How are We Doing with Adopting Tasks to Reduce Software Supply Chain Security Risk?

Laurie Williams
Supply Chain Security as an (inter)national priority

Executive Order on Improving the Nation’s Cybersecurity

MAY 12, 2021 • PRESIDENTIAL ACTIONS

(Section 4e) Within 90 days of publication of the preliminary guidelines pursuant to subsection (c) of this section, the Secretary of Commerce acting through the Director of NIST, in consultation with the heads of such agencies as the Director of NIST deems appropriate, shall issue guidance identifying practices that enhance the security of the software supply chain. Such guidance may incorporate the guidelines published pursuant to subsections (c) and (i) of this section.
But, what “should” we do?
And, what’s everyone else doing?
Doing secure software supply chain science: empirical studies to answer those questions
Chatham House Rules and other non-disclosures of company/agency identification

I could tell you, but then I’d have to kill you.
But, what “should” we do?
Oh, so many guiding frameworks ...

NIST Special Publication 800-218

Secure Software Development Framework (SSDF) Version 1.1:
Recommendations for Mitigating the Risk of Software Vulnerabilities

NIST Special Publication
NIST SP 800-161r1

Cybersecurity Supply Chain Risk Management Practices for Systems and Organizations
And also ...

Software Supply Chain Best Practices

BSIMM 13

Software Component Verification Standard
Proactive Software Supply Chain Risk Management (P-SSCRM) framework
P-SSCRM Framework - Lifecycle View

Source Integrity

Build Integrity

Application Integrity

Runtime Integrity

Green: Business management
Red: Architect/Developer
Blue: IT
Purple: DevOps
Orange: Software Security

Governance
- A Perform compliance
- B Develop security policies
- C Manage suppliers
- D Training
- E Assess and manage risk

Product
- F Develop security requirements
- G Build security in
- H Manage component and container choices
- I Discover vulnerabilities
- J Manage vulnerable components and containers

Environment
- K Safeguard artifact integrity
- L Safeguard build integrity
- M Secure software development environment

Deployment
- N Respond to/disclose vulnerabilities
- O Monitor intrusions/violations
<table>
<thead>
<tr>
<th>Governance (23 tasks)</th>
<th>Product (19 tasks)</th>
<th>Environment (23 tasks)</th>
<th>Deployment (8 tasks)</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Perform compliance (5)</td>
<td>● Develop security requirements (2)</td>
<td>● Safeguard artifact integrity (6)</td>
<td>● Respond to vulnerabilities (6)</td>
</tr>
<tr>
<td>● Develop security policies (6)</td>
<td>● Build security in; software security (5)</td>
<td>● Safeguard build integrity (7)</td>
<td>● Monitor intrusions/violations (2)</td>
</tr>
<tr>
<td>● Manage suppliers (5)</td>
<td>● Manage component choices (5)</td>
<td>● Secure environment (10)</td>
<td></td>
</tr>
<tr>
<td>● Train (3)</td>
<td>● Discover vulnerabilities (4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>● Assess and manage risk (4)</td>
<td>● Manage vulnerable components (2)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| G.1.1 Org security requirements | EO: 4e(ix)  
SSDF: PO.1.1  
BSIMM: CP1.1, CP1.2, CP1.3, SR1.1, SR2.2, SR3.3  
800-161: SA-15  
CNCF SSC: C: Establish and adhere to contribution policies  
Self-attestation: 2 |
|--------------------------------|---------------------------------------------------------------|
| G.1.2 Software licenses        | 800-161: CM-10  
OWASP SCVS: 5.12  
S2C2F: SCA-2  
CNCF SSC: AU: Scan software for license implications |
| G.1.3 Attestation              | EO: 4e(i)(F), 4e(ii), 4e(v)  
SSDF: PO.3.3  
BSIMM: SM1.4, SR1,3  
800-161: SA-15, AU-2, AU-3, AU-12  
SLSA: Distributing attestation  
Self-attestation: 1f |
Task Naming Convention

P.2.1: Security Design Review

Group  Practice  Task
<table>
<thead>
<tr>
<th>Task Name</th>
<th>Objective</th>
<th>Definition</th>
<th>Question(s)</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.1.1 Organizational security requirements and policies</td>
<td>Organizational security requirements, such as those imposed by standards and regulations, are included in the SDLC.</td>
<td>Identify, document, communicate, and maintain security requirements and policies for the organization's software development infrastructure and secure SDLC. Maintain the requirements and policies over time. Incorporate constraints imposed by standards and regulations and customer-driven security requirements.</td>
<td>Do you have a defined secure SDLC that the engineers are aware of? Do you define security requirements and policies for the organization, its development infrastructure, contributions, and processes? How are these requirements and contributions maintained over time? Are constraints imposed by regulatory and compliance drivers included in these requirements, policies, and the SDLC?</td>
<td>E0: 4e(ix) SSDF: PO.1.1 S3IMM: CP1.1, CP1.2, CP1.3, SR1.1, SR2.2, SR3.3 800-161: SA-15 CNCF SSC: C. Establish and adhere to contribution policies Self-attestation: 2</td>
</tr>
<tr>
<td>G.1.2 Software licenses</td>
<td>Software licenses that conflict with the organization's objectives are identified.</td>
<td>Software licenses may or may not allow certain types of usage, contain distribution requirements or limitations, or require specific action if the software is modified. Risk is increased if the licenses of components are in conflict with an organization's objectives. Software licenses should be documented and tracked to enable tracing the users and use of licenses to access control information and processes according to software usage restrictions. License metadata should be recorded during build and made available in the SBOM.</td>
<td>Do you scan software to check if the license is in compliance with an organization's use policies? Is the process automated? Do you document and track users and uses of software licenses relative to access control policies and software usage restrictions?</td>
<td>800-161: CM-10 OWASP SCVS: 5.12 S3C2F: SCA-2 CNCF SSC: AU: Scan software for license implications E0: 4e(i) E0: 4e(ii), 4e(iii), 4e(v) SSDF: PO.3.3 S3IMM: SM1.4, SR1.3 800-161: SA-15, AU-2, AU-3, AU-12 SLSA: Distributing attestation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Configure tools to generate artifacts to create an audit trail of the use of secure software development practices in a manner that conforms with record retention requirements and preserves the integrity of the findings and the confidentiality of the information. Assign responsibility for creating artifacts that tools cannot generate. Attestation should be immutable and published in the source repository releases, in the package registry, or elsewhere with their existence in a</td>
<td>Is the toolchain configured such that artifacts that attest to using secure development practices and other auditable are recorded consistent with retention requirements? Is responsibility assigned for creating needed artifacts that tools cannot generate? Do you use a framework, like in-toto, to produce authenticated meta-data about artifacts such as for attestation? Do you need to provide self-attestation for your product? Is the attestation immutable and published in the source repository releases, in the package registry, or elsewhere with their existence in a</td>
<td></td>
</tr>
</tbody>
</table>
# Task coverage with all the frameworks

<table>
<thead>
<tr>
<th>Framework</th>
<th>Governance</th>
<th>Product</th>
<th>Environment</th>
<th>Deployment</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-SSCRM</td>
<td>23</td>
<td>19</td>
<td>22</td>
<td>8</td>
<td>73</td>
</tr>
<tr>
<td>EO / SSDF</td>
<td>11</td>
<td>14</td>
<td>4</td>
<td>5</td>
<td>34/34</td>
</tr>
<tr>
<td>Self-attestation</td>
<td>8</td>
<td>12</td>
<td>4</td>
<td>5</td>
<td>23/34 SSDF</td>
</tr>
<tr>
<td>BSIMM</td>
<td>17 [1]</td>
<td>14</td>
<td>2</td>
<td>4</td>
<td>37/125</td>
</tr>
<tr>
<td>SLSA</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>6/6</td>
</tr>
<tr>
<td>OWASP SCVS</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>11/11</td>
</tr>
<tr>
<td>S2C2F</td>
<td>3</td>
<td>7 [1]</td>
<td>3</td>
<td>2 [1]</td>
<td>15/15</td>
</tr>
</tbody>
</table>
Empiricism
And, what’s everyone else doing?
Interview study

- Nine companies
  - Seven large (1000s)
  - Two medium (100s)
- Early adopter / progressive companies
- 61 interviews of approximately 1.5 hours (12/22 - 10/23)
  - 1 Chief Information Security Officer (CISO)
  - 27 Governance (software security group, risk management, vendor management)
  - 23 Product (architect, developer, testers)
  - 10 Environment/Deployment (DevOps, Product Security Incident Response (PSIRT))
Where everybody’s at

<table>
<thead>
<tr>
<th>Rating</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>Does not achieve objective</td>
</tr>
<tr>
<td>0.25</td>
<td>Emerging</td>
</tr>
<tr>
<td>0.50</td>
<td>Progress being made</td>
</tr>
<tr>
<td>0.75</td>
<td>Almost there</td>
</tr>
<tr>
<td>1.00</td>
<td>Achieves objective/exemplary</td>
</tr>
</tbody>
</table>

Average

G.1 Compliance (5)

G.2 Policy (6)

G.3 Suppliers (5)

G.4 Training (3)

G.5 Risk (4)

P.1 Sec Req (2)

P.2 Soft Security (5)

P.3 Comp & Cont Choice (5)

P.4 Discover Vuln (5)

P.5 Comp & Cont Mgmt (2)

E.1 Artifact (6)

E.2 Build (7)

E.3 Devel Environment (10)

D.1 Disclosure (6)

D.2 Monitor (2)

Average
# Top 10 Tasks

(4 Governance, 1 Product, 4 Environment, 1 Deployment)

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Task Name</th>
<th>Firm Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>E.3.1</td>
<td>Authentication</td>
<td>1.00</td>
</tr>
<tr>
<td>P.4.2</td>
<td>Automated security scanning tools</td>
<td>0.97</td>
</tr>
<tr>
<td>G.4.1</td>
<td>Role-based training</td>
<td>0.97</td>
</tr>
<tr>
<td>E.2.7</td>
<td>Build output</td>
<td>0.94</td>
</tr>
<tr>
<td>G.2.3</td>
<td>Roles and responsibilities</td>
<td>0.92</td>
</tr>
<tr>
<td>E.3.7</td>
<td>Boundary protection</td>
<td>0.91</td>
</tr>
<tr>
<td>G.1.2</td>
<td>Software licenses</td>
<td>0.89</td>
</tr>
<tr>
<td>G.2.6</td>
<td>Protection of information at rest</td>
<td>0.86</td>
</tr>
<tr>
<td>D.1.3</td>
<td>Vulnerability disclosure</td>
<td>0.86</td>
</tr>
<tr>
<td>E.3.2</td>
<td>Environmental separation</td>
<td>0.84</td>
</tr>
</tbody>
</table>
## Bottom 11 (due to tie) Tasks
(3 Governance, 4 Product, 3 Environment, 1 Deployment)

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Task Name</th>
<th>Firm Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>E.2.6</td>
<td>Reproducible Builds</td>
<td>0.03</td>
</tr>
<tr>
<td>P.5.1</td>
<td>SBOM consumption</td>
<td>0.03</td>
</tr>
<tr>
<td>P.3.3</td>
<td>Require signed commits</td>
<td>0.08</td>
</tr>
<tr>
<td>G.1.4</td>
<td>Deliver provenance</td>
<td>0.08</td>
</tr>
<tr>
<td>P.3.5</td>
<td>Prevent component vetting bypass</td>
<td>0.14</td>
</tr>
<tr>
<td>G.1.3</td>
<td>Produce attestation</td>
<td>0.17</td>
</tr>
<tr>
<td>D.2.2</td>
<td>Build process monitoring</td>
<td>0.18</td>
</tr>
<tr>
<td>G.1.5</td>
<td>Deliver SBOM</td>
<td>0.19</td>
</tr>
<tr>
<td>E.3.9</td>
<td>Ephemeral credentials</td>
<td>0.22</td>
</tr>
<tr>
<td>E.2.3</td>
<td>Defensive compilation and build</td>
<td>0.25</td>
</tr>
<tr>
<td>P.3.2</td>
<td>Trusted repositories</td>
<td>0.25</td>
</tr>
</tbody>
</table>
Oops! Accidental dependency vulnerability

Nightmare before Christmas: What to know about the Log4j vulnerability

BY NICOLE SGANGA
UPDATED ON: DECEMBER 17, 2021 / 12:44 PM / CBS NEWS
Code dependencies as an attack vector

Code dependencies as a weapon

INTERNATIONAL - UKRAINE INVASION

Russia’s largest bank tells its clients to delay downloading software updates after ‘protestware’ attacks target Russian users

BY NICHOLAS GORDON
March 22, 2022 7:07 AM EDT

node-ipc  

11.1.0 • Public • Published 24 days ago
Key Takeaways

- Adoption of Tasks is dangerously low

- Product Practice P3: Manage component & container choices (5 tasks): average adoption: 0.23
  - P.3.1 Component and container choices: 0.39
  - P.3.2 Trusted repositories: 0.25
  - P.3.3 Require signed commits: 0.08
  - P.3.4 Vetted third-party repositories: 0.31
  - P.3.5 Prevent component vetting bypass: 0.14

- Product Practice P5: Manage vulnerable components (2 tasks): average adoption = 0.24
  - P.5.1 SBOM consumption: 0.03
  - P.5.2 Dependency update: 0.44

- Environment Task E.2.2: Verify dependencies and environment: average adoption 0.28
Key Takeaways

- Third-party vendor’s security/compliance is rarely re-reviewed: Product Task P.4.5 average adoption - 0.58
Build infrastructure as an attack vector

Russian hackers behind SolarWinds hack are trying to infiltrate US and European government networks

By Sean Lyngaas, CNN
Updated 3:27 PM ET, Wed October 6, 2021

DIVE BRIEF

Codecov hack — likened to SolarWinds — targets software supply chain

Published April 23, 2021 • Updated April 30, 2021
Key Takeaways

- Adoption of Tasks is dangerously low

- Environment Practice E2 **Safeguard Build Artifacts** (7 tasks): average adoption 0.42
  - E.2.1 Release policy verification: 0.33
  - E.2.2 Verify dependencies and environment: 0.28
  - E.2.3 Defensive compilation and build: 0.25
  - E.2.4 CI/CD automation and protection: 0.47
  - E.2.5 Secure orchestration platform: 0.64
  - E.2.6 Reproducible builds: 0.03
  - E.2.7 Build output: 0.94
Large Language Models (LLMs) as an attack vector
Key Takeaway

MORE OF THE SAME AHEAD
Another cross-cutting Takeaway

The community is having a technical challenge with building and maintaining a comprehensive asset inventory: Governance Task 0.2.4 average adoption is 0.41

https://www.cornerstoneresults.com/danger-lurks-in-your-inventory
“The Executive Order is forcing industry to adopt security practices that should have been adopted 20 years ago. We want to actually be more secure [reduce software supply chain risk], not just comply.”

-- Summit attendees
What we would **not** know if we looked only at the SSDF

- Components and containers flow pretty freely into an organization without vetting or pre-screening

- A Solarwind—type of attack through the build infrastructure could happen pretty easily
What we would not know if we looked only at the SSDF

Almost no one is requiring SBOMs from their suppliers or using an SBOM to react to security incidents or to identify which components need to be updated.

The “screws need to be tightened” on the security requirements of third-party suppliers and continued compliance with these requirements.
What we would **not** know if we only looked at the SSDF

- Attack vectors that could lead to unauthorized or accidental access and alteration of project artifacts are still viable.

- Attack vectors through the development environments are pretty secure.
Call to Action

- Close down the novel attack vectors through adoption of newer tasks
- Develop tools to make securing the software supply chain easier
Future work

- Publishing P-SSCRM and empirical results
- Risk-based task adoption based on current state of supply chain attacks
- Mapping tasks to MITRE ATT&CK TTPs mitigations and more NIST controls
- Expanding mapped standards to include more non-US sources
- More interviews, longitudinal studies
- Feedback and collaboration welcome!
Thank You!

- Synopsys colleagues
  - Sammy Migues
  - Jamie Boote
  - Ben Hutchison
- Yahoo colleagues
  - Chris Madden
  - DJ Schleen
  - Robert Hines
- NIST
  - Karen Scarfone
- All the interviewees
- NSF
Resources

P-SSCRM v0.4

http://tinyurl.com/2p8xx2b9