, 2025 and May 26, 2025; we collected 113 complete and partial responses.

Our margin of error for the results of this survey is 10%, at a confidence level of 95%, and this report treats comparative results of 10% or less as insignificant. Segmented responses may have lower confidence levels due to smaller sample sizes.

Demographics

Before diving in, we noted certain key audience details in order to establish a more solid impression of the sample from which results are derived:

- 19% of respondents described their primary role in their organization as "Developer team lead," 18% described "Developer/Engineer," 13% described "Technical architect," 11% described "DevOps Lead," and 11% described "Consultant/solutions architect." No other role that we provided was selected by more than 10% of respondents.*
- 60% of respondents said they are currently developing "Enterprise business applications," 57% said "Web apps/ Services (SaaS)," 42% said "Native mobile apps," and 42% said "Boxed software with updates over the web."
- "JavaScript (client-side)" (58%) was the most popular language ecosystem used at respondents' companies, followed by "Python" (48%), "Java" (45%), "Node.js (server-side JavaScript)" (36%), and "C/C++" (33%).
- Regarding responses on the primary language respondents use at work, the most popular was "Java" (25%), followed by "Python" (22%), and "JavaScript (client-side)" (14%). No other language was selected by more than 10% of respondents.
- On average, respondents said they have 20.89 years of experience as an IT professional, with a median of 20 years.
- 55% of respondents work at organizations with < 100 employees or reported being self-employed, 14% of respondents work at organizations with 100-999 employees, and 31% of respondents work at organizations with 1,000+ employees.*

Note: For brevity, throughout the rest of these findings we will use the term "developer" or "dev" to refer to **anyone actively involved in the creation and release of software, regardless of role or title. Additionally, we will define "small" organizations as having < 100 employees, "mid-sized" organizations as having 100-999 employees, and "large" organizations as having 1,000+ employees. Finally, due to the low sample size of respondents claiming to work at mid-sized organizations (n=13), our confidence levels regarding analysis based on this subset of respondents is lower than usual.*

Major Research Targets

In our 2025 Software Supply Chain Security survey, we aimed to gather data regarding various topics related to the following major research targets:

- 1. Threat and risk detection
- 2. DevSecOps and software integrity

In this report, we review some of our primary research findings. Many secondary findings of interest are not included.

Research Target One: Threat and Risk Detection

Our research related to **threat** and **risk detection** centered around three key topic areas:

- Supply chain and platform threats
- Software bills of materials
- AI-augmented threat defense

Supply Chain and Platform Threats

We asked the following questions:

- Which of the following software supply chain threats impact your organization?
- What are your biggest challenges related to managing security across a complex toolchain?*
- How would you rate your organization's ability to detect and respond to incidents across the software supply chain?

**This question was only asked to respondents who selected "Toolchain complexity" for the question, "Which of the following software supply chain threats impact your organization?"*

Results:

Figure 1. Software supply chain threats [n=95]

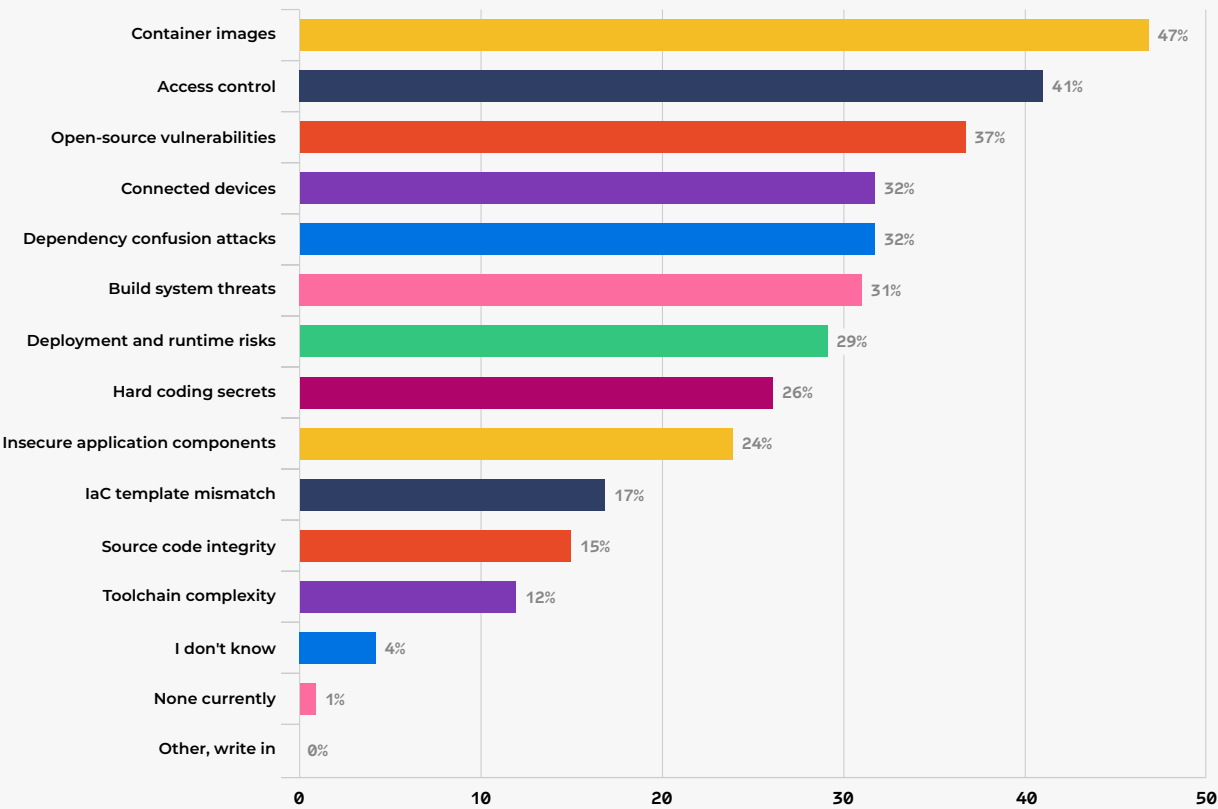
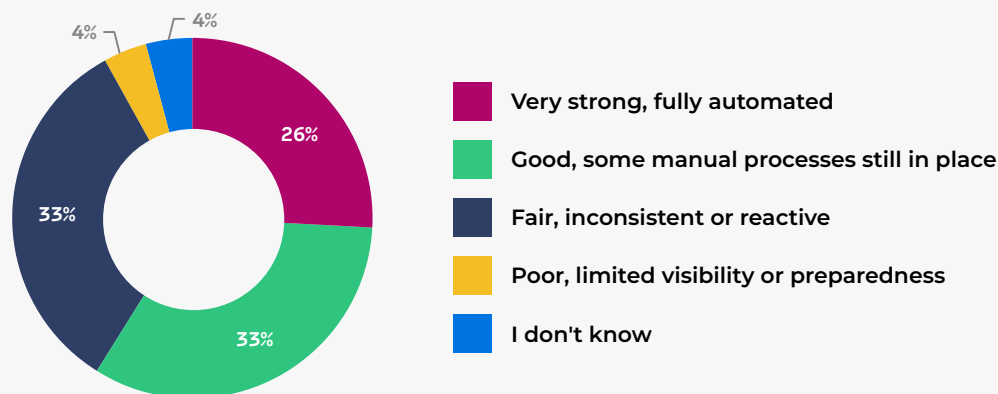


Figure 2. Opinion: organizations' ability to respond to supply chain incidents [n=94]



OBSERVATIONS

A large majority of respondents (95%) said that their organization is impacted by at least one of the 12 supply chain threats provided as answer options, and over two-thirds (72%) selected three or more threats. The most commonly selected threats were "Container images," "Access control," and "Open-source vulnerabilities." Compared to the results of our 2024 [Enterprise Security](#) survey, both "Dependency confusion attacks" and "Container images" had significantly higher response rates.

Furthermore, while the change in response rates from our 2024 survey was just within our margin of error, "Insecure application components," "Open-source vulnerabilities," and "Source code integrity" all saw a consistent and significant fall from our 2023 [Enterprise Security](#) survey. Further year-over-year (YOY) data is available in Table 1.

Segmenting the data by respondents' organization size, we noted the following (additional details in Table 2):

- Respondents at **large organizations** were more likely than others to say they did not know what supply chain threats impact their organization.
- Respondents at **mid-sized organizations** were more likely than others to select "Open-source vulnerabilities," "Deployment and runtime risks," and "Hard coding secrets" as supply chain threats, and less likely to select "Toolchain complexity."
- Respondents at **small organizations** were more likely than others to select "Connected devices" and "Dependency confusion attacks" as supply chain threats, and less likely to select "Access control," "Open-source vulnerabilities," and "Build system threats."

Response rates were fairly even between respondents who said they would rate their organization's ability to detect and respond to incidents across the software supply chain "Very strong, fully automated," those who said they would rate it "Good, some manual processes still in place," and those who said they would rate it "Fair, inconsistent or reactive." Very few respondents rated their organization's supply chain incident detection/response ability as "Poor, limited visibility or preparedness" or said that they did not know.

Segmented by organization size, we found that respondents at large organizations were much more likely than others to rate their organization's ability to detect/respond to supply chain incidents "Very strong" and less likely to rate it "Good" or "Fair." Further details are available in Table 3.*

**Note: We use the results of this question to segment other responses later in this report, referring to this rating as the "supply chain incidence response rating."*

SEE ADDITIONAL TABLES ON NEXT PAGE

ADDITIONAL TABLES

Table 1. Software supply chain threats: 2023–2025

Threat	2023	2024	2025	% Change YOY
Container images	35%	35%	47%	+12%
Access control	64%	44%	41%	-3%
Open-source vulnerabilities	57%	47%	37%	-10%
Connected devices	39%	27%	32%	+5%
Dependency confusion attacks	22%	18%	32%	+14%
Build system threats	29%	27%	31%	+4%
Deployment and runtime risks	29%	25%	29%	+4%
Hard coding secrets	42%	28%	26%	-2%
Insecure app components	44%	34%	24%	-10%
IaC template mismatch	18%	7%	17%	+10%
Source code integrity	35%	25%	15%	-10%
Toolchain complexity	-	-	12%	-
I don't know	8%	6%	4%	-2%
None currently	7%	8%	1%	-7%
Other, write in	4%	3%	0%	-3%
<i>n</i> =	147	126	95	

Table 2. Software supply chain threats by organization size*

Threat	Organization Size			Overall
	1-99	100-999	1,000+	
Container images	49%	46%	47%	47%
Access control	31%	54%	50%	41%
Open-source vulnerabilities	29%	54%	43%	37%
Connected devices	53%	0%	10%	32%
Dependency confusion attacks	37%	23%	27%	32%
Build system threats	25%	38%	37%	31%
Deployment and runtime risks	25%	46%	30%	29%
Hard coding secrets	24%	38%	27%	26%
Insecure app components	25%	31%	20%	24%
IaC template mismatch	16%	15%	20%	17%
Source code integrity	12%	15%	20%	15%
Toolchain complexity	12%	0%	17%	12%
I don't know	0%	0%	13%	4%
None currently	0%	0%	3%	1%
Other, write in	0%	0%	0%	0%
<i>n</i> =	51	13	30	95

*% of columns

Table 3. Ability to respond to supply chain incidents by organization size*

Rating	Organization Size			Overall
	1-99	100-999	1,000+	
Very strong, fully automated	12%	15%	55%	26%
Good, some manual processes still in place	37%	38%	24%	33%
Fair, inconsistent or reactive	42%	46%	10%	33%
Poor, limited visibility or preparedness	6%	0%	3%	4%
I don't know	4%	0%	7%	4%
n=	52	13	29	94

*% of columns

CONCLUSIONS

Developers are grappling with a rapidly shifting landscape where both emerging threats and evolving detection methods require ongoing adaptation and coordination, with software supply chain security as a shared concern, but the specific risks and perceptions vary based on organizational context.

The correlation between organization size and both threat awareness and response capability suggests that resources and infrastructure heavily influence how supply chain security is prioritized and managed. Developers at smaller organizations may face a broader set of risks with fewer established controls, while those at larger firms benefit from more advanced tooling but may be distanced from direct visibility into specific threats. The variation in confidence around incident response highlights differing levels of maturity across organizations and points to a need for more consistent practices.

Software Bills of Materials

We asked the following questions:

- How does your organization utilize software bills of materials (SBOMs) in your software development and security processes?
- How frequently is your organization's SBOM updated?*
- How is your organization's SBOM shared or made accessible?*
- What formats does your organization primarily use for its SBOMs?*

*These questions were only asked to respondents who answered "We generate SBOMs and use them as a compliance and/or security tool," "We generate and sign SBOMs," or "We generate SBOMs" for the question "How does your organization utilize SBOMs in your software development and security processes?"

Results:

Figure 3. Organizations' SBOM use/maturity [n=99]



Figure 4. Update frequency of organizations' SBOMs [n=57]

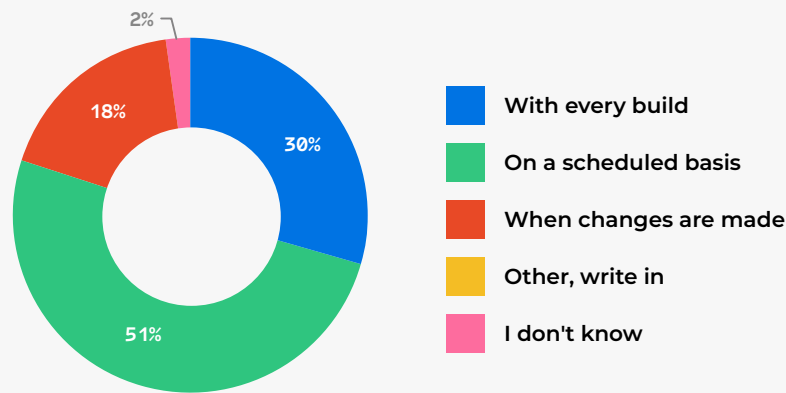


Figure 5. Accessibility/availability of organizations' SBOMs [n=59]

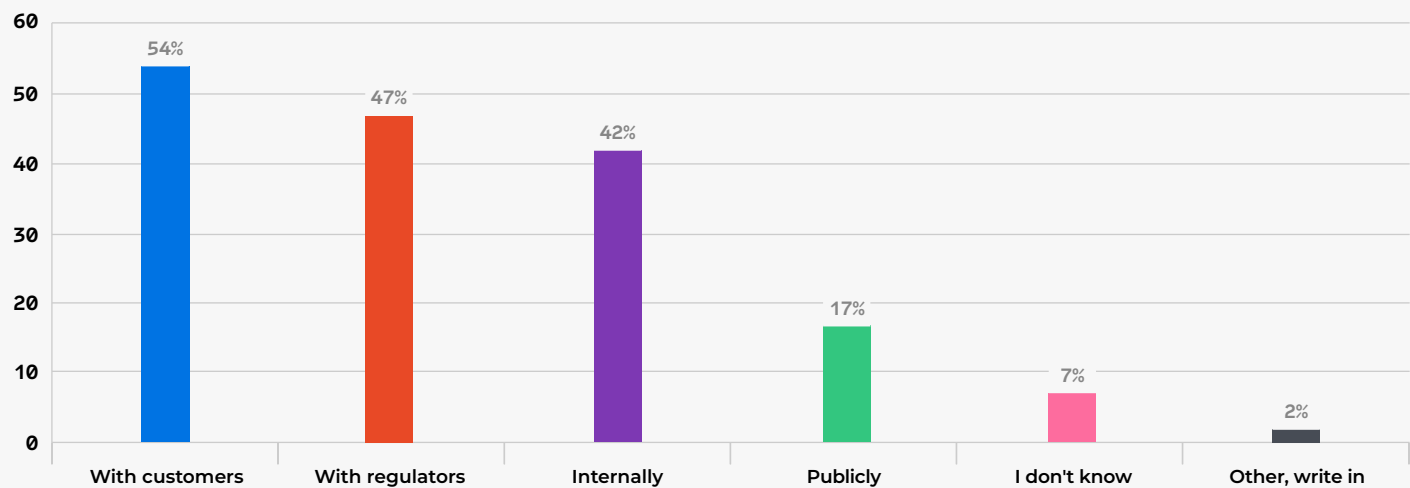
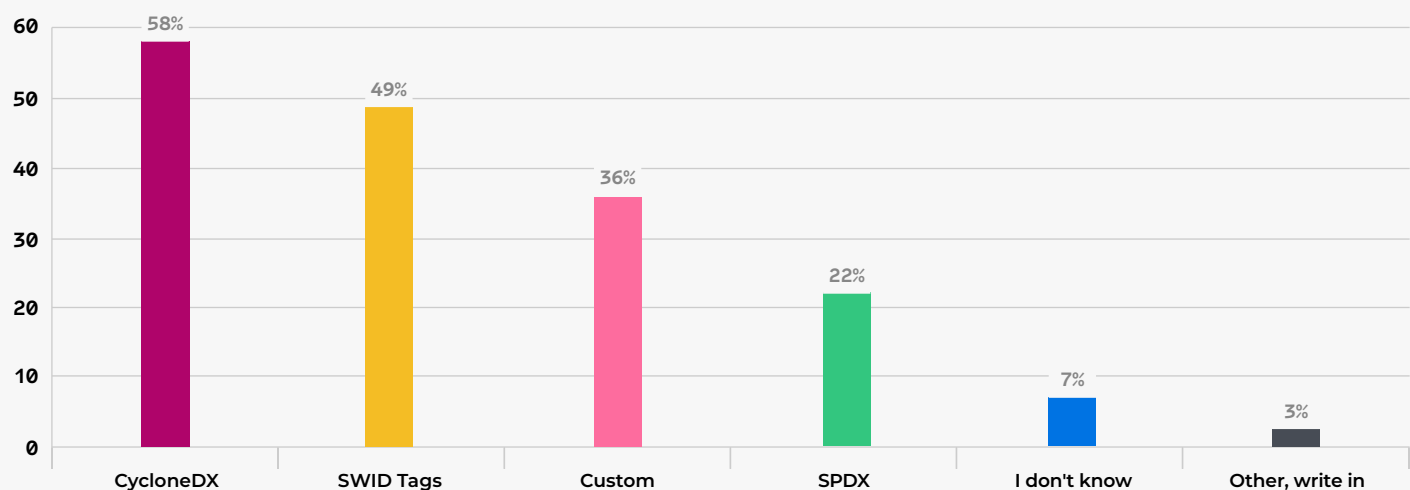


Figure 6. Primary formats for organizations' SBOMs [n=59]



OBSERVATIONS

Most respondents (63%) indicated that their organization has at least a basic level of SBOM adoption, and about a quarter of respondents reported their organization as having an advanced level, selecting "We generate SBOMs and use them as a compliance and/or security tool."

Segmenting results by organization size, we noted the following (more details in Table 4):

- Respondents at **large organizations** were much more likely to say their organization uses SBOMs as compliance/security tools.
- Respondents at **mid-sized organizations** were more likely to say their organization doesn't use SBOMs, both with and without plans to adopt SBOMs in the future.
- Respondents at **small organizations** were more likely than others to say "We generate SBOMs" and "We generate and sign SBOMs," and they were less likely to say that they didn't know how their organization uses SBOMs.

Furthermore, we found that respondents who said their organization uses SBOMs as compliance/security tools were much more likely to rate their organization's supply chain incident response rating as "Very strong" (46%) compared to all other respondents (16%). Further data on these results can be made available on request.

The majority of respondents (51%) said that their organization updates their SBOM "On a scheduled basis (e.g., weekly, monthly)," while almost a third of respondents (30%) said their organization updates their SBOM "With every build."

Respondents at mid-sized organizations were more likely than others to say their organization updates their SBOM "When changes are made" and less likely to say their SBOM is updated "On a scheduled basis."* Respondents at small organizations were more likely than others to say their organization updates its SBOM "On a scheduled basis."

Additionally, respondents who said their organization updates SBOMs "With every build" were more likely to give their organization a supply chain incident response rating of "Very good" compared to other respondents. Details on these segmented results are available in Tables 5 and 6, respectively.

Most respondents (92%) said their organization shares their SBOM with one of the four options we provided — "With customers," "With regulators," "Internally," or "Publicly" — and almost half of respondents (46%) selected two or more of those options. The most commonly selected method of sharing organizations' SBOMs was "With customers."

Respondents at large organizations were more likely than others to say they didn't know how their organization's SBOM is shared. Respondents at mid-sized organizations were less likely than others to say their organization shares their SBOM "Publicly."* And respondents at small organizations were more likely than others to say their organization shares their SBOM "With customers" and "With regulators," and less likely than others to say it is shared "Internally." Respondents who said that their organization shares their SBOM "Internally" were more likely than others to say their organization has a "Very good" supply chain incident response rating (details on segmented results can be found in Tables 7 and 8, respectively).

Almost half of respondents (46%) said their organization uses at least two of the four SBOM formats we listed — "CycloneDX," "SWID Tags," "SPDX," and "Custom." The most commonly selected format was "CycloneDX," with a majority of respondents saying their organization uses that format, and nearly half of respondents said their organization uses "SWID Tags."

Segmented by respondents' organization size, we observed the following (further details in Table 9):

- Respondents at **large organizations** were less likely than others to say their organization uses "CycloneDX" and "SWID Tags" and more likely than others to say they do not know what format their organization uses for SBOMs.
- Respondents at **mid-sized organizations** were less likely than others to say their organization uses a "Custom" SBOM format.
- Respondents at **small organizations** were more likely than others to say their organization uses "SWID Tags" and less likely to say they use "SPDX."

We also found that respondents who said their organization uses "SPDX" were more likely to rate their organization's supply chain incident response as "Very good" (additional data is available on request).

**Note: For mid-sized organizations in these segmentations, because of the extremely small sample size (n=3), confidence in these correlations is much lower than normal (< 80%).*

ADDITIONAL TABLES

Table 4. SBOM use/maturity by organization size*

SBOM Use	Organization Size			Overall
	1-99	100-999	1,000+	
We generate SBOMs and use them as a compliance and/or security tool	16%	8%	47%	24%
We generate and sign SBOMs	24%	8%	3%	16%
We generate SBOMs	33%	8%	10%	22%
We don't use SBOMs but plan to adopt them	10%	31%	10%	12%
We don't use SBOMs and have no plans to adopt them	10%	23%	7%	10%
I don't know	8%	23%	23%	15%
n=	51	13	30	99

*% of columns

Table 5. SBOM update frequency by organization size*

Frequency	Organization Size			Overall
	1-99	100-999	1,000+	
With every build	25%	33%	35%	30%
On a scheduled basis	61%	0%	41%	51%
When changes are made	14%	67%	18%	18%
I don't know	0%	0%	6%	2%
n=	36	3	17	57

*% of columns

Table 6. Supply chain incident response rating by SBOM update frequency*

Frequency	Incident Response Rating					n=
	Very strong	Good	Fair	Poor	I don't know	
With every build	41%	24%	29%	0%	0%	17
On a scheduled basis	14%	38%	45%	0%	3%	29
When changes are made	30%	50%	20%	0%	0%	10
I don't know	0%	0%	100%	0%	0%	1
Overall	25%	35%	37%	0%	2%	57

*% of rows

Table 7. SBOM accessibility/availability by organization size*

SBOM Shared	Organization Size			Overall
	1-99	100-999	1,000+	
Internally	28%	67%	72%	42%
With customers	72%	33%	28%	54%
With regulators	58%	33%	33%	47%
Publicly	19%	0%	11%	17%
Other, write in	3%	0%	0%	2%
I don't know	0%	0%	17%	7%
n=	36	3	18	59

*% of columns

Table 8. Supply chain incident response rating by SBOM accessibility/availability*

SBOM Shared	Incident Response Rating					n=
	Very strong	Good	Fair	Poor	I don't know	
Internally	44%	28%	28%	0%	0%	25
With customers	22%	38%	41%	0%	0%	32
With regulators	25%	29%	46%	0%	0%	28
Publicly	10%	50%	30%	0%	0%	10
Other, write in	0%	0%	100%	0%	0%	1
I don't know	0%	25%	25%	0%	25%	4
Overall	24%	36%	36%	0%	2%	59

*% of rows

Table 9. Primary SBOM formats by organization size*

SBOM Format	Organization Size			Overall
	1-99	100-999	1,000+	
CycloneDX	72%	67%	33%	58%
SWID Tags	69%	33%	11%	49%
Custom	39%	0%	33%	36%
SPDX	17%	33%	28%	22%
I don't know	0%	0%	17%	7%
Other, write in	0%	0%	6%	3%
n=	36	3	18	59

*% of columns

CONCLUSIONS

SBOM practices appear to be a growing standard across organizations, though implementation levels and usage maturity vary significantly by organization size and structure. Developers at smaller organizations tend to report more awareness and engagement with SBOM practices, possibly due to closer involvement in day-to-day tooling and security processes. Larger organizations, while more likely to leverage SBOMs for compliance and security, often show gaps in internal awareness of SBOM formats and sharing methods.

The correlations we observed between advanced SBOM use and higher confidence in supply chain incident response capabilities points toward SBOM maturity being both a driver and a reflection of broader organizational readiness in supply chain security. Overall, consistent updating, internal sharing, and the use of standardized formats like SPDX and CycloneDX seem to be strong indicators of proactive and well-integrated supply chain security practices.

AI-Augmented Threat Defense

We asked the following questions:

- *In what ways are you implementing AI/ML to detect and/or mitigate risk at your organization?*
- *Has your organization adopted AI/ML-powered tools for threat detection within the software supply chain?*
- *What benefits have you seen from the use of AI/ML for threat detection?**

**This question was only asked to respondents who selected "Yes, widely adopted across systems" or "In limited or pilot use" for the question, "Has your organization adopted AI/ML-powered tools for threat detection within the software supply chain?"*

SEE RESULTS ON NEXT PAGE

Results:

Figure 7. Organizations' AI/ML use cases for detecting/mitigating risk [n=100]

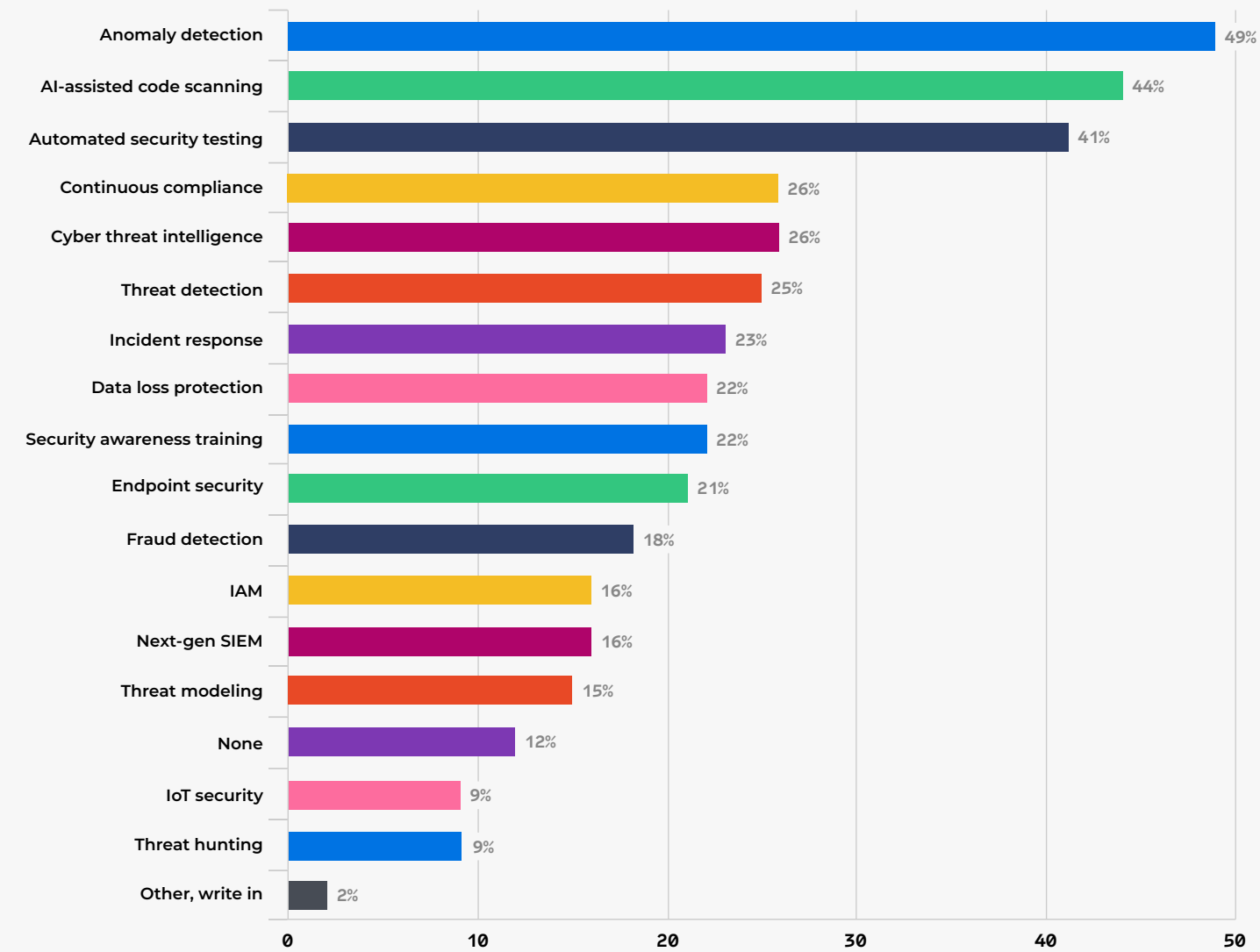
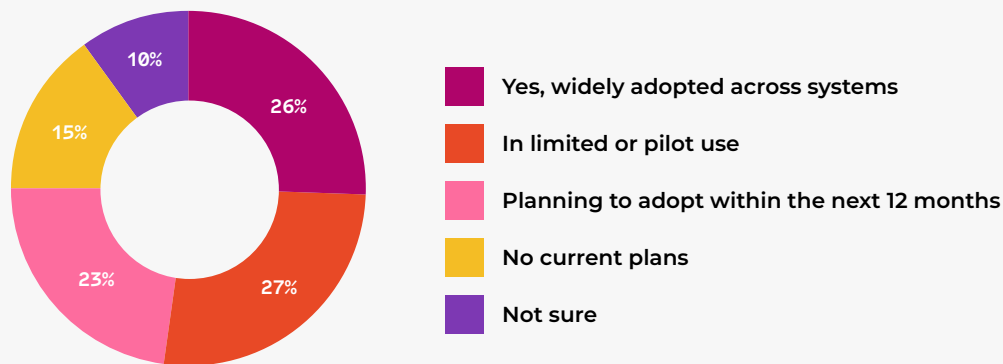
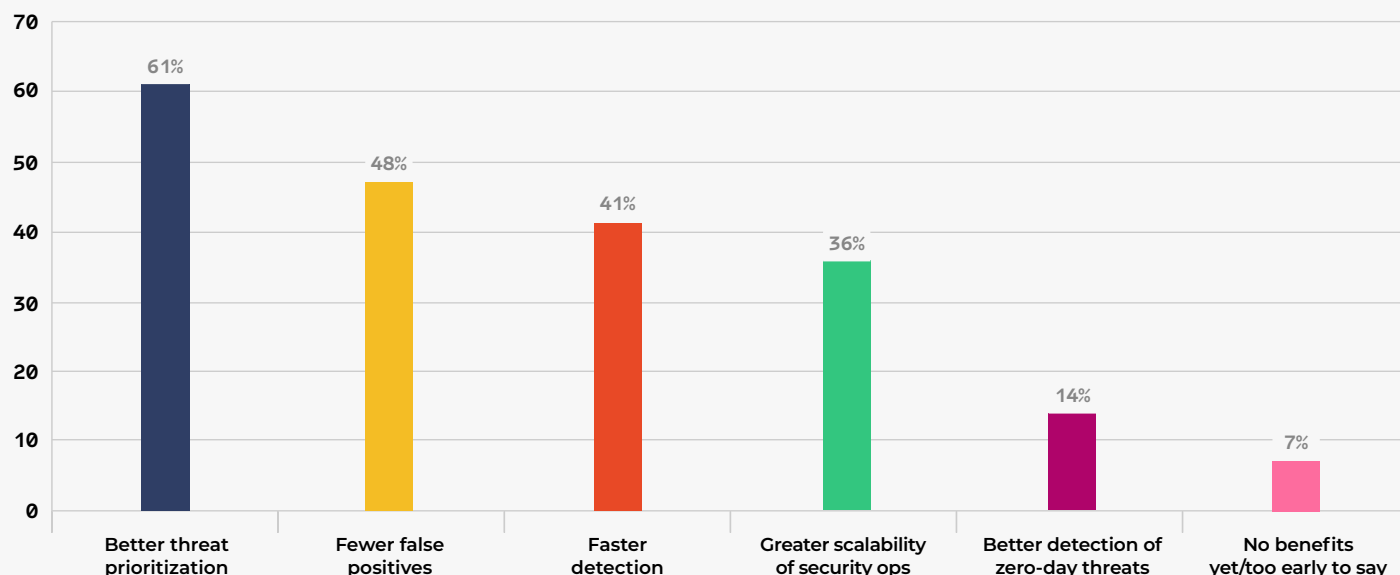


Figure 8. Organizations' adoption of AI/ML tools for threat detection [n=94]



FIGURES CONTINUE ON NEXT PAGE

Figure 9. Opinion: benefits of AI/ML for threat detection [n=44]



OBSERVATIONS

86% of respondents selected at least one of the 16 AI/ML use cases for detecting or mitigating risks used at their organization, and over two-thirds of respondents (70%) selected three or more. "Anomaly detection," "AI-assisted code scanning," and "Automated security testing (e.g., DAST, pen tests)" were the most commonly selected use cases, and were chosen significantly more often than others.

With the results segmented by respondents' organization size, we found:

- Respondents at **large organizations** were more likely than others to select several use cases, including "AI-assisted code scanning," "Continuous compliance," "Data loss protection," and "Fraud detection."
- Respondents at **mid-sized organizations** were less likely than others to select "Anomaly detection," "Next-gen SIEM," and "Threat modeling."
- Respondents at **small organizations** were less likely than others to select "AI-assisted code scanning," "Fraud detection," "Endpoint security," and "Incident response."

Additional data is available in Table 10.

About a quarter of respondents said that their organization has "widely adopted" AI/ML threat detection tools, about another quarter said their organization has adopted these tools "In limited or pilot use," and about another quarter said their organization is "Planning to adopt within the next 12 months."

Segmenting the results by organization size, we noted the following (details in Table 11):

- Respondents at **large organizations** were more likely than others to say their organization has "widely adopted [these tools] across systems," though they were also more likely than others to say they did not know. They were less likely to say their organization is in the "Planning" stages of adoption.
- Respondents at **mid-sized organizations** were more likely than others to say their organization is "Planning to adopt within the next 12 months" and more likely to say their organization has "No current plans" for adoption. They were less likely than others to say their organization has "widely adopted" these tools.
- Respondents at **small organizations** were more likely than others to say their organization had adopted AI/ML threat detection tools "In limited or pilot use."

Respondents who said their organization used AI/ML threat detection tools "widely... across systems" were more likely than others to say their organization has a "Very strong" supply chain incident response rating.

Of the benefits respondents saw from AI/ML threat detection tools, "Better threat prioritization" was the most commonly selected. Almost all respondents (93%) said they have seen at least one of the five benefits we provided as options, and about two-thirds (66%) said they have seen two or more.

Segmenting the data by reported organization size, we observed the following (details available in Table 13):

- Respondents at **large organizations** were more likely than others to see "Faster detection" and "Better threat prioritization" as benefits of AI/ML threat detection.
- Respondents at **mid-sized organizations** were more likely than others to find "Better detection of zero-day threats" as a benefit of AI/ML threat detection, but also more likely than others to find no benefits. They were less likely than others to select "Faster detection," "Better threat prioritization," and "Greater scalability of security ops."*
- Respondents at **small organizations** were more likely than others to find "Fewer false positives" and "Greater scalability of security ops" as benefits of AI/ML threat detection tools.

**Note: For mid-sized organizations in this segmentation, because of the extremely small sample size (n=3), confidence in these correlations is much lower than normal (< 80%).*

ADDITIONAL TABLES

Table 10. AI/ML use cases for detecting/mitigating risk by organization size*

Use Case	Organization Size			Overall
	1-99	100-999	1,000+	
Anomaly detection	50%	31%	60%	49%
AI-assisted code scanning	31%	46%	67%	44%
Automated security testing	44%	31%	37%	41%
Continuous compliance	23%	23%	33%	26%
Cyber threat intelligence (CTI)	25%	15%	33%	26%
Threat detection	23%	23%	30%	25%
Incident response	8%	38%	43%	23%
Data loss protection	17%	15%	37%	22%
Security awareness training	23%	15%	23%	22%
Endpoint security	8%	31%	43%	21%
Fraud detection	8%	23%	37%	18%
IAM	8%	15%	30%	16%
Next-gen SIEM	21%	0%	17%	16%
Threat modeling	13%	0%	27%	15%
None	8%	15%	13%	12%
IoT security	12%	0%	10%	9%
Threat hunting	10%	0%	13%	9%
Other, write in	2%	0%	3%	2%
n=	52	13	30	100

*% of columns

TABLES CONTINUE ON NEXT PAGE

Table 11. Adoption of AI/ML tools for threat detection by organization size*

Tool Adoption	Organization Size			Overall
	1-99	100-999	1,000+	
Yes, widely adopted across systems	21%	8%	40%	26%
In limited or pilot use	35%	17%	17%	27%
Planning to adopt within the next 12 months	27%	50%	7%	23%
No current plans	13%	25%	13%	15%
I don't know	4%	0%	23%	10%
<i>n</i> =	52	12	30	94

*% of columns

Table 12. Supply chain incident response rating by adoption of AI/ML tools for threat detection*

Tool Adoption	Incident Response Rating					<i>n</i> =
	Very strong	Good	Fair	Poor	I don't know	
Yes, widely adopted across systems	54%	33%	13%	0%	0%	24
In limited or pilot use	24%	48%	20%	8%	0%	25
Planning to adopt within the next 12 months	9%	36%	50%	0%	5%	22
No current plans	14%	14%	50%	14%	7%	14
I don't know	13%	13%	50%	0%	25%	8
Overall	26%	33%	32%	4%	4%	93

*% of rows

Table 13. Opinion: benefits of AI/ML for threat detection by organization size*

Benefit	Organization Size			Overall
	1-99	100-999	1,000+	
Better threat prioritization	57%	33%	77%	61%
Fewer false positives	57%	33%	31%	48%
Faster detection	25%	0%	85%	41%
Greater scalability of security ops	43%	0%	31%	36%
Better detection of zero-day threats	14%	33%	8%	14%
No benefits yet/too early to say	4%	33%	8%	7%
<i>n</i> =	28	3	13	44

*% of columns

CONCLUSIONS

AI/ML-powered threat detection tools are being adopted in multiple ways across organizations, with notable differences in use case focus, scale of implementation, and perceived benefits, depending on the organization. Developers in large organizations report the widest adoption of AI/ML tools and tend to associate them with a broader range of use cases and operational benefits, particularly for threat prioritization and detection speed. Smaller organizations, meanwhile, more often engage in limited or pilot use, finding value in outcomes like reducing false positives and scaling security operations.

Across the board, increased AI/ML adoption tends to correlate with higher reported confidence in supply chain incident detection and response, suggesting these tools have begun to play an important role in perceived organizational security readiness.

Research Target Two: DevSecOps and Software Integrity

Our research related to **DevSecOps** and **software integrity** centered around two key topic areas:

- 1. CI/CD and SDLC application security
- 2. Supply chain compliance readiness and practices

CI/CD and SDLC Application Security

We asked the following questions:

- *At what stage in the SDLC does your organization first implement security?*
- *How often does your organization scan applications to detect and identify vulnerabilities?*
- *Which of the following security tools/tests are integrated into your DevOps pipeline?*

Results:

Figure 10. SDLC stage organizations first implement security [n=99]

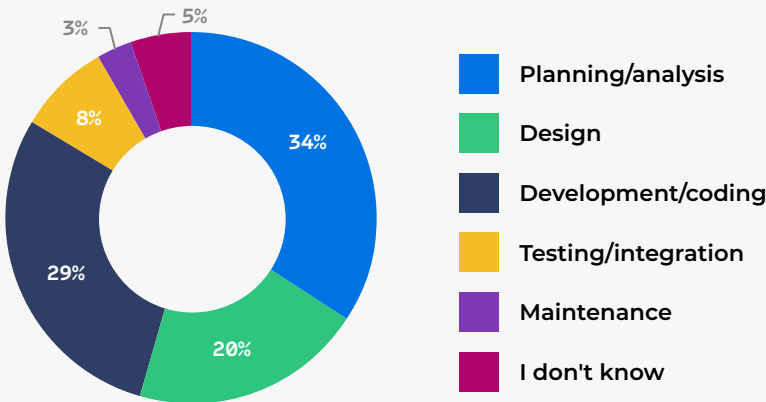
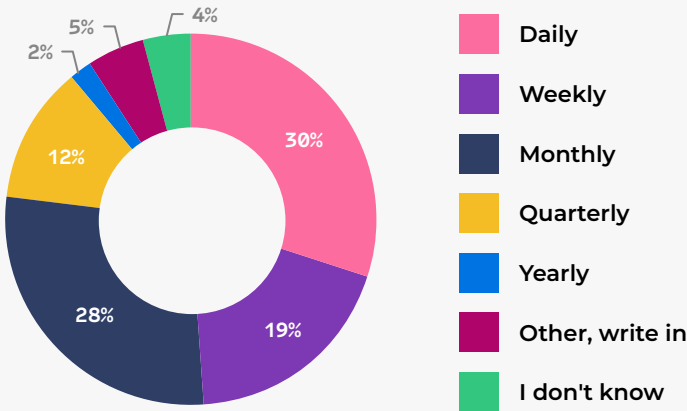
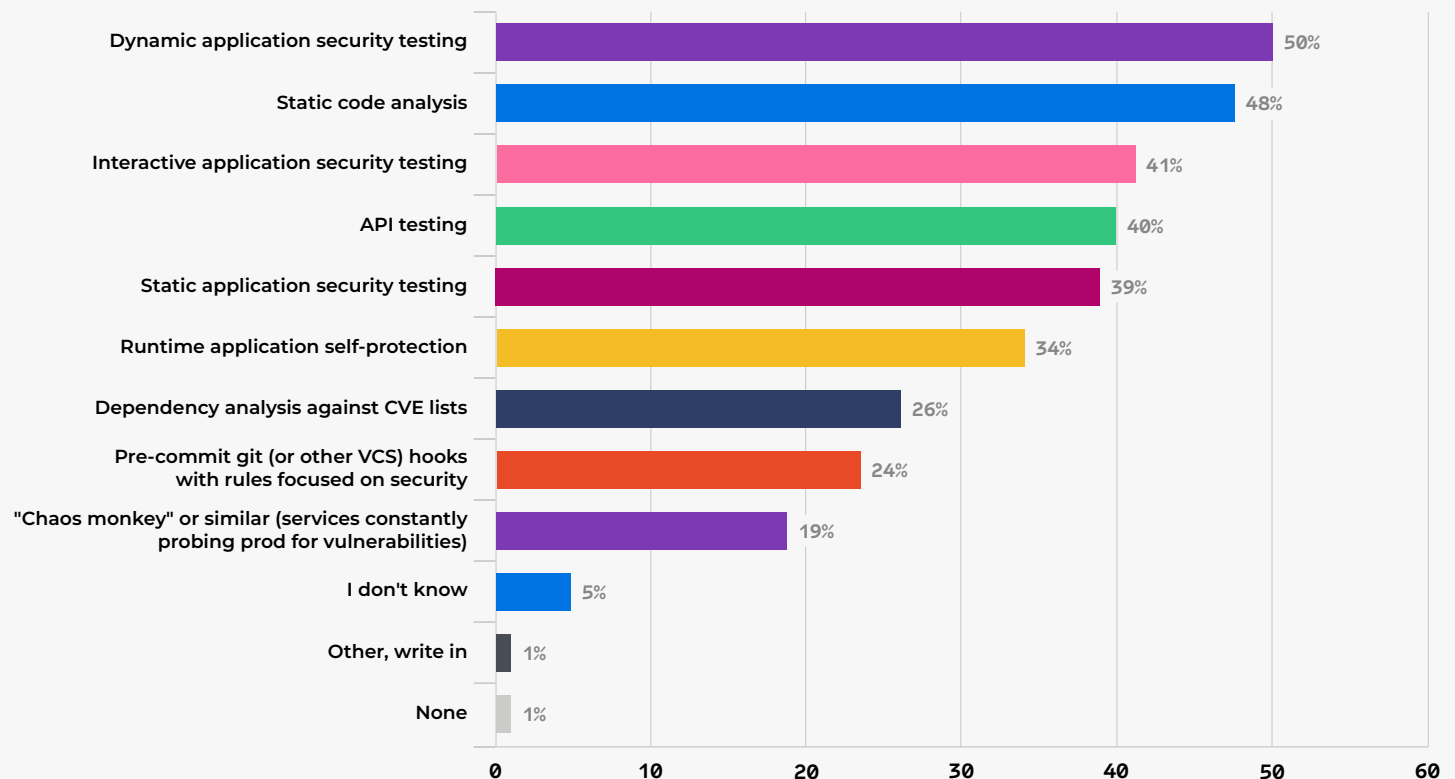


Figure 11. Frequency of organizations' scans for application vulnerabilities [n=102]



FIGURES CONTINUE ON NEXT PAGE

Figure 12. Security tools and tests in DevOps pipelines [n=94]



OBSERVATIONS

● A majority of respondents (54%) said their organization first implements security in either the "Planning/analysis" or "Design" stage of the SDLC. Compared to 2024, the response rate for "Planning/analysis" significantly increased while the response rate for "Design" decreased (data for 2023–2025 can be found in Table 14).

Respondents at large organizations were more likely than others to say their organization first implements security in the "Planning/analysis" stage and less likely to say the "Deployment/coding" stage, while respondents at small organizations were less likely to choose "Planning/analysis." Furthermore, respondents at organizations implementing security in the "Planning/analysis" stage were more likely than others to give their organization a "Very strong" supply chain incident response rating (details in Tables 15 and 16, respectively).

● Nearly half of respondents (49%) said their organization scans applications for vulnerabilities either "Daily" or "Weekly." Compared to our 2024 Security survey results, there were no significant changes in the frequency of application scans, but compared to the results from 2023, respondents selecting "Monthly" increased, and respondents selecting "I don't know" decreased (YOY data from 2023–2025 in Table 17).

Respondents at large organizations were more likely than others to say their organization performs "Daily" application vulnerability scans and less likely than others to say they perform "Monthly scans," and the inverse was the case for respondents at small organizations (details in Table 18). Respondents who said their organization scans their applications for vulnerabilities "Daily" were more likely than others to rate their organization as "Very good" in terms of supply chain incident response (further data available on request).

● Most respondents (93%) said they have at least one of the nine tools/tests we provided integrated into their DevOps pipeline, and over two-thirds of respondents (68%) said they have three or more. "Dynamic application security testing (DAST)" and "Static code analysis" were the most commonly selected tests, with each integrated into the DevOps pipeline of about half of respondents. Compared to our results from 2024, "Dynamic application security testing (DAST)," "Interactive application security testing (IAST)," and "Runtime application self-protection (RASP)" all saw significant increases in response rates. Details can be found in Table 19.

Segmented by reported organization size, we noted the following (with additional data in Table 20):

- Respondents at **large organizations** were more likely than others to say their organization implements "API testing," "SAST," and "Chaos monkey or similar" into their DevOps pipeline.
- Respondents at **mid-sized organizations** were less likely than others to say their organization implements "DAST" and "IAST."
- Respondents at **small organizations** were more likely than others to say their organization implements "IAST" and "RASP," and they were less likely to say their organization implements "Static code analysis," "API testing," "Dependency analysis against CVE lists," and "Pre-commit VCS hooks with rules focused on security."

ADDITIONAL TABLES

Table 14. SDLC stage security first implemented: 2023–2025

SDLC Stage	2023	2024	2025
Planning/analysis	25%	20%	34%
Design	28%	37%	20%
Development/coding	21%	22%	29%
Testing/integration	8%	13%	8%
Maintenance	8%	1%	3%
I don't know	10%	7%	5%
<i>n=</i>	145	123	94

Table 15. SDLC stage security first implemented by organization size*

SDLC Stage	Organization Size			Overall
	1-99	100-999	1,000+	
Planning/analysis	19%	31%	63%	34%
Design	25%	23%	13%	20%
Development/coding	37%	31%	20%	29%
Testing/integration	12%	8%	0%	8%
Maintenance	4%	0%	0%	3%
I don't know	4%	8%	3%	5%
<i>n=</i>	50	13	30	94

*% of columns

Table 16. Supply chain incident response rating by SDLC stage security first implemented*

SDLC Stage	Incident Response Rating					<i>n=</i>
	Very strong	Good	Fair	Poor	I don't know	
Planning/analysis	42%	36%	9%	3%	9%	33
Design	30%	30%	40%	0%	0%	20
Development/coding	10%	31%	52%	7%	0%	29
Testing/integration	0%	43%	57%	0%	0%	7
Maintenance	0%	50%	50%	0%	0%	2
I don't know	33%	0%	0%	33%	33%	3
Overall	26%	33%	33%	4%	4%	94

*% of rows

Table 17. Application vulnerability scan frequency: 2023–2025

Frequency	2023	2024	2025
Daily	30%	25%	30%
Weekly	21%	20%	19%
Monthly	15%	21%	28%
Quarterly	10%	10%	12%
Yearly	4%	7%	2%
Other, write in	3%	7%	5%
I don't know	17%	11%	4%
<i>n</i> =	145	122	98

Table 18. Application vulnerability scan frequency by organization size*

Frequency	Organization Size			Overall
	1-99	100-999	1,000+	
Daily	14%	31%	60%	30%
Weekly	24%	15%	17%	19%
Monthly	43%	15%	3%	28%
Quarterly	16%	15%	7%	12%
Yearly	2%	8%	0%	2%
Other, write in	0%	15%	7%	5%
I don't know	2%	0%	7%	4%
<i>n</i> =	51	13	30	98

*% of columns

Table 19. Security tools and tests in DevOps pipelines: 2024–2025

Tool/Test	2024	2025
Dynamic application security testing (DAST)	34%	50%
Static code analysis	44%	48%
Interactive application security testing (IAST)	23%	41%
API testing	42%	40%
Static application security testing (SAST)	40%	39%
Runtime application self-protection (RASP)	20%	34%
Dependency analysis against CVE lists	34%	26%
Pre-commit VCS hooks with rules focused on security	21%	24%
"Chaos monkey" or similar	13%	19%
I don't know	6%	5%
Other, write in	1%	1%
None	6%	1%
<i>n</i> =	125	94

TABLES CONTINUE ON NEXT PAGE

Table 20. Security tools and tests in DevOps pipelines by organization size*

Tool/Test	Organization Size			Overall
	1-99	100-999	1,000+	
Dynamic application security testing (DAST)	54%	38%	50%	50%
Static code analysis	28%	77%	70%	48%
Interactive application security testing (IAST)	52%	15%	37%	41%
API testing	24%	38%	70%	40%
Static application security testing (SAST)	24%	31%	67%	39%
Runtime application self-protection (RASP)	42%	23%	27%	34%
Dependency analysis against CVE lists	10%	38%	47%	26%
Pre-commit VCS hooks with rules focused on security	10%	38%	43%	24%
"Chaos monkey" or similar	14%	15%	30%	19%
I don't know	6%	0%	7%	5%
Other, write in	2%	0%	0%	1%
None	2%	0%	0%	1%
n=	50	13	30	94

*% of columns

CONCLUSIONS

Security is increasingly shifting left, with early-stage implementation being linked to stronger incident response capabilities. Frequent application scanning also aligns with more advanced supply chain security. Most organizations now integrate multiple security tools into their DevOps pipelines, often combining DAST, SAST, and other methods to ensure broader coverage. Larger organizations tend to use a wider array of tools, including API testing and chaos engineering, while smaller organizations more often adopt emerging practices like RASP and IAST. These differences suggest varying security strategies are shaped by organizational size and resource availability.

Supply Chain Compliance Readiness and Practices

We asked the following questions:

- How prepared is your organization for meeting evolving regulatory compliance standards (e.g., GDPR, CCPA)?
- How often do you review or update your software supply chain to remain compliant with regulations?
- Which compliance-related practices are currently implemented at your organization?
- Which IAM practices are in place across your software development and deployment environments?

Results:

Figure 13. Organizations' preparedness for meeting compliance standards [n=94]

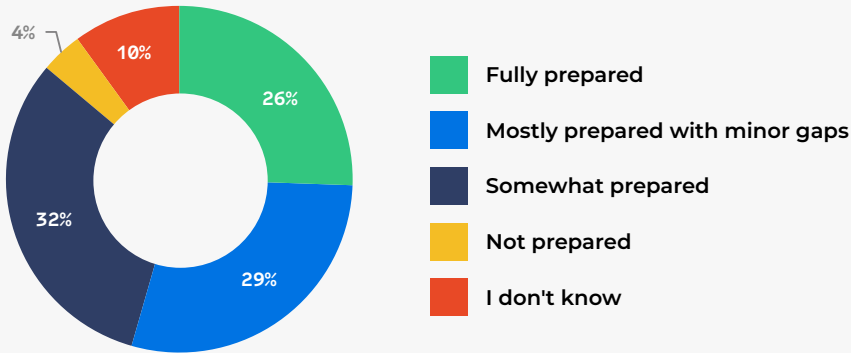


Figure 14. Frequency of review/updates to supply chain for regulatory compliance [n=93]

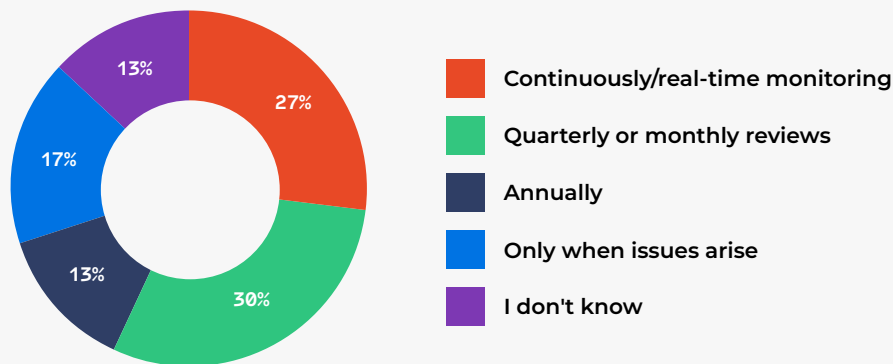
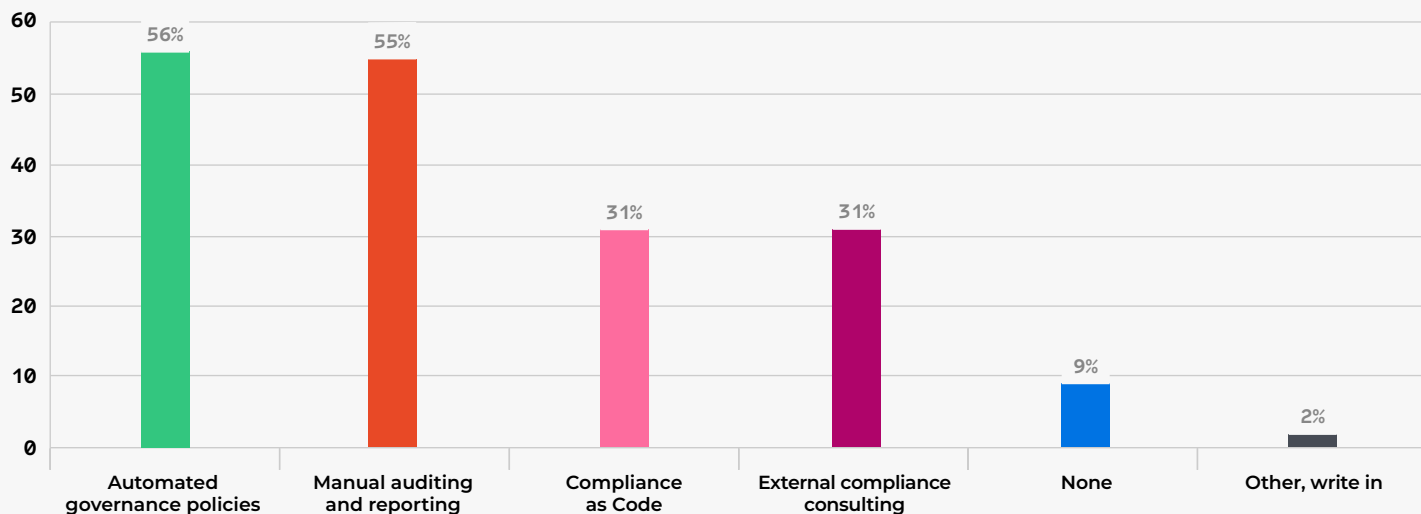


Figure 15. Organizations' compliance-related practices [n=95]



OBSERVATIONS

Most respondents (87%) said their organization is at least "Somewhat prepared" for meeting regulatory compliance standards, though only about a quarter of respondents (26%) said their organization is "Fully prepared." Almost no respondents (4%) said their organization is "Not prepared" at all.

Segmenting responses by organization size, we observed the following (further details in Table 21):

- Respondents at **large organizations** were more likely than others to say their organization is "Fully prepared" for meeting compliance standards, but they were also more likely than others to respond "I don't know." They were less likely than others to say their organization is "Somewhat prepared."
- Respondents at **mid-sized organizations** were more likely than others to say their organization is "Not prepared."*
- Respondents at **small organizations** were more likely than others to say their organization is "Somewhat prepared" and less likely than others to say they are "Fully prepared."

Respondents at "Fully prepared" organizations were much more likely than others to rate their organization "Very good" with regard to supply chain incident response (additional data available upon request).

**Note: For the "Not prepared" option in this segmentation, because of the extremely small sample size (n=4), confidence in these correlations is much lower than normal (< 80%).*

Over half of respondents (57%) said their organization reviews or updates its supply chain to remain regulation compliant either "Continuously/[using] real-time monitoring" or with "Quarterly or monthly reviews." Segmented by respondents' organization size, we noted (details in Table 22):

- Respondents at **large organizations** were more likely than others to select "Continuously/real-time monitoring," but also more likely to say they did not know the frequency of compliance reviews. They were less likely than others to say their organization did these reviews "Quarterly or monthly" or "Annually."
- Respondents at **small organizations** were more likely than others to say they did these reviews "Quarterly or monthly."

Respondents at organizations with continuous/real-time monitoring of supply chain compliance were more likely than others to give a "Very good" supply chain incident response rating (additional data available upon request).

A large majority of respondents (89%) selected at least one of the four compliance-related practices we listed as options, and over half of respondents (60%) selected two or more. "Automated governance policies" and "Manual auditing and reporting" were the most commonly selected practices by a significant margin, each being selected by more than half of respondents.

Segmenting the results regarding compliance-related practices by respondents' organization size, we found the following (further details in Table 23):

- Respondents at **large organizations** were more likely than others to say their organization implements "Compliance as Code" and less likely to say they use "Manual auditing and reporting."
- Respondents at **mid-sized organizations** were more likely than others to select "Manual auditing and reporting" and less likely to select "Compliance as Code" and "Automated governance policies."
- Respondents at **small organizations** were less likely than others to say their organization implements "External compliance consulting."

Additionally, respondents who said their organization implements "Compliance as Code" were more likely than others to say their organization has a "Very good" supply chain incident response rating (additional data available upon request).

ADDITIONAL TABLES

Table 21. Preparedness for meeting compliance standards by organization size*

Preparedness	Organization Size			Overall
	1-99	100-999	1,000+	
Fully prepared	12%	25%	50%	26%
Mostly prepared with minor gaps	35%	25%	20%	29%
Somewhat prepared	48%	25%	7%	32%
Not prepared	2%	17%	3%	4%
I don't know	4%	8%	20%	10%
n=	52	12	30	94

*% of columns

Table 22. Frequency of supply chain review/updates for regulatory compliance by organization size*

Frequency	Organization Size			Overall
	1-99	100-999	1,000+	
Continuously/real-time monitoring	15%	25%	48%	27%
Quarterly or monthly reviews	42%	25%	10%	30%
Annually	19%	17%	0%	13%
Only when issues arise	19%	25%	10%	17%
I don't know	4%	8%	31%	13%
n=	52	12	29	93

*% of columns

Table 23. Compliance-related practices by organization size*

Practice	Organization Size			Overall
	1-99	100-999	1,000+	
Compliance as Code	23%	8%	53%	31%
Automated governance policies	57%	33%	63%	56%
Manual auditing and reporting	62%	83%	30%	55%
External compliance consulting	23%	42%	40%	31%
Other, write in	2%	0%	3%	2%
None	13%	8%	3%	9%
n=	53	12	30	95

*% of columns

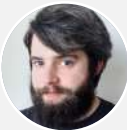
CONCLUSIONS

Most organizations appear at least moderately confident in their regulatory readiness, though relatively few respondents reported being fully prepared — particularly in smaller organizations, which may lack dedicated resources or mature compliance programs. Larger organizations are more likely to implement advanced practices like Compliance as Code and continuous monitoring, which may reflect their greater regulatory exposure and more complex supply chains.

These practices are also associated with stronger supply chain incident response, suggesting that deeper automation and real-time oversight may play a role in improving security outcomes. In contrast, smaller organizations more frequently rely on manual reviews or periodic assessments, which may reflect resource constraints or slower adoption of compliance tooling. Overall, organizations adopting proactive, automated compliance practices appear better positioned to detect and respond to supply chain threats.

Future Research

Our analysis here only touched the surface of the available data, and we will look to refine and expand our software supply chain security research as we produce future Trend Reports. Please contact publications@dzone.com if you would like to discuss any of our findings or supplementary data. 🎲



G. Ryan Spain
🐙 @grspain
🐱 @grspain
🦊 @grspain
🌐 gryanspain.com

G. Ryan Spain lives on a beautiful two-acre farm in McCalla, Alabama with his lovely wife and adorable dog. He is a polyglot software engineer with an MFA in poetry, a die-hard Emacs fan and Linux user, a lover of The Legend of Zelda, a journeyman data scientist, and a home cooking enthusiast. When he isn't programming, he can often be found playing Balatro with a glass of red wine or a cold beer.

SOFTWARE SECURITY

TREAT OR THREAT

Leveraging SBOMs to Control Your Supply Chain Chaos

Software supply chain security is on the rise as systems advance and hackers level up their tactics. Gone are the days of fragmented security checkpoints and analyzing small pieces of the larger software security puzzle. Now, software bills of materials are becoming the required norm instead of an afterthought. So the question is: *Are supply chains and SBOMs a sweet pairing or a sticky solution?* Dodge the sour code, unwrap the SBOM mystery flavors, and follow the sweet trail toward a strengthened security posture.



Required
SBOM Component



Recommended
SBOM Component



Key Practices
for SBOMs

START
HERE

Supplier
name

Cookie
Code
LAND

Ensure Source Code
Security, Integrity,
and Quality

PURL &
SWID tags

Chocolate
Repo RIDGE

Protect the Codebase
Repository and Version
Controls

Automated
generation

License
information

Verification
& validation

Licorice
Library
LANE

Use Secure Open-Source
and Third-Party Packages

Component
version

Component
hash

SBOM SAVORIES

63% generate and use SBOMs
in their development and
security processes

53% use SBOM generation and
validation as their primary
strategy to minimize
attack surfaces

Secure CI/CD
Build and Deploy
Processes

Lollipop
Build
LAGOON

Lifecycle
phase

Marshmallow
Monitor MOUNTAIN

Monitor and Secure
Running Applications

SBOM
data
author

STICKY VULNERABILITIES

63% name inconsistent or
duplicate security controls
as their top challenge in
complex toolchains

26% feel fully prepared to
meet evolving regulatory
compliance standards

Dependencies

Vulnerability
data

Known unknown
documentation

Frosted
FORTRESS

Maintain Governance and
Standards Compliance

Timestamp

Secure
versioning &
storage

CONGRATS!
Your Software
Is Secure!

SWEET DEPENDENCIES

63% note ZTA as the most
critical strategy to secure
hybrid or multi-cloud
environments

61% say using AI/ML for
threat detection led to
better threat prioritization

Software Supply Chain Security Regulations From a DevSecOps Perspective

By Apostolos Giannakidis, Principal Product Security Engineer at Microsoft

High-profile software supply chain attacks like [SolarWinds](#), Log4j, and MOVEit highlight the urgent need to address the vulnerabilities caused by insecure software supply chains, prompting a decisive regulatory response. In the United States, entities like the Federal Trade Commission (FTC), the [Computer Fraud and Abuse Act](#), and emerging state laws are already being used to penalize companies that contain or distribute vulnerable or malicious code. Meanwhile, Europe's [Cyber Resilience Act](#) and new [Product Liability Directive](#) impose stricter cybersecurity obligations, heavy fines, and even personal accountability for software-related harm.

This isn't just paperwork. Failing to securely manage dependencies and maintain accurate dependency records now mean significant liability and legal accountability for CISOs. In today's regulatory environment, compliance is central to DevSecOps. However, if these processes are not properly automated and implemented, regulations could stifle the very agility DevSecOps was born to deliver.

This article aims to provide an overview of key regulations related to software supply chain security, discuss their implications in organizations, and explore their impact on DevSecOps teams.

Regulatory Landscape Overview

You can't patch what you don't know. To address this critical visibility gap, global [software supply chain security regulations](#) are emerging, establishing accountability within organizations. These regulations mandate that software vendors provide a detailed inventory of all direct and transitive dependencies, including open-source libraries and third-party components, embedded within their software products. The primary objective behind this granular transparency is to ensure that only attested, secure packages are utilized in the software stack.

This regulatory push reflects a global consensus: A thorough understanding of software composition is fundamental to effectively mitigate the critical security risks that stem from insecure, unpatched dependencies.

For information system auditors and security assessors, precise package names or identifiers, versioning information, and supplier details are critical artifacts. Integrating this granular dependency data into audit processes allows for an in-depth assessment of the organization's software attack surface. This facilitates the identification of risks such as:

- Components with unpatched critical vulnerabilities
- End-of-life software
- Unauthorized or unlicensed packages
- Deviations from internal security policies
- Non-compliance with external regulatory compliance frameworks

Key Existing Regulations

[Software bills of materials](#) (SBOMs) have emerged as a critical foundation for transparency into software composition. However, SBOMs don't solve the problem by themselves — they're a tool, not a silver bullet. An SBOM is merely a detailed list; without automated tooling to correlate its contents with real-time vulnerability intelligence and contextual risk, it could become an artifact with limited immediate security utility, potentially devolving into a compliance checkbox. But for regulators, it is an essential data source for effective security practices and vendor accountability for product integrity and security.

The [evolution of SBOMs](#) has been significantly shaped by an evolving regulatory landscape, transitioning them from a niche best practice to an increasingly mandated compliance requirement. This systemic shift requires integrated, cross-functional collaboration between development, security, and legal stakeholders.

U.S. Regulatory Initiatives

Issued in the United States after the SolarWinds incident, [Executive Order 14028](#) mandates secure software development and supply chain risk management, directing the NIST to develop SBOM guidelines for federal procurement. (Note: The [June 6, 2025 U.S. EO](#) rolled back SBOM requirements for vendors, but EO 14028 can still be used as a foundation). [NIST SP 800-218](#) (SSDF) provides actionable guidance, recommending SBOMs to mitigate supply chain risks and complementing EO 14028. [NIST SP 800-53](#) details federal security controls, including controls for supply chains. [CISA also recommends](#) the use of SBOMs.

The National Telecommunications and Information Administration (NTIA) SBOM has released a document that serves as a [benchmark for SBOM practices](#) and defines the minimum SBOM elements:

- Supplier
- Component name
- Component version
- Unique identifiers
- Dependency relationship
- Author of SBOM data
- Timestamp

While these U.S. efforts are largely complementary, federal agencies can legally request SBOMs for software they designate as "critical." The definition of "critical software," however, is not strictly specified and often varies by agency. This ambiguity frequently requires the involvement of legal teams in discussions about software dependencies and in handling regulatory SBOM requests.

EU Regulatory Initiatives

The European Union is also raising the bar. The Cyber Resilience Act, broader than any single U.S. mandate, treats software like a consumer good. It imposes on manufacturers obligations to meet minimum security standards, apply patches, perform vulnerability disclosures, and provide technical documentation, including an SBOM, for 10 years after market entry.

The [Digital Operational Resilience Act](#) (DORA) targets the financial sector's ICT and third-party supply chain risk. The [NIS2 Directive](#) broadens cybersecurity requirements for essential and important entities, encompassing supply chain security and requiring member states to adopt relevant policies. These EU regulations are complementary, addressing product security (CRA), financial resilience (DORA), and critical entity network security (NIS2).

Although all these regulations differ in direct legal force and target audience, they universally recognize the critical need for software composition transparency, secure development practices, proactive vulnerability management, improved incident response, and increased transparency and trust.

Impact on DevSecOps Practices

Despite the regulatory mandates requiring changes, the "how to" often remains unclear, creating challenges for development, security, operations, and legal teams trying to adapt. Actually, it is concerning how few organizations feel adequately prepared or confident in their strategies for securing their software supply chains.

For example, a recent [RSA Conference survey](#) showed a mere 20% of companies could meet the CISA's software development attestation deadline. Is the primary difficulty simply the lack of precise technical steps, or does it point to deeper issues in how these teams approach shared security responsibility? I would argue this isn't simply a technical challenge, but predominantly a cultural one.

If developers don't genuinely buy into the culture, they're likely to see compliance mandates as roadblocks, not as essential security improvements. That's why building a *compliance-first culture* is critical. This means actively rewarding secure-by-default approaches, making compliance a natural part of engineering workflows, and elevating compliance artifacts, such as SBOMs and provenance logs, to be core components of your development ecosystem.

Organizations now require a coordinated effort where development teams generate SBOMs, security teams validate artifact signatures, and operations teams ensure only compliant images are allowed to be deployed.

Achieving regulatory compliance at scale requires a fundamental shift, moving away from legacy manual workflows toward an automated ecosystem. In this model, enforceable, Policy-as-Code practices are integrated into continuous integration and continuous delivery (CI/CD) pipelines to meet demands without creating friction in development teams.

Additionally, for DevSecOps teams operating in regulated industries, traceability is no longer optional — it's a fundamental compliance mandate directly impacting pipeline design. To meet this, integrating supply chain security tools like [Tekton Chains](#) and [Sigstore](#) directly into CI/CD workflows becomes critical for automatically generating cryptographic proof of every action. Furthermore, DevSecOps practices must evolve to incorporate systems that ensure non-repudiation via mechanisms like signed attestations to provide thorough auditability and complete security traceability that regulators now demand.

When incorporating open-source code, development teams need to be acutely aware of the potential regulatory implications, especially if these packages have known vulnerabilities (CVEs). The best practice, of course, is to avoid packages with known CVEs when possible, but if not, a well-documented justification is critical for regulatory due diligence. This is why, as you define your supply chain security practices, it's essential to ensure procurement and legal teams have a seat at the table.

The following are some of the key ways effective DevSecOps teams are implementing software supply chain security regulatory mandates.

Automation of SBOM Generation in CI/CD

Think of SBOMs as living documents, not one-time snapshots. Regulators and standards require the automatic creation of an SBOM every time you build your software. Tools like [Syft](#), [Trivy](#), or [CycloneDX](#) plugins for Maven/Gradle can achieve this directly in the CI/CD pipeline before or during release. This means that everyone who needs to know what packages are used in your software gets an up-to-date SBOM, not an old one.

There might be rare cases where these tools will not auto-detect some packages. In such cases, you'll still need to manually add any missed packages in the SBOM for completeness.

Another best practice is to adopt a single SBOM format across all projects in the organization and enforce this via pipeline policies.

SBOM Validation and Deployment Gates

Just having an SBOM isn't enough. In a secure environment, we also need to know:

- Can we trust it?
- Has it been tampered with?
- Is it authentic?

Thus, regulations also require DevSecOps teams to perform SBOM validation and gated deployments. To address this, digital signatures, hashing, and attestations are critical for verifying and validating SBOMs. And of course, modern DevSecOps pipelines embed these validation and compliance checks as code, with cryptographic attestations and policy enforcement gates built into every stage of the software development lifecycle (SDLC).

A best practice is to integrate SBOM validation, including [provenance verification](#), into CI/CD deployment gates using Policy as Code such as [Open Policy Agent](#) (OPA) and [Kyverno](#). Kubernetes admission controllers can enforce this by verifying signatures and compliance data. Using dependency scanning and license compliance checks in deployment gates ensure only secure, compliant, and signed artifacts with vetted dependencies are released.

Storing and Versioning SBOMs Securely

SBOMs must be immutable, tamper evident, and readily accessible for audits or incident response. This creates a challenge. Some organizations store SBOMs alongside container images in artifact registries, while others use dedicated metadata servers.

Solutions are now emerging to address versioning and metadata creation in software supply chains, such as [Sigstore's Rekor](#). By leveraging an immutable and tamper-resistant ledger, software vendors and build systems can record signed metadata about software projects. This practice enables downstream parties to confidently make decisions regarding an object's trust and lifecycle, backed by non-repudiable evidence.

Best Practices for Ensuring Security and Compliance

The following outlines best practices to effectively ensure security and compliance throughout the software supply chain.

Document Known Unknowns

Known unknowns might be one of the most challenging problems in software supply chain security. As described by the [NTIA](#) and [NIST](#), there can be instances in which the full [dependency graph](#) is not enumerated in the SBOM. In these situations, the SBOM author must clearly flag such "known unknowns."

An SBOM must clearly indicate whether the dependency information for a particular component is complete, partial, or unavailable, and explain the reason for any gaps. This prevents consumers of the SBOM from making dangerously optimistic assumptions about the completeness of the data, ensuring a more realistic assessment of potential risks stemming from unlisted or unanalyzed dependencies.

One solution I have found valuable in addressing the known unknown challenge is using [runtime monitoring tools](#) such as IAST. Runtime monitoring enables us to identify the known unknown dependencies and verify whether they are actually loaded or executed. This way, we can follow a risk-based vulnerability management strategy as required by the EU CRA and the NIS2 Directive.

Adopt Regulations While Maintaining Dev Velocity

It's not uncommon for teams to be concerned that regulatory demands on DevSecOps will slow down their development velocity. However, this perception often stems from outdated approaches. Regulations don't inherently have to be blockers; when security and compliance practices are automated and embedded early into the lifecycle — think "shift left" — they transform into enablers, not roadblocks.

If your developers are bogged down filling out manual compliance checklists, it's a clear sign your implementation needs rethinking. Automating supply chain integrity isn't just good security hygiene; it's the only strategy to maintain your development velocity while achieving compliance.

Eliminate SBOM Sprawl

The proliferation of SBOMs — potentially one for every image, build, or microservice — can lead to an unmanageable volume without proper indexing and deduplication strategies, a challenge commonly termed "SBOM sprawl." I would argue that we should handle SBOMs using the same level of lifecycle management as production data, and apply data governance, retention, and archival or clean-up policies. Centralized aggregation and analysis platforms, such as [Google's Graph for Understanding Artifact Composition](#) (GUAC), are emerging to address this. These systems can provide a unified view of software components and their interdependencies, enhance our supply chain vulnerability visibility, help vulnerability management efforts, and improve compliance tracking.

Eliminate Slopsquatting

AI code suggestion tools, like ChatGPT or Cursor, can sometimes "hallucinate" and recommend code or dependencies that seem plausible but are actually incorrect, insecure, or outright malicious. This is often called "slopsquatting" and effectively sabotages the software supply chain. I've personally seen LLMs propose dependencies that are cleverly [typosquatted](#) or malicious packages, but they look like legitimate ones.

To address the risk introduced by AI code suggestion tools, a critical first step is configuring these AI assistants to prioritize our own vetted, internal libraries over public repositories. But that's not enough. To catch these insecure or non-compliant dependencies that slip through due to AI hallucinations, we also need rigorous dependency validation using automated allow-lists, backed by robust software composition analysis (SCA) tools.

Continuous Monitoring and Auditing of Software Supply Chains

True supply chain diligence requires continuous monitoring. The following monitoring practices are mandated by regulations:

- Constantly scanning for new CVEs affecting deployed components
- Proactively implementing immediate alerts for drift, violations, or new critical vulnerabilities
- Regularly reviewing and updating your supply chain risk management processes to keep pace with organizational and regulatory changes

Adopt SLSA to Adapt to Evolving Regulations

[Supply-chain levels for software artifacts](#) (SLSA) offer a practical framework to meet increasing software supply chain security regulations. By providing incremental, verifiable levels of assurance against tampering and unauthorized modification, SLSA helps organizations demonstrate the integrity and provenance of their software artifacts. Adopting SLSA levels can provide auditable evidence of secure build practices and artifact handling, directly addressing many regulatory demands for greater supply chain visibility, security, and due diligence, thereby easing compliance burdens.

Conclusion

It's clear now that effective "security by design" fundamentally begins with a secure supply chain. Regulations are directly influencing software design and processes, and these mandates are also becoming more complex. However, if we treat compliance like any other technical problem — and build it into our DevSecOps tools and automation, while investing in standards and team knowledge — we can achieve compliance efficiently.

Keep in mind that although the software supply chain is global, regulations themselves are not, leading to different jurisdictions having varying requirements for SBOMs. Consequently, DevSecOps teams and security auditors must stay up to date on the latest developments in their specific regions to ensure awareness of the most current and applicable regulatory requirements.

[In the future](#), SBOMs will play a big part in secure software development and supply chain management. Leveraging them through robust DevSecOps automation is becoming essential for any organization's proactive regulatory compliance strategy and overall security. 

References and additional resources:

- [Software Supply Chain Security](#) by Justin Albano, DZone Refcard
- [SBOM Essentials](#) by Siri Varma Vegiraju, DZone Refcard
- ["How To Implement Supply Chain Security in Your Organization"](#) by Kellyn Gorman
- ["NIST SP 800-161 Rev. 1, Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations"](#) by Jon Boyens et al
- [Leader's Guide to Software Supply Chain Security](#), Gartner
- ["Gartner Predicts Half of Supply Chain Management Solutions Will Include Agentic AI Capabilities by 2030,"](#) Gartner
- ["LLMs can't stop making up software dependencies and sabotaging everything"](#) by Thomas Claburn
- [SLSA](#), Safeguarding artifact integrity across any software supply chain
- ["Report: Only 1 in 5 organizations have full visibility into their software supply chain"](#) by Jenna Barron



Apostolos Giannakidis

 [@agiannakidis](#)
 [@giannakidisapostolos](#)
 [@cyberApostle](#)

Apostolos is currently leading product security initiatives for Microsoft Identity, focusing on proactive security mitigation. Before joining Microsoft, he served as VP of Application Security at JP Morgan Chase and led the security strategy at Waratek. He has been acknowledged by Oracle, he is featured on Google's Vulnerability Hall of Fame, and he holds two MSc degrees in Computer Science and Cloud Computing. When he's not working, you might find him automating his coffee machine to brew the perfect cup of Greek coffee.



Wattwatchers

Securing IoT Devices and Firmware
in a Distributed Energy Landscape

Wattwatchers

Energy
50 employees
Sydney, Australia

Solutions Used

DigiCert® Device Trust
Manager and Software
Trust Manager (part of the
DigiCert® ONE platform)

Primary Outcome

Enhanced the security of
their distributed energy
IoT devices and firmware
with DigiCert's automated
certificate management
and code signing solutions,
supporting seamless
compliance and long-
term scalability

*"DigiCert's digital trust
platform enabled our small
team to deliver an efficient,
resilient, and secure digital
infrastructure without being
slowed down by complex
configuration on the back end."*

—Grace Young,
Chief Innovation Officer, Wattwatchers

Read the full case study [here](#).

Challenges

Wattwatchers evolved from their consumer-centric startup roots to focus on large enterprises like public utilities and energy retailers. This strategic shift toward a more sustainable business model promised accelerated growth in both their smart device technology and "energy data-as-a-service" offerings.

However, this pivot created new challenges, most obviously increased scrutiny of the company's security practices. Enterprise prospects had rigorous demands around data protection, device identity, and certificate management, driven by stricter standards and regulations within the corporate and energy sectors.

As Wattwatchers prepared to launch their next-gen 6MW IoT devices, Chief Innovation Officer Grace Young recognized a critical challenge. The company would require a trustworthy, robust, and scalable solution to avoid significantly increasing resources for security troubleshooting while meeting new customers' rigorous expectations. Their API-driven architecture also demanded a security solution that could seamlessly integrate with existing systems and automate key tasks like firmware signing and device identity validation using PKI.

"Our APIs and the ingress of data through the system into our database and then out to our customers is our primary 'product' — it's not just the hardware, but access to the data our devices generate," Young explained. "Because we have a strong ethos of running lean, we needed a solution with very strong API integration. Our production processes needed to be able to link into it, and we needed it to be baked into our continuous integration environment."

Solution

Wattwatchers automated certificate provisioning and firmware signing at scale using DigiCert® Device Trust Manager and DigiCert® Software Trust Manager, part of the DigiCert® ONE platform. They needed a robust and scalable way to ensure that firmware updates were authentic, untampered, and delivered securely to field devices. DigiCert's solution enabled secure code signing with centralized policy control, ensuring firmware integrity across all deployment stages, from development to distributing updates.

DigiCert ONE's API-first architecture allowed Wattwatchers to integrate certificate issuance and code signing workflows into their existing systems, enabling them to streamline implementation, automate critical processes, and

reduce the risk of human error. It also enabled secure provisioning and renewal of device certificates throughout the product lifecycle. Because DigiCert ONE is modular and flexible, Wattwatchers could adopt the solution without a major overhaul of their technology stack, allowing for fast time to value and minimal disruption to operations.

Results

DigiCert ONE's OpenAPI spec allowed Wattwatchers to quickly map out how it would fit within the company's own processes. As a result, DigiCert ONE was operational within a week, saving the company about AUD\$40,000. Wattwatchers now benefits from automated certificate lifecycle management and streamlined secure code signing, reducing manual overhead while increasing security and efficiency. The solution supports regulatory compliance, protects against firmware tampering, and enables the company to scale their device fleet confidently.

Trust Every Build and Every Release. Every Time.

Secure every stage of your software pipeline with automated, policy-driven signing—from development to delivery

You move fast. Your software security should, too.

In today's Software Development Lifecycle, speed without trust is a risk you can't afford. You need a solution that transforms costly chaos into a secure, streamlined flow that matches the pace of your software development needs.

DigiCert® Software Trust Manager makes it easy to integrate code signing into your existing CI/CD workflows.

- Enforce signing policies by project, team, or environment
- Centralize control over both public and private keys (no more developers storing certs locally)
- Sign code automatically in Jenkins, GitHub Actions, Azure DevOps & more
- Maintain full visibility with detailed logs and audit trails
- Stop unauthorized code before it ships

Whether you're shipping containers or desktop apps, DigiCert gives you the confidence your software is tamper-proof and verifiable every step of the way

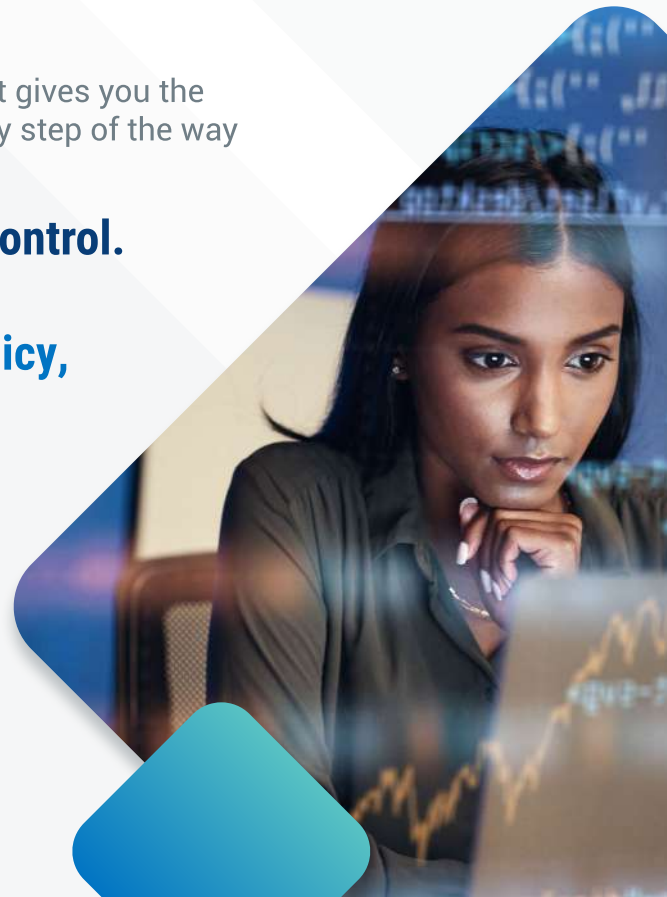
Eliminate bottlenecks. Enforce policy. Take Control.

See how DigiCert puts you in command of policy, governance, and trust.

[Explore DigiCert Software Trust Manager](#)

digicert®

The most-trusted global provider of high-assurance TLS/SSL, PKI, IoT and signing solutions





Maximizing Return on Investment When Securing Our Supply Chains

Where to Focus Our Limited Time to Maximize Reward

By Justin Albano, Software Engineer at IBM

The goal of DevOps and DevSecOps — and whatever future contractions come next — has been to break down walls, but in practice, it usually means that developers take on a greater burden. Now, developers are not just responsible for delivering a satisfactory product on time, but also managing the operations and security of the product. This begs the question: Is it possible for developers to accomplish all of this? The answer is yes, but only if we spend our time wisely.

In this article, we will delve into practical steps to build and grow an automated security process that alleviates many of the burdens of securing our software supply chain that can bog us down in the midst of tight schedules and an ever-growing backlog.

Implications of Supply Chain Security

The advent of supply chain security (SCS) and DevSecOps means that the scope of our development work has grown significantly. We are now responsible for:

- Knowing every dependency that makes it into our product, including transitive ones many levels down
- Reading and understanding Common Vulnerabilities and Exposures (CVE) reports
- Understanding the legal implications of different licenses
- Advising legal and security teams and resolving issues presented by these teams
- Fixing any problems when any of these processes do not go smoothly

In many teams, there is already more work than people to complete it, so how do we accomplish our existing tasks while taking on this additional load? The answer is understanding what is actually needed and automating it.

Automate

It is impossible to manually check our supply chain for each commit for any reasonably sized product. Each seemingly benign library can contain a critical future vulnerability or a nested license that is impermissible. We must always be diligent and minimize our dependencies, but we reach a point where it becomes intractable to manually track all our dependencies. The solution lies in automating our security process, and the first step in doing so is to understand what our security process needs to accomplish.

Understand the Security Process

As with continuous integration/continuous delivery (CI/CD), there is no one-size-fits-all approach. While patterns may arise, the specifics of our security process will vary. The first step is to understand what our security process, and our CI/CD process in general, should accomplish:

- What artifacts does our process need to deliver?
- What reports should accompany our artifacts?
- What security measures must be included (such as artifact signing and build process verification)?
- What is our desired security threshold?
 - What are the timelines for mitigating different vulnerabilities?
 - Which vulnerabilities are show-stoppers?
 - Which vulnerabilities can be mitigated in the next release?
- What licenses are permissible?
- What level of traceability or attestation is required?

The answers to these questions will vary by product — a military product will have different requirements than an internal product residing behind a DMZ. The important takeaway is to understand what is required of our product, create a meaningful starting point, and build on it until all of our requirements are satisfied.

Start With the Basics

Automating this process can seem overwhelming at first, but it's not if we focus on starting small and growing. Improving the process 20% today is far better than improving the process 100% some ethereal time in the future. The first place to start is by generating a software bill of materials (SBOM). This single document describes all of the dependencies in our supply chain and provides visibility into what our product relies on.

We must ensure that the SBOM we generate is:

- **Comprehensive.** The entire dependency graph must be accounted for, including transitive dependencies, build dependencies used to create artifacts (such as compilers), and dependencies in container images.
- **Readable.** The SBOM should be generated in a format that is parsable by tools and readable by developers, such as JSON or YAML. Human readability is essential for easier troubleshooting and understanding which dependencies are problems (or a suitable tool can be used to visualize the dependency graph).

Having a comprehensive and readable SBOM seems basic, but it is essential to securing our supply chain. We cannot audit or secure our supply chain if we do not know what goes into our product.

Note: An SBOM may provide a malicious actor with information about the attack surface of our product. We should consider carefully whether to deliver the SBOM with our product based on our security requirements.

Shifting Security Left

Generating an SBOM is a good first step, but it's not the end of the line. Having a security process in place should prevent vulnerabilities or compromising artifacts in our supply chain (such as impermissible licenses) from making it into our product. The next step is to analyze our SBOM and automatically fail our build if a dependency does not meet our standards. These standards include:

- No vulnerabilities that cross our security threshold
- No incompatible licenses
- No deprecated or unmaintained dependencies

When any of these standards are not met, we should break our build. The only way to guarantee that our supply chain is secure is to ensure that a substandard dependency is never introduced in the first place. In practice, though, we may be starting with a product that already has substandard dependencies, and we are trying to weed them out. We can't eat the entire elephant in a single bite, so we have to take it one step at a time.

A few pragmatic approaches are to:

- **Pick your battles wisely.** We should be honest about what's a showstopper and what's not. Having a deprecated dependency with zero known vulnerabilities may not be a showstopper, but having an incompatible license likely is. Start with breaking the build for standards that matter most and then gradually add standards as break-worthy.
- **Raise the security threshold over time.** At first, only break the build for critical or high vulnerabilities. Once these vulnerabilities are mitigated, keep increasing the threshold until our desired security threshold is reached.

These approaches require discipline: The standards should be raised over time and not kept relaxed indefinitely. Throughout this entire process, the goal should always be to start small and improve. Gradual improvement today is better than a perfect solution that never comes.

Other Considerations

With a solid automated SCS process in place, we can also add the following:

- Static code analysis
- Dynamic analysis of our executing product
- Third-party tools that aid in mitigation, such as tools that suggest mitigation for known vulnerabilities — or even automatically create pull requests (PRs) for mitigating vulnerabilities

Some of these tools may not even use our SBOM, so they can be pushed further left in the delivery process. For example, some tools scan the code in a repository or the code for each commit to ensure that it does not introduce any vulnerabilities. The goal is to fail the build as early as possible, so if tools allow us to check our code or commits — even before we build it — always lean on shifting as far left as possible.

Audit and Improve

Instituting SCS is not something we do once. It's a process that must be continually audited and improved. Over time, requirements will change, security constraints may tighten, and issues will arise. Our automated process should always be in alignment with our current needs.

A few ways we can continually improve our security process are to:

- **Watch for manual tasks.** Look for manual tasks that keep cropping up. For example, has the legal team been catching impermissible licenses in our product? Improve the license check in the automated process until no impermissible licenses are found again.
- **Get out ahead of other teams.** Look for opportunities to lessen the load on other teams. For example, has the security team been delaying releases because the security compliance checks are too burdensome? Improve the process by asking them what would help the process along and automate it (e.g., generating the vulnerability report in a different format or catching more vulnerabilities before sending the product to the security team or auditing).

There are always opportunities for improvement. The driving force for the long-term life of our automated SCS should be automating processes that free us up to focus on developing a quality product.

Conclusion

The introduction of DevOps and DevSecOps has increased the burden on developers. It can be easy to lose track of our main goal: delivering our product on time and under budget. While taking on new roles, such as operations and security, can seem like a Sisyphean endeavor, it doesn't have to be. We have to focus on what provides the greatest bang for the buck, including:

- Automating as much as possible
- Catching problems as early as possible
- Mitigating them as easily as possible (preferably with automated PRs)

Putting these principles into practice ensures that we can cover the roles of operations and security in addition to our main goal: timely delivery of a product that meets the needs of our customers. 🎲



Justin Albano

🎲 @albanoj2



Justin Albano is a software engineer at IBM responsible for building software-storage and backup/recovery solutions for some of the largest worldwide companies, focusing on Spring-based REST API and MongoDB development. When not working or writing, he can be found practicing Brazilian Jiu-Jitsu, playing or watching hockey, drawing, or reading.

People.ai

People.ai Transforms Security and Compliance With Endor Labs

ENDOR LABS**People.ai**

Revenue
Intelligence Software
250 employees
San Francisco, United States

Solutions Used

Endor Labs AppSec Platform

Primary Outcomes

With Endor Labs, People.ai was able to reduce false positives by 95.7%, shift security left, and achieve faster MTTR, all while enhancing compliance and customer trust.



“Endor Labs is a rising star in AppSec. They're helping us fix what's necessary by providing context, and context is king when you're dealing with open source software.”

—James Wedewer,
Director of Information Security,
People.ai

Read the full case study [here](#).

People.ai is the leading AI data platform transforming how go-to-market teams improve sales effectiveness and win rates via industry-leading comprehensive data foundation and generative AI capabilities.

Challenges

Application security is integral to the success of People.ai, and it was only becoming more important as the organization sought to unlock new business opportunities with a Microsoft 365 certification and follow the new ISO 42001 standard. But they found that their existing application security platform (Snyk) was not helping them achieve outcomes related to developer experience and compliance because of:

- Excessive false positives and manual triage
- Impact on developer experience and velocity
- Limited codebase and language coverage
- Evolving generative AI risks

Solution

People.ai's security team ran a competitive evaluation to find an AppSec platform that could deliver on four key requirements:

- High accuracy and trust
- Full coverage and CI integration
- Modern, flexible architecture
- Strong partnership and customer success

Results

People.ai chose Endor Labs' entire suite of application security tools because of its:

- **Reachability-based prioritization:** Endor Labs showed which vulnerabilities were actually exploitable in People.ai's code, reducing false positives dramatically.
- **Developer-centric approach:** Endor Labs is designed for modern workflows and has intuitive APIs, easy CI integration, and actionable reporting.
- **Modern architecture and AI security leadership:** From embedding scanning within AI code assistants to using agents to parse PRs, Endor Labs showed what it means to build an AI AppSec platform.
- **Strong partnership and responsiveness:** People.ai was impressed by the ease of implementing the platform, the outstanding customer success team, and the organization's readiness to act upon feedback.

AppSec that understands your code (and everything it depends on).



See More →

- ✓ Reduce noise by 92%
- ✓ Secure AI-generated code
- ✓ Fix issues 6x faster
- ✓ Streamline compliance



ENDOR LABS



DZZone®

Practical Steps to Secure the Software Supply Chain End to End

By Karteek Kotamsetty, Lead Customer Engineer at Google Public Sector LLC

The software supply chain has rapidly evolved into a critical vulnerability point and primary target for malicious actors. As we progress in 2025, organizations face an increasingly complex and dynamic threat landscape. This article offers a practical checklist for securing the software supply chain and clarifies the pivotal role of zero trust.

Software Supply Chain Security Checklist

The following sections offer insights and guidance for organizations seeking to implement or enhance the security of their software supply chains.

Preparation and Governance

Establishing a secure software supply chain fundamentally requires preparation and solid governance. It's important to drive a culture within the organization that enables security best practices, to review the security metrics in a defined schedule, and to track the progress made to improve security posture.

- ❑ Ensure employees complete security awareness trainings quarterly
- ❑ Prioritize security alongside development speed
- ❑ Encourage secure coding practices
- ❑ Update teams on the latest threats so that they can adapt and deliver solutions that align with organizational values
- ❑ Establish clear security objectives
- ❑ Review metrics like security training completion rate, number of open vulnerabilities and length of time open, and impact of potential breaches on organizational brand value
- ❑ Ensure automated CI/CD workflows have rigid controls that block deployment until code scanning is completed
- ❑ Ensure metrics are reviewed regularly and feedback is communicated to applicable teams

Secure Development Practices

In the SDLC, the development phase is where vulnerabilities might creep in. Therefore, it's important to implement secure coding best practices right from the start. It's critical for organizations to vet any third-party components and protect their source code repositories from any vulnerabilities.

- ❑ Review open-source and third-party components with SCA tools
- ❑ Provide developers clear guidance on approved/unapproved components
- ❑ Enforce use of approved components and limit use of unapproved components
- ❑ For new components needed, review the support level provided by the open source community and how frequently they respond to security incidents
- ❑ Deploy automated secrets detection tools that can prevent any accidental exposure of secrets like API keys or credentials
- ❑ Standardize code review processes and ensure necessary approvals are obtained before code is pushed to source code repositories
- ❑ Enforce the principle of least privilege (PoLP) and MFA to protect access to source code repositories
- ❑ Use cryptographic hashes to prevent tampering

Secure Build and Release Processes

To secure the software's journey from development to deployment, the build and release phases are critical to maintaining an application's integrity since attackers can introduce malicious code or tamper with its components.

Hardening the build environment by using advanced scanning techniques and automated security configurations can mitigate the risk of malicious alterations in CI/CD pipelines.

- ❑ Protect build systems, CI/CD pipelines, and artifact repositories
- ❑ Set up private artifact repositories to centralize the management, storage, and distribution of all build artifacts, creating a single point for tracking and securing every component
- ❑ Enforce code signing for all software packages
- ❑ Implement reproducible builds and verify that identical source code always yields to identical binary output to detect unwanted changes
- ❑ Use ephemeral and isolated build environments that are destroyed after each use to limit the impact of a compromised system
- ❑ Embed security into builds for development, testing, and prod environments to seamlessly integrate security with CI/CD pipelines
- ❑ Store all CI/CD pipeline definitions in a version control system
- ❑ Map secrets to the correct environments
- ❑ Use the PoLP limit
- ❑ Use specialized tools to scan CI/CD workflows for vulnerabilities and identify common pitfalls like exposed secrets in pipeline definitions

Secure Deployment and Runtime

The deployment and runtime phases are critical steps after the build phase is completed. Securing deployments and continuously monitoring during runtime ensures the integrity of the deployment and protects it from any threats.

- ❑ Scan final build binaries before deployments to uncover misconfigurations, detect malware, or identify missed vulnerabilities
- ❑ Test images that are delivered from external sources in a dedicated isolated image scanner to uncover malicious components or backdoors in applications
- ❑ Maintain artifact lineage
- ❑ Deploy images from trusted sources via an organizational policy
- ❑ Verify the publisher's identity and ensure artifact integrity through cryptographic signatures
- ❑ Apply the PoLP and enforce strong authentication and authorization for user and service accounts and CI/CD pipelines
- ❑ Enact strict network segmentation to limit lateral movement within infrastructure
- ❑ Regularly audit deployment infrastructure security configurations to ensure systems are patched and hardened
- ❑ Use cloud security posture management tools to identify configuration drift
- ❑ Continuously monitor running software using cloud workload protection platforms to actively scan for new vulnerabilities
- ❑ Ensure monitoring systems send alerts to the security team when vulnerabilities are detected
- ❑ Prioritize risks with clear, actionable insights

CHECKLIST CONTINUES ON NEXT PAGE

Identity and Access Management

Robust identity and access management (IAM) is the key foundation to secure software supply chains. It's important to implement the PoLP and a strong authentication and authorization framework across all the phases in SDLC to protect the organizational resources from unauthorized access.

- ❑ Use adaptive authentication to enforce authentication dynamically when suspicious activity is detected
- ❑ Use role-based access controls (RBACs) to enforce authorization based on the PoLP
- ❑ Implement dedicated secrets management tools to store credentials and API keys securely
- ❑ Ensure credentials are not hard-coded in configuration files

- ❑ Implement strict access controls to limit users who can retrieve credentials
- ❑ Implement policies to automate key rotations at defined schedules
- ❑ Ensure every access attempt to secrets is logged for auditing and forensic analysis
- ❑ Implement policies to revoke access to secrets when suspicious activity is detected

Incident Response and Compliance

Security incidents are an inevitable reality in the software supply chain, and they serve as a true test of an organization's resilience and ability to respond rapidly. To limit potential damage and preserve user trust, a well-defined incident response plan (IRP) and the capability to respond swiftly are absolutely essential. Aligning with industry standards can help organizations stay compliant.

- ❑ Develop and test an IRP that includes the responsibilities of teams and individuals and clear communication guidelines
- ❑ Focus on early detection, containment, and eradication of threats and perform a safe recovery
- ❑ Regularly train personnel on the IRP

- ❑ Conduct mock drills to assess team preparedness and identify gaps in response
- ❑ Review and update compliance policies regularly
- ❑ Routinely conduct internal audits and independent third-party assessments to identify areas that need to be addressed

Conclusion

The modern software supply chain demands an integrated defence and in-depth strategy to protect against sophisticated threats. To achieve true organizational resilience, organizations need to have an end-to-end security posture that spans every single stage of the SDLC. This means cultivating a security-first culture across teams, where every individual contributes to and leverages collective threat intelligence to stay ahead of emerging risks. This holistics approach can help organizations navigate the complex security landscape. 🌐



Karteek Kotamsetty

🌐 @kkotamsetty
in @karteek-kotamsetty
🌐 kkotamsetty

At Google, Karteek works with various states on modernizing their data lakes and data warehouses on Google Cloud and helps customers drive business outcomes with AI. Karteek has extensive experience in supporting state and local agencies to modernize their legacy infrastructure into modern platforms that can be powered by generative AI. Prior to Google, Karteek worked at Oracle for 8+ years.



AI-Powered Security for the Modern Software Supply Chain

Reinforcing Software Integrity in an Era of Autonomous Code and Expanding Risk

By Akanksha Pathak, Senior Cybersecurity Consultant at Visa Inc

In today's software landscape, the supply chain has grown from a controlled pipeline to a vast, interconnected ecosystem. Modern development relies heavily on third-party dependencies, open-source components, distributed CI/CD pipelines, and ephemeral cloud-native environments. While this fosters rapid innovation, it also amplifies risk exposure. High-profile breaches like SolarWinds and Log4Shell revealed how a single weak link can cascade across thousands of organizations.

AI and automation now stand at a pivotal juncture — offering unprecedented defense capabilities and introducing new attack vectors. The challenge lies in responsibly integrating AI to reinforce supply chain integrity without compromising control, compliance, or clarity.

AI-Powered Security: Reinventing Supply Chain Defense

The use of AI in cybersecurity has shifted from reactive defense to proactive, real-time protection. AI's capacity to detect patterns, automate decisions, and augment human capabilities makes it an ideal fit for today's fast-moving software delivery ecosystems.

AI-Driven Threat Detection and Intelligent Protection

Modern threats often slip past signature-based tools. AI enhances threat detection by identifying anomalous behavior in real time. Behavior-based detection tools like [Falco](#) track deviations from expected patterns in containerized environments, helping detect zero-day and insider threats.

During the coding and commit phase, tools such as [TruffleHog](#) and [Gitleaks](#) analyze codebases for secrets exposure, catching leaks before they reach production. They are capable of scanning large codebases efficiently and identifying even deeply buried credentials.

Static application security testing (SAST) is made more effective with tools like [Semgrep](#) and [SonarQube Community Edition](#), which integrate AI-based rulesets to identify insecure code practices in context. These tools improve developer productivity by surfacing actionable issues without overwhelming them with noise.

AI also supports enhanced log correlation, surfacing connections across multi-source logs that traditional filters may miss. These capabilities improve detection of low-and-slow attacks and reduce alert fatigue, providing security teams with a clearer, more actionable incident picture.

Integration With Open-Source SIEM and SOAR

To maximize AI's effectiveness, it must be integrated into broader detection and response systems. Open-source SIEM platforms like [Wazuh](#) and the [ELK Stack](#) leverage machine learning to enrich event data, detect threats, and reduce false positives through adaptive learning.

These platforms allow users to define correlation rules and continuously refine anomaly detection models. Wazuh, in particular, supports compliance dashboards and threat intelligence integration, making it a solid foundation for regulated environments.

For orchestrated response, [TheHive](#) and [Cortex](#) enable AI-assisted triage, case management, and incident enrichment. Cortex can automate fetching of indicators of compromise (IOCs), run response playbooks, and tag incidents based on severity — all within seconds.

AI Agents in DevSecOps Pipelines

AI is becoming indispensable in DevSecOps pipelines. Beyond assisting developers, it is now embedded into workflows that automatically manage dependencies, monitor code quality, and anticipate security regressions with minimal human involvement.

Secure Development Through Embedded AI

Integrated development environments (IDEs) are being enhanced by AI copilots and linters that assist in writing secure code. Tools built on models like [CodeBERT](#) can highlight insecure logic and suggest better patterns as developers type. These smart linters go beyond syntax checks to flag unsanitized inputs, weak encryption, and insecure API calls, turning IDEs into the first line of defense. Some AI copilots can even explain the implications of insecure code and recommend fixes in natural language, enhancing learning.

Autonomous Dependency Management and SBOM Verification

Automating software bill of materials (SBOM) generation and vulnerability scanning is vital for securing sprawling dependency trees. [OSS Review Toolkit](#) (ORT), [Syft](#), and [Grype](#) provide integrated workflows to track, audit, and assess the risk of third-party components. ORT is particularly valuable for license compliance, ensuring that OSS components align with corporate policy. Syft and Grype operate together — Syft creates SBOMs, while Grype scans them for known vulnerabilities, supporting real-time security insights.

Tools like [Renovate](#) act as intelligent agents that automatically suggest or implement version upgrades based on known vulnerabilities, usage context, and semantic compatibility. Renovate's AI logic helps prioritize patches that are most critical and least disruptive.

Table 1. Review of open-source tools for SBOM and dependency management

Tool Name	Purpose
OSS Review Toolkit	End-to-end compliance and licensing for OSS dependencies
Syft	SBOM generator for container images and filesystems
Grype	Vulnerability scanner using SBOM or Syft output
Renovate	Dependency updater and patch management bot

Predictive Risk Modeling and Attack Surface Mapping

The ability to anticipate threats before they occur is a game-changer. Predictive risk modeling uses telemetry and machine learning to map likely paths of attack, highlight high-risk assets, and guide remediation prioritization. Tools like [Dependency-Track](#) and [CycloneDX](#) present dynamic visualizations of your SBOMs, enabling better risk prioritization. They offer dashboards that map dependency usage across teams, systems, and products, helping to prevent the reuse of vulnerable packages.

For container and cloud-native environments, [Clair](#) and [kube-bench](#) help quantify misconfiguration and base-image vulnerabilities. Clair supports multiple vulnerability sources, while kube-bench tests Kubernetes configurations against industry benchmarks like CIS. Combining these tools enables organizations to continuously assess their supply chain posture and shrink the exploitable surface area.

The Dark Side: New Threat Vectors Introduced by AI

While AI boosts security, it also introduces new risks. From unvetted tools to adversarial manipulation of models, defenders must understand how attackers might exploit AI systems.

Shadow AI

Shadow AI refers to unapproved AI tools used by developers or operations teams without organizational oversight. These include browser-based LLMs or unauthorized plugins that access sensitive codebases or credentials. Such tools can violate compliance, leak source code, or bypass review processes. Without centralized oversight, shadow AI increases the risk of data misuse, misconfigurations, and API exposure.

Adversarial AI and Model Poisoning

Attackers now use obfuscation techniques to craft adversarial code that evades AI detection models. In addition, public datasets used to train open-source models may be poisoned to introduce bias or backdoors.

AI models in security workflows are only as good as the data they're trained on. Poisoned datasets can cause false negatives or encourage insecure development practices. Attackers may even impersonate copilots or inject malicious snippets into training repositories. And compromised AI agents in DevSecOps pipelines can push unsafe patches, misclassify malicious code as benign, or recommend insecure practices — subverting the very systems meant to defend the pipeline.

Regulatory, Compliance, and Ethical AI Challenges

As AI use increases, so does the need for accountability. Regulatory frameworks and ethical guidelines are critical to ensuring responsible deployment across the software supply chain.

AI Governance and Risk Management

Regulations like the [EU AI Act](#) and the [NIST AI RMF](#) aim to promote trustworthy and auditable AI. Their principles guide risk assessment, model explainability, and lifecycle governance. Organizations must ensure their open-source models are legally licensed, traceable, and aligned with internal compliance mandates. Failing to track provenance can expose the business to legal and operational risks.

Balancing Automation with Human-in-the-Loop Oversight

AI is powerful, but it shouldn't operate in a vacuum. Critical decisions — especially around patching or remediation — should be validated by humans who understand the business context. Human-in-the-loop systems combine the speed of automation with the judgment of experienced security professionals. This hybrid approach improves accuracy and accountability, and ensures AI tools remain aligned with real-world needs.

Conclusion

AI is reshaping supply chain security — accelerating detection, automating analysis, and surfacing hidden vulnerabilities across complex systems. Yet its power comes with responsibility. By combining open-source tooling with strong governance, ethical oversight, and human validation, organizations can build software that is not only fast and scalable but also secure and trustworthy. In the age of autonomous code, security must be equally autonomous, transparent, and collaborative. 🌐

Further related reading:

- ["Security in the Age of AI: Challenges and Best Practices"](#) by Akanksha Pathak
- ["Guide to Securing Your Software Supply Chain: Exploring SBOM and DevSecOps Concepts for Enhanced Application Security"](#) by Akanksha Pathak
- ["Building Resilient Cybersecurity Into Supply Chain Operations: A Technical Approach"](#) by Akanksha Pathak
- [Software Supply Chain Security Core Practices](#) by Justin Albano, DZone Refcard
- [Secrets Management Core Practices](#) by Apostolos Giannakidis, DZone Refcard
- [Threat Detection Core Practices](#) by Sudip Sengupta, DZone Refcard



Akanksha Pathak

🌐 [@pathakakanksha991](#)

in [@Pathakakanksha991](#)

Akanksha Pathak is a senior cybersecurity consultant specializing in cloud security, application security, threat analysis and response, vulnerability management, and product security. As a senior member of the corporate governance team, she oversees the third-party cybersecurity practice. Her expertise lies in managing supplier relationships while also architecting and analyzing application designs.



Solutions Directory

This directory contains tools for securing software supply chains. It provides pricing data and product category information gathered from vendor websites and project pages. Solutions are selected for inclusion based on several impartial criteria, including solution maturity, technical innovativeness, relevance, and data availability.

DZONE'S 2025 SOFTWARE SUPPLY CHAIN SECURITY SOLUTIONS DIRECTORY

2025 PARTNERS

Product	Purpose	Availability	Website
DigiCert® Software Trust Manager	Automate secure code signing, key-pair management, and RBAC	Trial period	digicert.com/software-trust-manager
Endor Labs Application Security Platform	Identify, prioritize, and fix supply chain and OSS security risks	By request	endorlabs.com/product
Company	Purpose	Availability	Website
A10 Defend	Intelligent and automated DDoS protection	By request	a10networks.com/products/a10-defend
A10 Harmony Controller	Service analytics and management	Trial period	a10networks.com/products/harmony-controller
A10 Thunder Application Delivery Controller	Application delivery and load balancing	Trial period	a10networks.com/products/thunder-adc
A10 Thunder Convergent Firewall	Application and infrastructure security	By request	a10networks.com/products/thunder-cfw
A10 Thunder SSL Insight	SSL decryption and visibility	By request	a10networks.com/products/thunder-ssli
Acunetix by Invicti	Web app and API security testing	By request	acunetix.com
Aikido	Security platform for code and cloud	Free tier	aikido.dev
Airlock Gateway	Web application and API protection	By request	airlock.com/en/secure-access-hub/components/gateway
Airlock IAM	Customer identity and access management	By request	airlock.com/en/secure-access-hub/components/iam
Airlock Microgateway	Kubernetes-native protection of APIs and microservices	By request	airlock.com/en/secure-access-hub/components/microgateway
Airlock Secure Access Hub	Secure access management	By request	airlock.com/en/secure-access-hub/overview
Akamai App & API Protector	Security for websites, apps, and APIs	Trial period	akamai.com/products/app-and-api-protector
Akamai Enterprise Application Access	Zero-trust network access	Trial period	akamai.com/products/enterprise-application-access
Akamai Secure Internet Access Enterprise	Cloud-based DNS firewall	Trial period	akamai.com/products/secure-internet-access-enterprise
Amazon Cognito	Authentication service	Free tier	aws.amazon.com/cognito
AWS Secrets Manager	Cloud secrets management	Trial period	aws.amazon.com/secrets-manager
AWS Shield	Managed DDoS protection	By request	aws.amazon.com/shield
AWS WAF	Web app protection	By request	aws.amazon.com/waf
Anchore Enterprise	SBOM-powered, end-to-end software supply chain security	Trial period	anchore.com/platform

DZONE'S 2025 SOFTWARE SUPPLY CHAIN SECURITY SOLUTIONS DIRECTORY

Product	Purpose	Availability	Website
Anomali Platform	AI-powered security and IT operations	By request	anomali.com/platform
Apiiro Platform	Application security posture management	By request	apiiro.com
Aqua CloudSploit	Cloud security posture management	Open source	github.com/aquasecurity/cloudsploit
Aqua kube-bench	Kubernetes deployment security	Open source	github.com/aquasecurity/kube-bench
Aqua Platform	CNAPP	By request	aquasec.com/aqua-cloud-native-security-platform
Aqua Trivy	Vulnerability and misconfiguration scanning	Open source	aquasec.com/products/trivy
Arctic Wolf Aurora Platform	Open XDR platform	By request	arcticwolf.com/aurora-platform
Armo Kubescape	Kubernetes security platform	Open source	armosec.io/kubescape
Armor	Managed detection and response, compliant cloud solutions	By request	armor.com
Arnica	Application security	Free tier	arnica.io
Azul Vulnerability Detection	Vulnerability detection in Java apps	By request	azul.com/products/vulnerability-detection
Bandit	App security testing for Python code	Open source	bandit.readthedocs.io/en/latest
Barracuda Application Protection	Web app and API protection	Trial period	barracuda.com/products/application-protection
Barracuda Web Application Firewall	Website and app threat protection	Trial period	barracuda.com/products/application-protection/web-application-firewall
BeyondTrust Endpoint Privilege Management	Enforce least privilege, prevent attacks, and control applications	By request	beyondtrust.com/products/endpoint-privilege-management
BeyondTrust Secure Remote Access	Remote access management	By request	beyondtrust.com/secure-remote-access
Bitdefender Gravityzone Business Security Enterprise	Endpoint protection, detection, and response	Trial period	bitdefender.com/en-us/business/products/gravityzone-enterprise-security
Black Duck Polaris	Cloud-based app security testing	By request	synopsys.com/software-integrity/polaris.html
BMC AMI Security	Automatic mainframe threat detection and response	By request	bmc.com/it-solutions/bmc-ami-mainframe-security.html
Broadcom ACF2	Scalable, modern mainframe security	By request	broadcom.com/products/mainframe/security/acf2
Broadcom Symantec Enterprise Cloud	Data-centric hybrid security	By request	broadcom.com/products/cybersecurity
BeEF	Web browser penetration testing	Open source	beefproject.com
Censys Search	Discover, monitor, and analyze internet assets for security	By request	censys.com/platform
Chainguard Images	Software supply chain security with hardened container images	Free tier	chainguard.dev
Check Point CloudGuard CNAPP	CNAPP	By request	checkpoint.com/cloudguard/cnapp
Checkmarx KICS	Static code analysis of IaC	Open source	checkmarx.com/product/opensource/kics-open-source-infrastructure-as-code-project
Checkmarx One	CNAPP	By request	checkmarx.com/product/application-security-platform

DZONE'S 2025 SOFTWARE SUPPLY CHAIN SECURITY SOLUTIONS DIRECTORY

Product	Purpose	Availability	Website
CIRT.net Nikto	Web server scanner	Open source	cirt.net/Nikto2
Cisco Secure Endpoint	Cloud-native endpoint security	Trial period	cisco.com/site/us/en/products/security/endpoint-security/secure-endpoint/index.html
Cisco Umbrella	Cloud security	Trial period	umbrella.cisco.com
Cisco XDR	Extended detection and response	By request	cisco.com/site/us/en/products/security/xdr/index.html
Codenotary Trustcenter/Enterprise	Secure, real-time software supply chain management	By request	codenotary.com/products/trustcenter
CodeQL	Semantic code analysis engine	Open source	codeql.github.com
CodeSecure CodeSentry	Binary composition analysis	By request	codesecure.com/our-products/codesentry
CodeSecure CodeSonar	SAST platform	By request	codesecure.com/our-products/codesonar
Contrast Runtime Security Platform	RASP	Sandbox	contrastsecurity.com
Conviso Platform	App security posture management	Free tier	convisoappsec.com
CrowdStrike Falcon Platform	AI-native cybersecurity	Trial period	crowdstrike.com/platform
Cybeats SBOM Studio	SBOM lifecycle management with cybersecurity insights	By request	cybeats.com/product/sbom-studio
CyberArk Conjur	Secrets management	Open source	conjur.org
Cybereason Defense Platform	AI-driven security for threat detection and response	By request	cybereason.com/platform
Cycode	App security posture management	By request	cycode.com
Darktrace ActiveAI Security Platform	Cybersecurity platform	By request	darktrace.com/platform
Deep Instinct DSX	Deep learning-based cybersecurity	By request	deepinstinct.com
Deepfactor	Runtime software composition analysis	Trial period	deepfactor.io
Digital.ai Application Security	App monitoring and protection	By request	digital.ai/products/application-security
Edgescan	Unified security platform	By request	edgescan.com/the-platform
Enov8 Test Data Manager	Test data management and data compliance	By request	enov8.com/data-compliance-suite-devops-edition
Exabeam	Threat detection, investigation, and response	By request	exabeam.com
F5 BIG-IP DDoS Hybrid Defender	Network and app layer DDoS protection	By request	f5.com/products/big-ip-services/ddos-hybrid-defender
F5 Distributed Cloud API Security	API endpoint identification, mapping, and protection	By request	f5.com/products/distributed-cloud-services/api-security
F5 Distributed Cloud DDoS Mitigation Service	Network and app protection against L3-L7 attacks	By request	f5.com/products/distributed-cloud-services/l3-and-l7-ddos-attack-mitigation
F5 Distributed Cloud WAF	Distributed web application protection	By request	f5.com/products/distributed-cloud-services/distributed-cloud-waf
F5 NGINX App Protect	Secure, automate, and scale modern applications and APIs	Trial period	nginx.com/products/nginx-app-protect
Fastly Next-Gen WAF	App, API, and microservices protection	Free tier	fastly.com/products/web-application-api-protection

DZONE'S 2025 SOFTWARE SUPPLY CHAIN SECURITY SOLUTIONS DIRECTORY

Product	Purpose	Availability	Website
Forcepoint Data Security Cloud	Data security platform	By request	forcepoint.com/solutions/data-security-cloud
Fortinet Universal ZTNA	Zero-trust network access	By request	fortinet.com/solutions/enterprise-midsize-business/network-access/application-access
Fortra Alert Logic	Managed detection and response	By request	alertlogic.com
Fortra Core Security	Threat prevention and IAM	By request	fortra.com/product-lines/core-security
Fortra Digital Guardian	Data loss prevention and endpoint detection and response	By request	digitalguardian.com/platform
GitGuardian	Secrets security and NHI governance	Free tier	gitguardian.com
GitHub Advanced Security	Supply chain security for repositories	Free tier	github.com/security/advanced-security
GitLab	AI-powered DevSecOps	Free tier	about.gitlab.com
Google Cloud Build	Build, test, and deploy securely on a serverless CI/CD platform	Free tier	cloud.google.com/build
Google Cloud Workstations	Fully managed development environments	By request	cloud.google.com/workstations
GrammaTech	Binary analysis and software assurance	By request	grammatech.com
Graylog API Security	End-to-end API threat monitoring, detection, and response	Free tier	graylog.org/products/api-security
Graylog Enterprise	Centralized log management	By request	graylog.org/products/enterprise
Graylog Open	Open log management	Open source	graylog.org/products/source-available
Graylog Security	SIEM	By request	graylog.org/products/security
GuardRails	Automate security checks in DevSecOps pipelines	Free tier	guardrails.io
Harness ChaosNative	Kubernetes security and resilience	Open source	github.com/chaosnative
HCL AppScan 360°	Cloud-native application security	By request	hcl-software.com/appscan/products/appscan360
HCL AppScan on Cloud	Fully hosted and managed application security on cloud platform	Trial period	hcl-software.com/appscan/products/appscan-on-cloud
Hillstone Networks W-Series Web Application Firewall	Web server, application, and API security	By request	hillstonenet.com/products/application-protection/waf
IBM Cloud Secrets Manager	Secrets management	Trial period	ibm.com/products/secrets-manager
IBM NSI	Managed service for authoritative DNS and traffic steering	Free tier	ibm.com/products/nsi-connect
IBM QRadar SOAR	Incident response automation and process standardization	By request	ibm.com/products/qradar-soar
Imperva API Security	Continuous API endpoint security	Trial period	imperva.com/products/api-security
Imperva Application Security Platform	Integrated security, threat detection, and compliance	Trial period	imperva.com/products/application-security
Imperva DDosS Protection	Cyberattack mitigation	Trial period	imperva.com/products/ddos-protection-services
Imperva Web Application Firewall	Application security	Trial period	imperva.com/products/web-application-firewall-waf

DZONE'S 2025 SOFTWARE SUPPLY CHAIN SECURITY SOLUTIONS DIRECTORY

Product	Purpose	Availability	Website
in-toto	Software supply chain security framework	Open source	in-toto.io
Infoblox Ecosystem	Automate SecOps response	By request	infoblox.com/solutions/security-ecosystem
Infoblox SOC Insights	AI-driven analytics to correlate DNS threat and asset data	By request	infoblox.com/products/soc-insights
Infoblox Threat Defense	Hybrid DNS-layer security	By request	infoblox.com/products/threat-defense
Invicti	DAST-first application security platform	Trial period	invicti.com
JFrog Software Supply Chain Platform	DevOps, software supply chain security, and pipeline automation	Trial period	jfrog.com/platform
Jit	AppSec platform	Free tier	jit.io
JupiterOne	Cyber asset management and supply chain risk	By request	jupiterone.com
Kusari	Software supply chain security for DevSecOps	By request	kusari.dev
Legit Security	AI-native ASPM platform	By request	legitsecurity.com
LevelBlue MTDR	Security monitoring	By request	levelblue.com/products/managed-threat-detection-and-response
LevelBlue Network Security	Network security solutions	By request	levelblue.com/network-security
LevelBlue USM Anywhere	XDR security	By request	levelblue.com/xdr
Lookout Threat Intelligence	Mobile threat intelligence	By request	lookout.com/products/endpoint-security/threat-intelligence
Lookout Mobile Endpoint Security	Mobile endpoint detection and response	By request	lookout.com/products/endpoint-security
Mend AppSec Platform	Automated open-source security and license compliance management	By request	mend.io
Microsoft Defender for Cloud Apps	SaaS security	Trial period	microsoft.com/en-us/security/business/siem-and-xdr/microsoft-defender-cloud-apps
Microsoft Defender for Endpoint	AI-powered endpoint security across devices	Trial period	microsoft.com/en-us/security/business/endpoint-security/microsoft-defender-endpoint
Microsoft Azure Key Vault	Cryptographic key and other secrets security	Trial period	azure.microsoft.com/en-us/products/key-vault
Microsoft Azure Web Application Firewall	Cloud-native web application firewall	By request	azure.microsoft.com/en-us/products/web-application-firewall
Nmap	Network exploration and security auditing	Open source	nmap.org
NowSecure Platform	Continuous automated mobile application security testing	By request	nowsecure.com/products/nowsecure-platform
NowSecure Workstation	Mobile app security testing toolkit	By request	nowsecure.com/products/nowsecure-workstation
OffSec Services Kali Linux	Penetration testing distribution	Open source	kali.org
Oligo	Runtime application security	By request	oligo.security
Onapsis Platform	Secure ERP solutions for SAP and Oracle	By request	onapsis.com
Network Detection & Response	Threat detection and response	By request	opentext.com/products/network-detection-and-response

DZONE'S 2025 SOFTWARE SUPPLY CHAIN SECURITY SOLUTIONS DIRECTORY

Product	Purpose	Availability	Website
OpenText Core Application Security	Application security	By request	opentext.com/products/saas/core-application-security
OpenText Core Data Discovery & Risk Insights	Cloud-first data security platform	By request	opentext.com/products/voltage-fusion-platform
OpenText EnCase Endpoint Security	Endpoint detection and response	By request	opentext.com/products/encase-endpoint-security
OpenText Enterprise Security Manager	Enterprise SIEM security	By request	opentext.com/products/enterprise-security-manager
OpenText Identity Manager	Identity access and lifecycle management	By request	opentext.com/products/identity-manager
OpenText Static Application Security Testing	Static application testing and code analysis	By request	opentext.com/products/static-application-security-testing
OpenText Threat Intelligence	AI-/ML-powered threat intelligence	By request	opentext.com/products/threat-intelligence
OPSWAT MetaDefender	Advanced threat prevention	By request	opswat.com/products/metadefender
Orca Security	Agentless cloud security	By request	orca.security
OWASP Mobile Application Security	Security standards and testing guide for mobile applications	Open source	mas.owasp.org
OWASP Top 10	Reference standard for web application security risks	Open source	owasp.org/www-project-top-ten
OX	Application security software	By request	ox.security
Palo Alto Networks Checkov	Scan IaC for misconfigurations and security risks	Open source	checkov.io
Palo Alto Networks Cortex XDR	Extended detection and response	By request	paloaltonetworks.com/cortex/cortex-xdr
Palo Alto Networks Prisma Access	Secure access service edge	By request	paloaltonetworks.com/sase/access
Palo Alto Networks Prisma Cloud	Cloud security	By request	paloaltonetworks.com/prisma/cloud
Perforce Perfecto	Secure, scalable application testing platform in the cloud	Trial period	perfecto.io
PortSwigger Burp Suite DAST	Automated DAST scanning	By request	portswigger.net/burp/dast
Protect AI Guardian	AI model security	By request	protectai.com/guardian
Protect AI Layer	Runtime security for AI	By request	protectai.com/layer
Qualys Enterprise TruRisk Platform	Measure, communicate, and eliminate cyber risks	Trial period	qualys.com/enterprise-trurisk-platform
Qualys KCS	Discover, track, and continuously secure Kubernetes and containers	Trial period	qualys.com/apps/container-security
Qualys TotalAppSec	Web app scanning and API security	Trial period	qualys.com/apps/web-app-scanning
Qualys VMDR	Vulnerability management	Trial period	qualys.com/apps/vulnerability-management-detection-response
Rapid7 Command Platform	Unify security operations for visibility, automation, and response	Trial period	rapid7.com/platform
Rapid7 InsightAppSec	Dynamic application security testing	Trial period	rapid7.com/products/insightappsec

DZONE'S 2025 SOFTWARE SUPPLY CHAIN SECURITY SOLUTIONS DIRECTORY

Product	Purpose	Availability	Website
Rapid7 InsightCloudSec	Cloud-native application protection	By request	rapid7.com/products/insightcloudsec
Rapid7 InsightIDR	Security information and event management	Trial period	rapid7.com/products/insightidr
ReversingLabs Spectra Assure	Software supply chain security	Trial period	reversinglabs.com/products/software-supply-chain-security
RunSafe Security	Build-time SBOM generation and vulnerability identification	By request	runsafesecurity.com
Scribe Security	End-to-end software supply chain security	Trial period	scribesecurity.com
Secure Code Warrior	Secure code training	By request	securecodewarrior.com
SecureAuth	Workforce and customer IAM	Trial period	secureauth.com
Security Compass SD Elements	Proactively identify threats and generate security requirements	By request	securitycompass.com/sdelements
Security Innovation	Secure software via testing, training, and lifecycle integration	By request	securityinnovation.com
Semgrep	Lightweight static analysis	Open source	github.com/semgrep/semgrep
SentinelOne Singularity	AI-powered enterprise cybersecurity	By request	sentinelone.com/platform
ServiceNow SecOps	Simplify and automate threat and vulnerability management	By request	servicenow.com/products/security-operations.html
Sigstore	Secure, transparent signing and verification of software artifacts	Open source	sigstore.dev
Snyk AI Trust Platform	Developer security platform	Free tier	snyk.io/platform
SonarQube	Continuously inspect code quality and security	Free tier	sonarsource.com/solutions/secure-by-design-code
Sonatype	Software supply chain automation	By request	sonatype.com/products/software-supply-chain-management
SOOS	Application security posture management	Free tier	soos.io
Splunk Enterprise Security	Analytics-driven SIEM for threat detection and response	By request	splunk.com/en_us/products/enterprise-security.html
Splunk SOAR	Security workflow orchestration, automation, and response	Trial period	splunk.com/en_us/products/splunk-security-orchestration-and-automation.html
Spotbugs	Static analysis for Java code	Open source	spotbugs.github.io
StackHawk	Dynamic application and API security testing	Trial period	stackhawk.com
Tenable One	Unify exposure visibility, risk prioritization, and remediation actions	By request	tenable.com/products/tenable-one
TUF	Framework for securing software update systems	Open source	theupdateframework.io
Tigera Calico Commercial Editions	Unified network security and observability	Trial period	tigera.io/tigera-products/calico-commercial-editions
Tigera Calico Open Source	Container and Kubernetes networking and security	Open source	tigera.io/tigera-products/calico
Trend Vision One	Holistic cybersecurity	Trial period	trendmicro.com/en_us/business/products/one-platform.html
TruffleHog	Secrets scanning engine	Free tier	trufflesecurity.com
Varonis	Unified data security	By request	varonis.com/data-security-platform

DZONE'S 2025 SOFTWARE SUPPLY CHAIN SECURITY SOLUTIONS DIRECTORY

Product	Purpose	Availability	Website
Veracode	Application risk management platform	By request	veracode.com
Vercara UltraDDoS Protect	Cloud-based DDoS protection and mitigation	By request	vercara.digicert.com/ddos-protection
Vercara UltraWAF	Cloud-based web app firewall protection	By request	vercara.digicert.com/ultrawaf
vDefend Advanced Threat Prevention	Advanced threat prevention	By request	vmware.com/products/cloud-infrastructure/vdefend-advanced-threat-prevention
Wallarm	Application and API security	By request	wallarm.com/product/api-security-overview
Wapiti	Web application vulnerability scanner	Open source	wapiti-scanner.github.io
Waratek Secure	Enterprise-grade app and API security	By request	waratek.com/products
Wiz Cloud Security Platform	CNAPP	By request	wiz.io
Xygeni All-in-One AppSec Platform	Secure application development and delivery	By request	xygeni.io
ZAP	Web application security scanner	Open source	zapproxy.org
Zimperium MAPS	Mobile application protection suite	By request	zimperium.com/maps
Zscaler Zero Trust Exchange Platform	AI-powered zero trust	By request	zscaler.com/products-and-solutions/zero-trust-exchange-zte



SOFTWARE DESIGN AND ARCHITECTURE

Cloud Architecture | Containers | Security
Integration | Microservices | Performance

Software design and architecture focus on the development decisions made to improve a system's overall structure and behavior in order to achieve essential qualities such as modifiability, availability, and security.

The Zones in this category are available to help developers stay up to date on the latest software design and architecture trends and techniques.

VISIT THE ZONE



TUTORIALS



TECHNIQUES



CODE SNIPPETS



RESEARCH



3343 Perimeter Hill Dr, Suite 100
Nashville, TN 37211
888.678.0399 | 919.678.0300

At DZone, we foster a collaborative environment that empowers developers and tech professionals to share knowledge, build skills, and solve problems through content, code, and community. We thoughtfully — and with intention — challenge the status quo and value diverse perspectives so that, as one, we can inspire positive change through technology.

Copyright © 2025 DZone. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by means of electronic, mechanical, photocopying, or otherwise, without prior written permission of the publisher.