Takeaways from Government and Industry Conversations on Software Supply Chain Security

MITRE Software and Supply Chain Assurance (SSCA) Fall Forum

Laurie Williams · North Carolina State University

Secure Software Supply Chain Center (S3C2) · September 23, 2026



S3C2

SECURE SOFTWARE SUPPLY CHAIN CENTER

NC STATE UNIVERSITY

Carnegie Mellon University

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

NSF

Secure Software Supply Chain Summits [Meetings]
Why we convene: honest sharing to guide research

Goals of the summits/meeting

1. Share practical experiences and challenges
2. Form new collaborations
3. Learn practitioners' challenges to inform our research direction

Chatham House Rule

You may use what was discussed.

You may not reveal who attended, who said it, or where they work.

This deck follows the rule: no participants or affiliations are named.

Small groups · Participant-chosen topics · Reviewed reports on arXiv

Three conversations in 2026

Industry summit

Feb 12 · East Coast

14

participants from 12 institutions

Topics chosen by pre-summit vote; takeaways gathered live with polls

slido

Government meeting

Jul 17 - Here

17

practitioners from 12 agencies

Topics chosen by pre-meeting survey

Industry summit

Sep 10 · West Coast

16

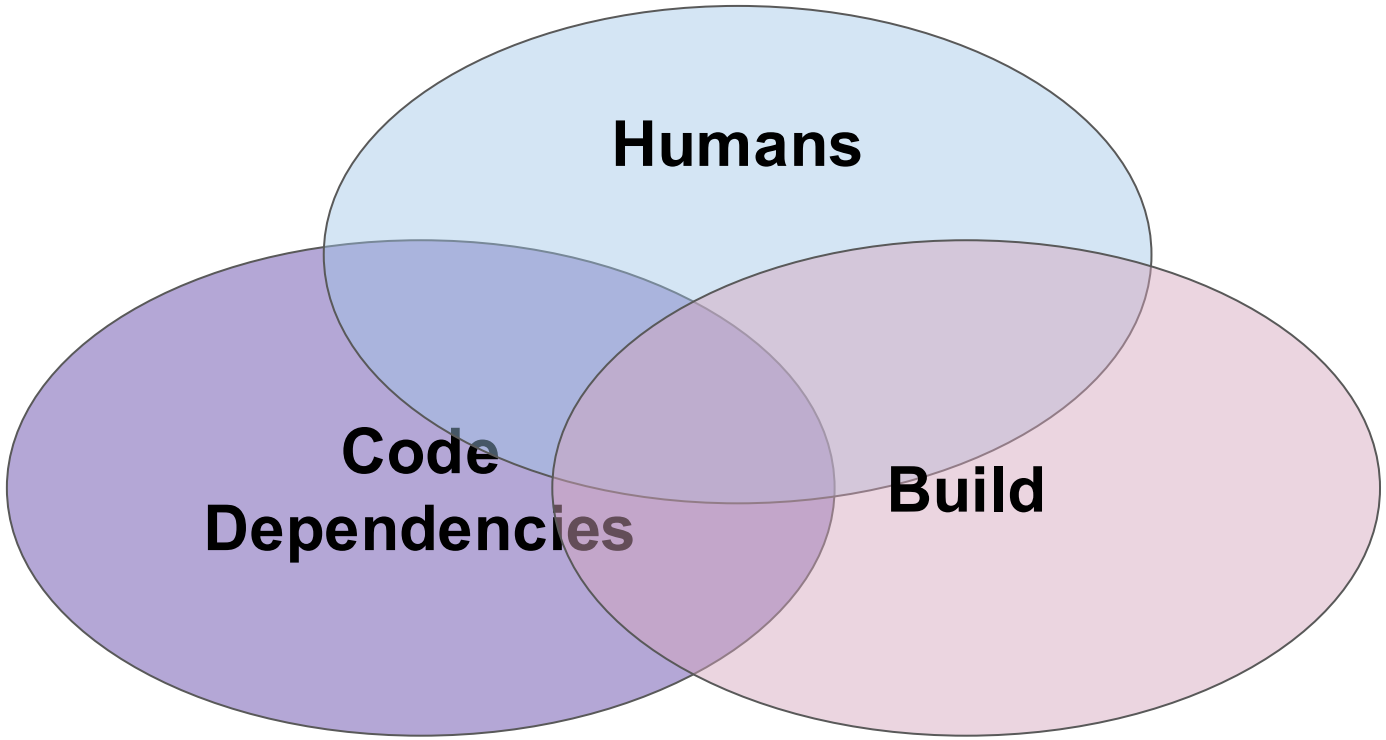
Participants from 8 institutions

Topics chosen by pre-summit vote; takeaways gathered live with polls

slido

Same format, shared topics: we can track what continues and what is new

Attack Vectors Through the Software Supply Chain



Six takeaways

1. AI accelerates old problems
2. “Human in the loop” needs a definition
3. Models, agents and skills join the supply chain
4. Provenance and intent become the control plane
5. Open source and its data are under strain
6. Culture and incentives decide what gets done

TAKEAWAY 1 · 1 OF 2

AI accelerates old problems

7 of 13

Sep poll respondents said code and vulnerability reports are growing faster than they can review and fix

~3 days

critical patching window after CISA BOD 26-04, as agency participants described it

~10×

more open source projects abandoned within 18 months, industry participants *predicted*

The pattern across all three: AI raises the volume of findings and reports faster than people can absorb them.

TAKEAWAY 1 · 2 OF 2

AI accelerates old problems

Industry · Feb, Sep

- Detection and response described as “effectively 100% AI” (Feb)
- Maintainers flooded with AI-found reports may “walk away” (Feb)
- AI helps in narrow cases; pairing it with deterministic techniques works better (Sep)

Government · Jul

- “Analyst bloat”: AI tools produce far more output, including 600-page design documents
- Adjudicating AI results can take as long as doing the task manually
- Disclosures can become exploits before an official patch lands

“Finite bugs or endless vulnerabilities? The answer changes the strategy.” (Sep)
If vulnerabilities are finite, AI could help defenders win.



TAKEAWAY 2 · 1 OF 2

“Human in the loop” needs a definition

Direction: less human review per change

Two-person review

Still useful, but no longer sufficient, and untenable for single-maintainer projects

AI as judge, humans validate

For example, 4 LLMs plus 1 person; humans review the test strategy, not every test

No-person review

Agency participants describe “no person code review” as an emerging practice

TAKEAWAY 2 · 2 OF 2

“Human in the loop” needs a definition

Government · Jul

- For AI use broadly, human-in-the-loop is the prevailing policy, driven by risk posture
- Some hope agencies never deviate from that oversight as pressure to automate grows
- Yet code review is drifting toward “no-person” review (previous slide) — unresolved

Industry · Feb, Sep

- Human approval of AI input via a review should not be “merely a blame sink.”
- Who accepts the liability, and who sets the risk budget?
- The perception remains that the probably of actually being exploited is low.

Policy says humans stay in the loop; practice is already drifting away from it.

TAKEAWAY 3 · 1 OF 2

Models, agents and skills join the supply chain

Models

- Opaque third-party components; only 2–3 viable US providers (Sep)
- Prefer established frontier models over custom or fine-tuned ones (Feb)
- Foreign-adversary models are nearly disqualifying in agency vendor vetting (Jul)

Agents

- Sprawl proxy: 500 models stood up per-agent hit production in Jan 2026 — still too slow (Feb)
- Ephemeral sub-agents break traditional IAM; agents need identities of their own (Feb)
- Shadow AI is real and hard to detect (Feb)

TAKEAWAY 3 · 2 OF 2

Models, agents and skills join the supply chain

Skills, plugins and MCP

- “Skills and plugins are new attack vectors; we are reinventing package management” (Sep)
- Transitive skill dependencies add risk; approved-path controls are not yet in place (Sep)
- MCP servers can give agents a vetted source of dependency data (Feb)

Bills of materials

- CycloneDX and SPDX add AI metadata; CISA and partners published AI BOM minimum elements (May 2026)
- Agents may need to be watched in real time and vetted before they're trusted (Jul, Feb)
- AI BOM work centers on models and data, leaving skills and tooling gaps (Sep)

“Models, context, inputs and agent settings are part of the supply chain now.” (Sep)

TAKEAWAY 4 · 1 OF 2

Provenance and intent become the control plane

Source: record intent

- INTENT.md: leave breadcrumbs of stated intent so AI can flag drift (Feb)
- SLSA source track v1.2 cited as a way to see if a project's intent changed (Feb)
- Two-person review alone is no longer sufficient (Feb)

Build: prove provenance

- Exact reproducible builds are hard to hit; some argue a verifiable build record matters more (Feb, Sep)
- Agencies push toward reproducible vendor builds to test independently (Jul)
- New controls: OIDC on reads, actions pinned to commit hashes. Still unresolved: long-lived tokens (Feb, Sep)



TAKEAWAY 4 · 2 OF 2

Provenance and intent become the control plane

Release gate: enforce

- Registries can now gate deployment: no push without a verified SBOM and attestations. (Feb)
- Detecting a violation is worthless unless something can actually block it (Feb)
- Cooldowns: is 3-days/a week enough, and what if everyone waits? (Feb, Jul)

Procurement: require it

- SLSA level 2 in contracts is more actionable than asking vendors how they verify builds (Jul)
- No harmonized requirements across agencies fuels vendor pushback (Jul)
- Vendors who submit false attestations risk False Claims Act liability (Jul)

Open tension: do these controls hold against AI agents in the pipeline? “Now LLMs do the same” (Feb). Standards like SLSA need to cover agents too (Sep).



TAKEAWAY 5 · 1 OF 4

Open source and its data are under strain

Still unresolved since 2023

- No agreed definition of end of life; CISA settled on “end of support” (Jul)
- Health signals stay subjective; Scorecard is “helpful for context,” not a gate
- The Cyber Resilience Act turns shipping end-of-life components into a legal exposure

Government angle · Jul

- Requiring software to be free of “dead code” may push vendors to build proprietary components with AI instead of maintaining open source
- MITRE Hipcheck and OpenEOX discussed as ways to find abandoned components

“We want open source to thrive. Keeping it healthy takes work.”



TAKEAWAY 5 · 2 OF 4

Open source and its data are under strain

New in 2026

Reducing contributors

Projects not accepting contributions of new/unknown contributors (Sept)

Inlining/Vendoring/backporting

Copy/fix only the lines needed to remove vulnerability (Feb, Sept) rather than wait for fix from unknown//untrusted developer (Sept)

Private forks and reimplementation

Cheaper with AI, but they lose shared defense while attackers still share techniques … desire for OSS to live (Sep)

Agents as maintainers

Pruning dead code and “cron job” pull requests that swap out unmaintained components (Feb)



TAKEAWAY 5 · 3 OF 4

Vulnerability data is losing trust

38%

of AI-recommended dependency versions were hallucinated or suggested a downgrade to a known-vulnerable version

Sonatype 2026 report, cited Feb

Vulnerability data

- NVD and CVE seen as “on shaky ground”; records lose the fidelity and blast-radius data needed to prioritize (Feb)
- Malicious commits almost never become CVEs, so they are hard to broadcast (Feb)
- A discussed ~20% of security fixes ship without a CVE (Sep)

LLMs are “objectively bad at picking dependencies, specifically versions” (Feb)

TAKEAWAY 5 · 4 OF 4

SBOMs, VEX and deciding what to fix

SBOM and VEX in practice

- SBOMs arrive with procurement, but ingestion practices are still under development (Jul)
- VEX adoption is repeating SAST's slow, manual growing pains from the 2010s (Jul)
- Reproducible builds make SBOMs more trustworthy when investigating an attack afterward (Jul)

Deciding what to fix

- Make not updating the harder path: written justification, hard gates at CVSS 7.5 (Feb)
- Teams avoid saying “not exploitable” for legal reasons; customers “want it fixed” (Feb)
- Moving to not accepting a vulnerability without a PoC (Sept)



TAKEAWAY 6 · 1 OF 1

Culture and incentives decide what gets done

Make the secure path easy

- A “well-lit path”: pre-approved tools and templates make the secure choice the easy one (Feb, Jul, Sep)
- Make security automated, default and transparent; give developers guardrails (Jul)
- Have your 3 most-respected engineers co-write the roadmap of 8-12 new practices; don't hand it down (Feb)

People and skills

- Juniors using AI often lack fundamentals (Feb)
- Money is shifting to data centers and away from security (Feb)
- Volunteer security champions; need training tied to what developers do day to day (Feb)

Same problems, different levers · 1 of 2

Theme	Government (Jul)	Industry (Feb, Sep)
Pace of change	Cooldowns extend the “normal software pipeline”; some agencies get in trouble for lagging	Cooldowns “won’t fly” under pressure to move fast; if everyone waits, discovery is delayed
Human oversight	Human-in-the-loop is the prevailing pattern, with little time for two-person review	Two-person review is not enough; AI judges with human validation
Model choice	Foreign-adversary models nearly disqualifying	Only 2–3 viable US providers; some Chinese models run in isolated production workflows



Same problems, different levers · 2 of 2

Theme	Government (Jul)	Industry (Feb, Sep)
Artifacts and builds	SLSA level 2 and reproducible builds in contracts; SBOM and VEX hard to consume	Exact reproducibility is unrealistic; provenance and registry gates preferred instead
Levers for change	Procurement, FAR risk management, False Claims Act; “two different languages”	CRA fines, customer pressure, executive accountability,

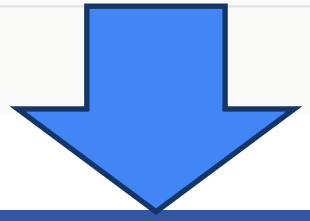
Common ground: AI report volume · undefined EOL · paved paths · talent and skills · SLSA and AI BOMs · two-way trust between agencies and vendors

P-SSCRM: Proactive Secure Software Supply Chain Risk Management framework

The uniting framework for secure software supply chain risk management of code dependencies, build infrastructure, AI models and data, and the humans who maintain them.

Software supply chain frameworks

EO Executive Order 14028 (EO)	**800-161** NIST Cybersecurity Supply Chain Ris...	**SSDF** NIST Secure Software Development Fra...
SSDF-AI NIST Secure Software Development P...	**Self-attestation** DHS/CISA Secure Software S...	**SLSA** Supply-chain Levels for Software Artifact...
BSIMM Building Security In Maturity Model Ver...	**CNCF-SSC** Cloud Native Computing Foundatio...	**OSSF-Scorecard** OpenSSF Scorecard metrics
OWASP-SCVS Open Web Application Security ...		



P-SSCRM

PROACTIVE SECURE SOFTWARE SUPPLY CHAIN RISK MANAGEMENT

https://p-sscrm.org/

Five groups organize the controls

Governance

Policy, compliance, measurement, accountability.

- Perform compliance 5
- Develop security policies 6
- Manage suppliers 5
- Training 3
- Assess and manage risk 4

5 practices · 23 controls

Product

The software you build and its dependencies.

- Develop security requirements 2
- Build security in 6
- Manage component and container choices 4
- Discover vulnerabilities 5
- Manage vulnerable components and containers 2

Environment

Source code, components, build infrastructure.

- Safeguard artifact integrity 6
- Safeguard build integrity 8
- Secure software development environment 10

3 practices · 24 controls

Deployment

Vulnerabilities in what you have shipped.

- Respond to/disclose vulnerabilities 5
- Monitor intrusions/ violations 2

2 practices · 7 controls

Unassigned

Identified, not yet tied to a standard.

- Identify SSC Control Gaps 5

`P.3.5` Prevent component vetting bypass

`D.1.5` Emergency artifact fix

`G.5.5` Support Upstream Dependencies

`E.3.11` Development Environment Scanning Tools

`D.1.7` Establish Response Partnerships

P-SSCRM · 3 OF 8

P-SSCRM controls

In process (v3): updated descriptions and additional assessment questions for AI

P.4.4 **Executable security testing**

Discover vulnerabilities only detected through executable testing

Executable testing, such as done by in-house testers or third-party penetration testers, is performed to find previously-undiscovered vulnerabilities. In-house testers should be guided by design review results, threat modeling, security requirements, and the security mechanisms in security features. Testing should include containers and AI models. Automate security tests as possible. Scope the testing, design the tests, perform the testing, and document the results, including recording and triaging all discovered issues and recommended remediations in the development team's workflow or issue tracking system. Test the build process and scripts.

ASSESS YOURSELF

- How are security features tested?
- How are test cases developed to test the security of an application?
- How is security testing factored into a product's test suite?
- How is executable code testing, such as in-house or external penetration testing, used to find vulnerabilities not found by previous reviews, analysis, and testing?
- How do you conduct executable testing on containers?
- How do you test your build process, scripts, GitHub actions?

STANDARD MAPPINGS

EO 4e(iv), 4e(v) | SSDF PW.8.1, PW.8.2 | SSDF-AI PW.8.1, PW.8.2 | BSIMM ST1.1, ST1.3, ST1.4, ST2.4, ST2.5, ST2.6, ST3.3, ST3.4, ST3.5, ST3.6, PT1.1, PT1.2, PT1.3, PT2.3, PT3.1, CMVM3.4 | 800-161 SA-11 | OSSF-SCORECARD ci-tests | SELF-ATTESTATION 2, 4 | SAMM G-PC-1-A, V-RT-2-A, V-RT-2-B, V-ST-3-A, V-ST-3-B | OSPS OSPS-QA-06, OSPS-VM-05, OSPS-VM-06

ATT&CK TECHNIQUES

T1055 | T1190

ROLES

Developer/testing

Standards → controls

SLSA Supply-chain Levels for Software Artifacts v1.2 (SLSA)

G.1 Perform compliance

- G.1.3 — Produce attestation `Producing artifacts`
- G.1.4 — Deliver provenance `Distributing provenance, Source L3: Signed and auditable provenance`
- G.1.5 — Deliver SBOM `Source L3: Signed and auditable provenance`

G.3 Manage suppliers

- G.3.1 — Security-related contract terms `Verifying artifacts`

P.1 Develop security requirements

- P.1.2 — Software release integrity `Distributing provenance, Source L3: Signed and auditable provenance`

P.3 Manage component and container choices

- P.3.4 — Vetted third-party component and container repositories `Verifying artifacts, Build L1: Provenance exists`

P.4 Discover vulnerabilities

- P.4.1 — Security code review `Source L4: Two-party review`

E.1 Safeguard artifact integrity

- E.1.1 — Safely store release artifacts `Source L3: Signed and auditable provenance`
- E.1.2 — Version control `Source L1: Version controlled`
- E.1.5 — Branch protection `Source L2: Controls`

E.2 Safeguard build integrity

- E.2.1 — Release policy verification `Build L2: Hosted build platform`
- E.2.2 — Verify dependencies and environment `Verifying artifacts`
- E.2.3 — Defensive compilation and build `Verifying build platforms`
- E.2.4 — CI/CD hosting and automation `Build L2: Hosted build platform, Source L3: Signed and auditable provenance`
- E.2.5 — Secured orchestration platform `Build L3: Hardened builds`
- E.2.8 — Hardened and isolated builds `Build L3: Hardened builds`

Roles → controls

Bar colours: Governance Product Environment Deployment

— hover a bar for exact counts

Search a role or control, e.g. G.1.1 or "SBOM"... Expand all Collapse all

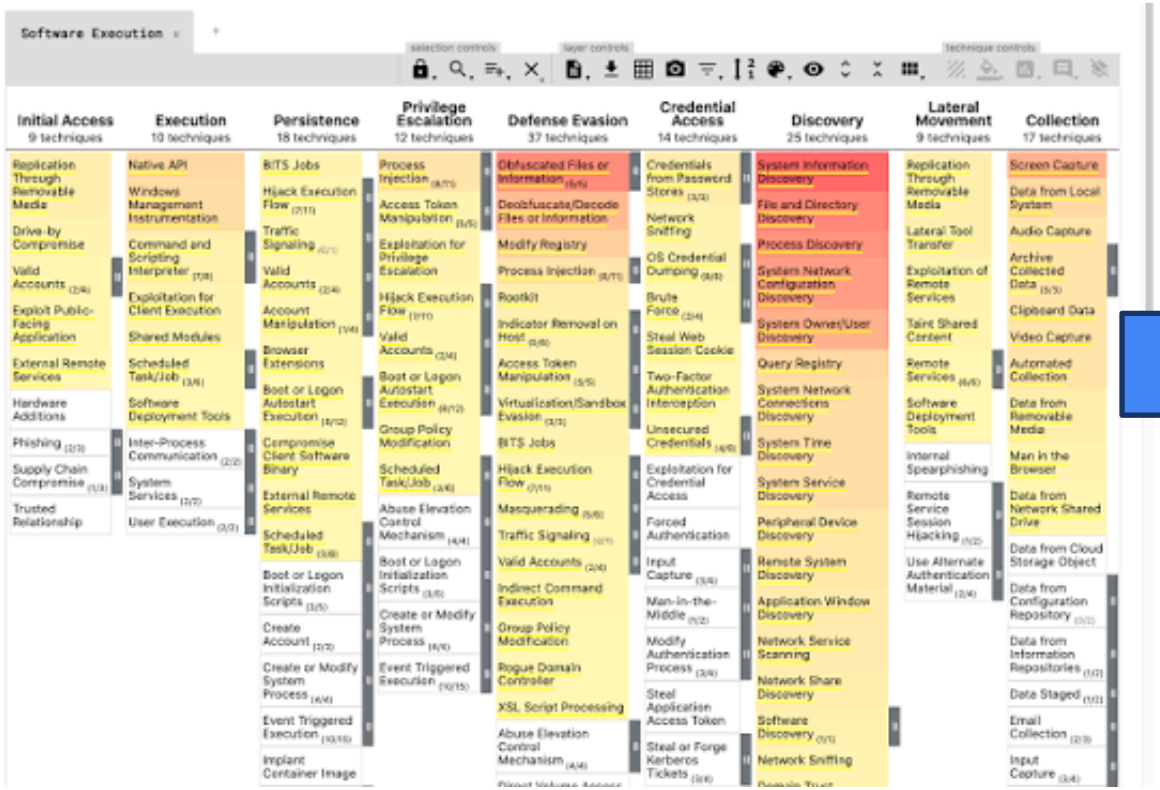
- Business Management 21
- Architect 9
- Developer/developing 20
- Developer/testing 9
- Information Technology 20

Operations/build 17

- G.1.3 Produce attestation
- G.1.4 Deliver provenance
- G.1.5 Deliver SBOM
- P.4.2 Automated security scanning tools
- P.5.2 Dependency update
- E.1.1 Safely store release artifacts
- E.2.1 Release policy verification
- E.2.2 Verify dependencies and environment
- E.2.3 Defensive compilation and build
- E.2.4 CI/CD hosting and automation
- E.2.5 Secured orchestration platform
- E.2.6 Reproducible builds
- E.2.7 Build output
- E.3.9 Ephemeral credentials
- D.2.2 Build process monitoring
- P.3.5 Prevent component vetting bypass
- E.3.11 Development Environment Scanning Tools

- Operations/deploy 12
- Software security 47

MITRE ATT&CK techniques → P-SSCRM controls → other framework controls



Governance
Policy, compliance, measurement, accountability.

- Perform compliance 5
- Develop security policies 6
- Manage suppliers 5
- Training 3
- Assess and manage risk 4

5 practices · 23 controls

Product
The software you build and its dependencies.

- Develop security requirements 2
- Build security in 6
- Manage component and container choices 4
- Discover vulnerabilities 5
- Manage vulnerable components and containers 2

Environment
Source code, components, build infrastructure.

- Safeguard artifact integrity 6
- Safeguard build integrity 8
- Secure software development environment 10

3 practices · 24 controls

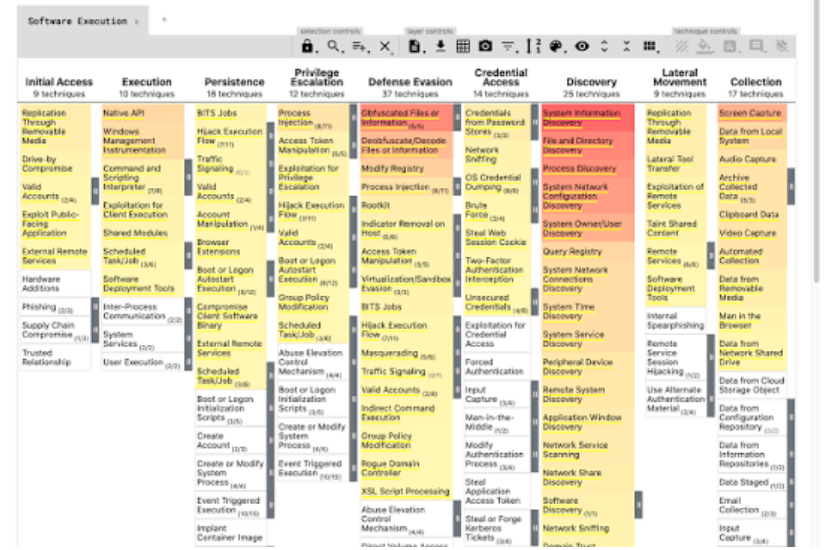
Deployment
Vulnerabilities in what you have shipped.

- Respond to/disclose vulnerabilities 5
- Monitor intrusions/ violations 2

2 practices · 7 controls

Cyberthreat intelligence → MITRE ATT&CK → top controls





Task	Frameworks
E.3.3 Role-based access control	800-161, CNCF
D.2.1 System monitoring	800-161
E.3.7 Boundary protection	800-161
E.3.6 Monitor changes to configuration settings	800-161
E.3.11 Environmental scanning tools	[novel]
P.5.2 Dependency update	SSDF, BSIMM
P.2.1 Security design review	SSDF, BSIMM
E.3.4 Information flow enforcement	800-161
G.2.6 Protection of information at rest	800-161
D.1.2 Risk-based vulnerability remediation	SSDF, 800-161

Based on Solarwinds, XZ Utils, log4J

Organization interview-based assessment

One assessment engagement ➔ 10 framework analyses comparisons



Invitation: Measure and Improve

S3C2 · SOFTWARE SUPPLY CHAIN RISK MANAGEMENT

WE WANT YOU



FOR A P-SSCRM V3 ASSESSMENT

5	16	78
GROUPS	PRACTICES	CONTROLS

https://p-sscrm.org/

S3C2 · SOFTWARE SUPPLY CHAIN RISK MANAGEMENT

WE WANT YOU



TO PARTICIPATE IN A 2027 SUMMIT

12	152	2027
SUMMITS SO FAR	PARTICIPANTS TO DATE	NEXT COHORT

Nominate yourself or a colleague to take part
s3c2.org/software-industry

Thank you

To every summit participant for their time, openness and thoughtful contributions.

Organizers

Laurie Williams, William Enck, Dominik Wermke, Ranindya Paramitha, Alexander Cooper and Yining Hong, with Michel Cukier, Christian Kästner and Alexandros Kapravelos

This material is based upon work supported by the National Science Foundation Grant Nos. 2207008. Any opinions expressed are those of the presenter and do not necessarily reflect the views of the National Science Foundation.













Sharing = Learning = Research Direction = Progress



Read the summit reports

s3c2.org/software-industry/

Reports are also published on arXiv

Contact me:
Laurie Williams
Laurie_Williams@ncsu.edu

Questions?