An Industry Interview Study of Software Signing for Supply Chain Security





Kelechi G. Kalu
kalu@purdue.edu

Tanmay Singla
singlat@purdue.edu

Chinenye Okafor
okafor1@purdue.edu

Santiago Torres-Arias
santiagotorres@purdue.edu

James C. Davis
davisjam@purdue.edu

PURDUE UNIVERSITY
Elmore Family School of Electrical and Computer Engineering

NSF

CISCO

34TH USENIX SECURITY SYMPOSIUM
AUGUST 13–15, 2025
SEATTLE, WA, USA

Contributions

1. First interview study on software signing in practice.

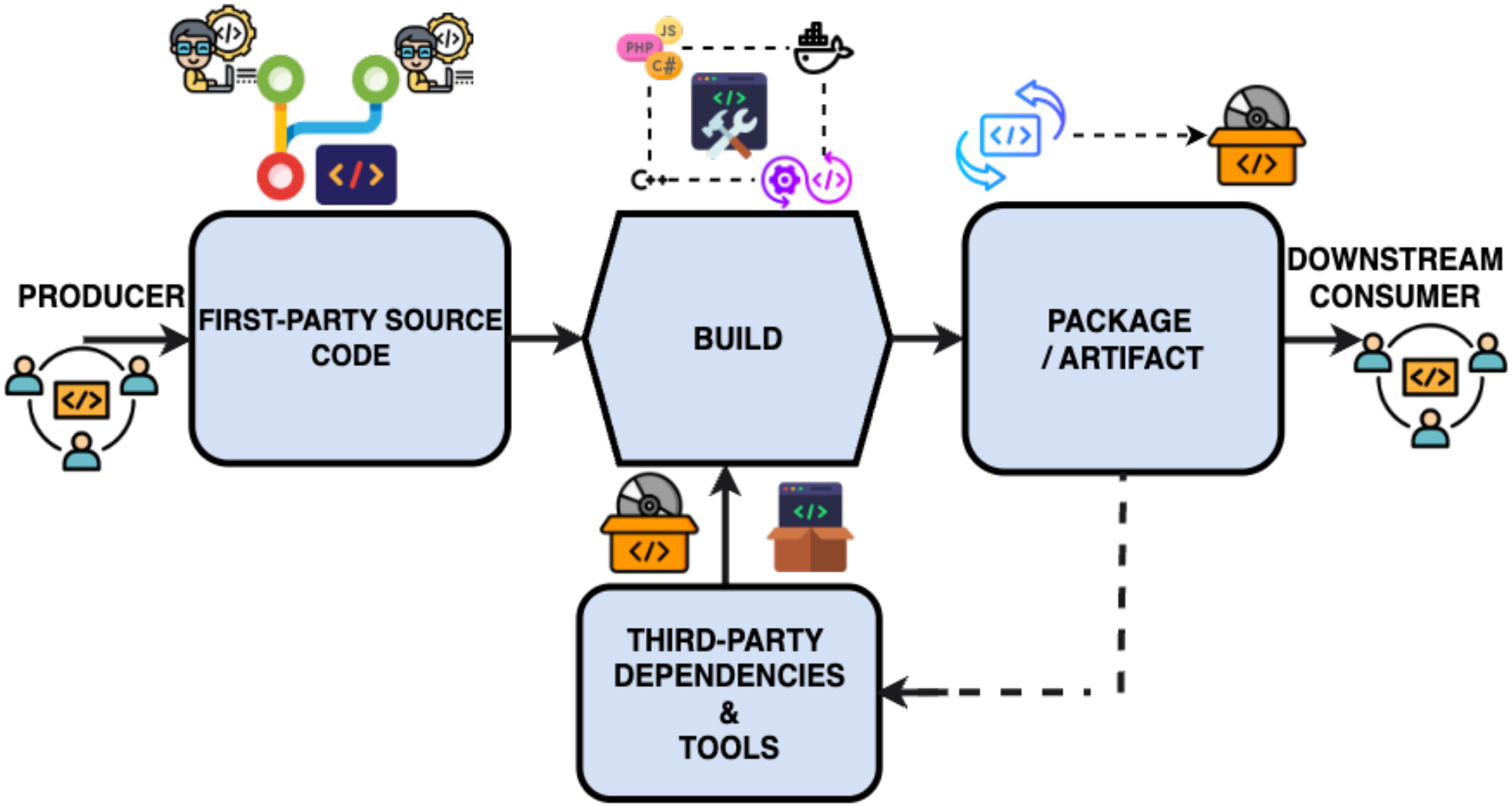
2. Refined supply chain factory model reflecting diverse industry signing workflows.

3. Empirical data on organizations' signing challenges.

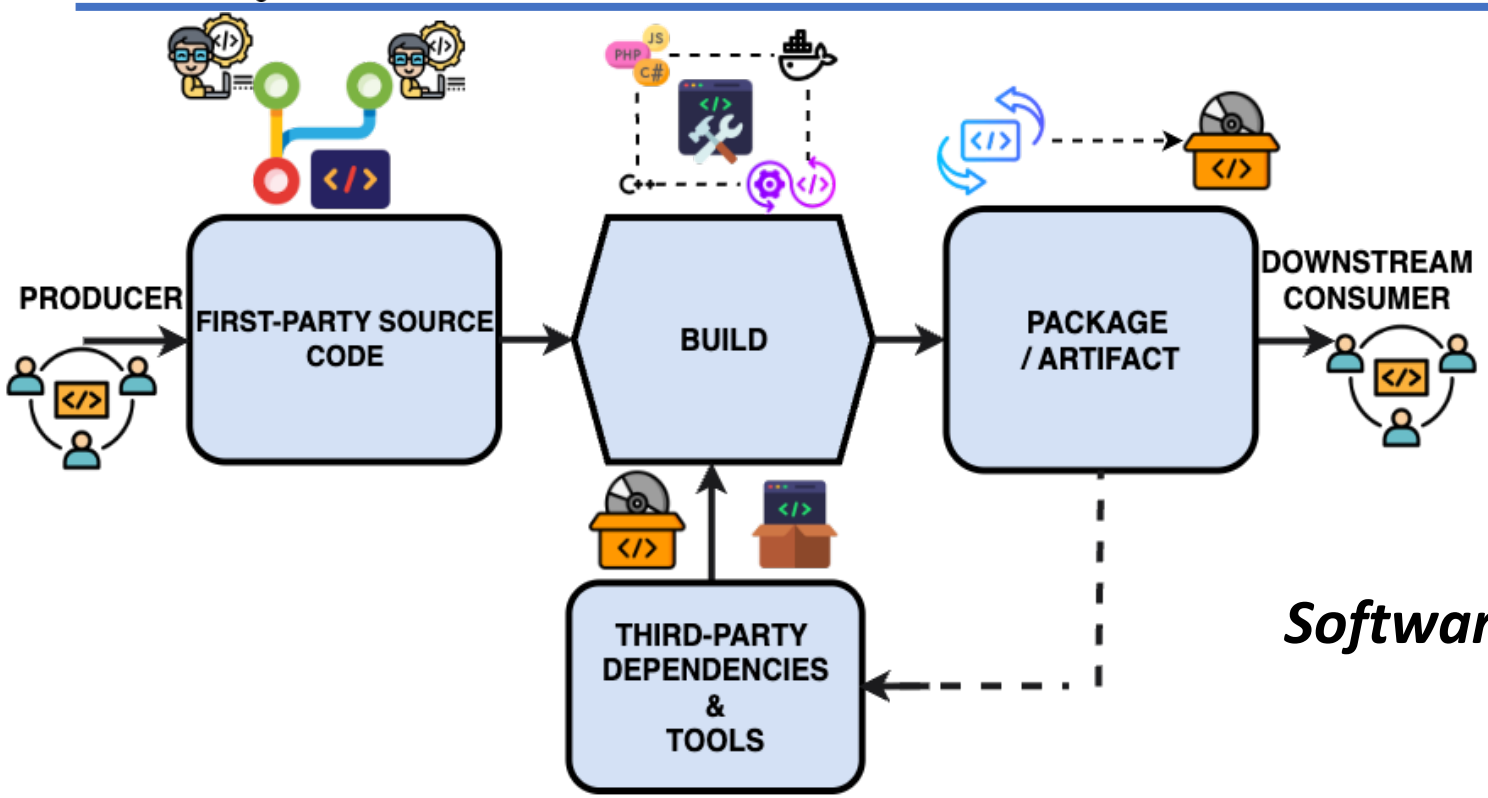
4. Practitioner views on signing importance and impacts of supply chain attacks and regulations.

Motivation & Background

Software Supply Chain



Software Supply Chain – Security Properties

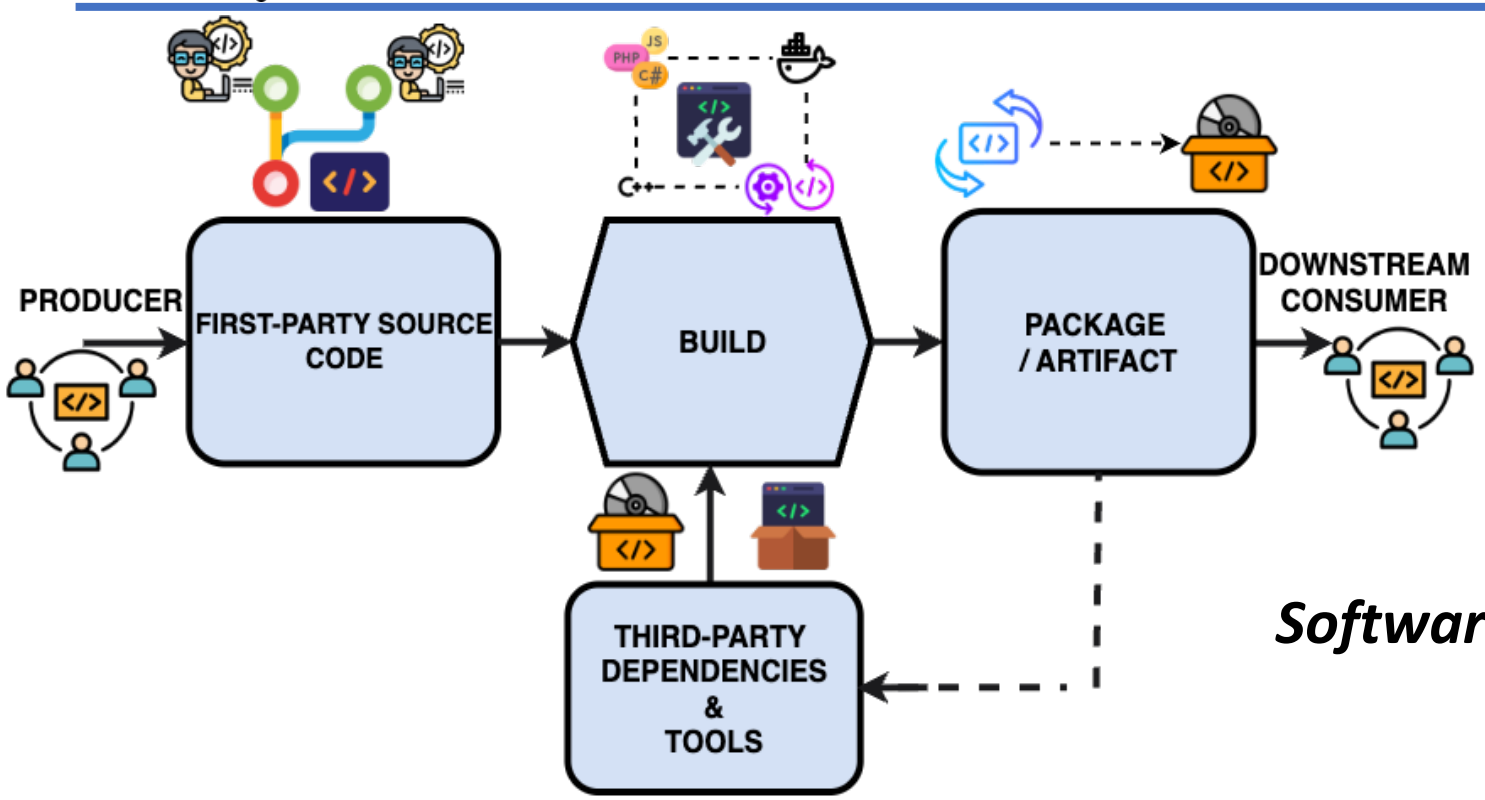


Properties of Secure Software Supply Chains

PROVENANCE

Software Signing – an exemplar of Provenance

Software Supply Chain – Security Properties



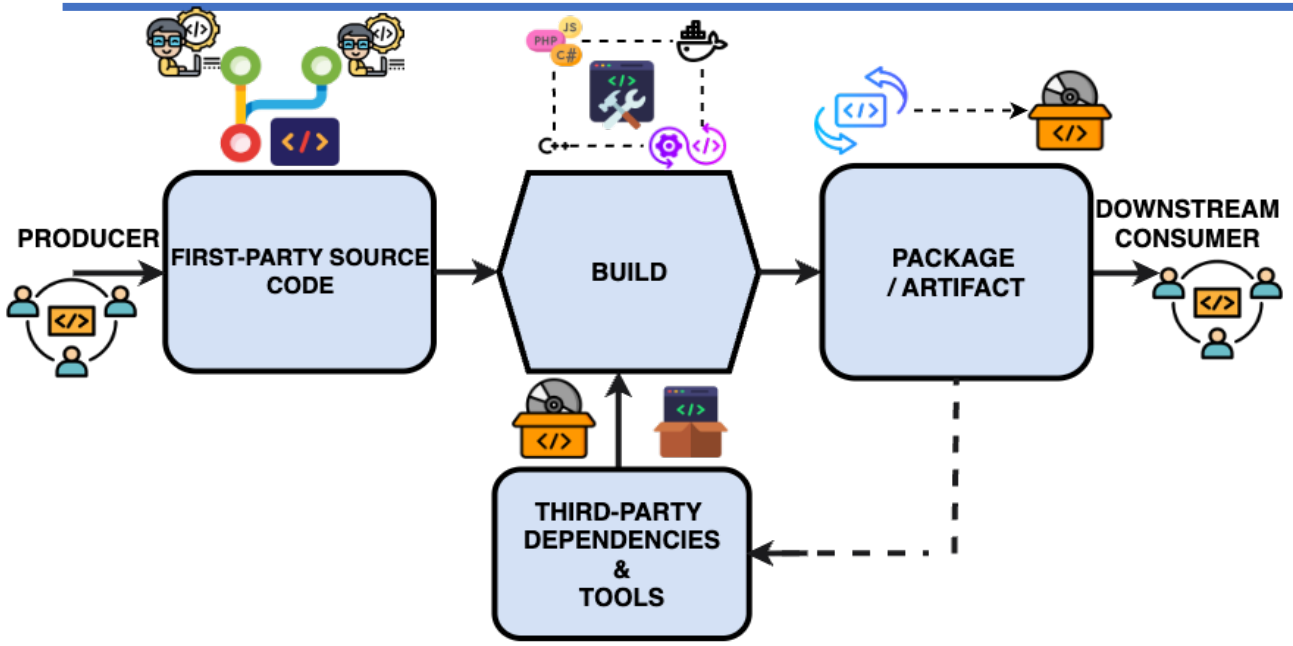
Properties of Secure Software Supply Chains

PROVENANCE

Software Signing – an exemplar of Provenance

sigstore

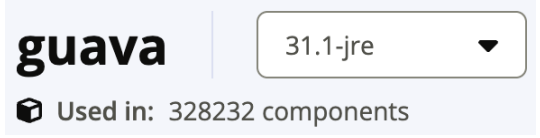
Software Supply Chain – Provenance & Software Signing



Properties of Secure Software Supply Chains

PROVENANCE

Software Signing – an exemplar of Provenance



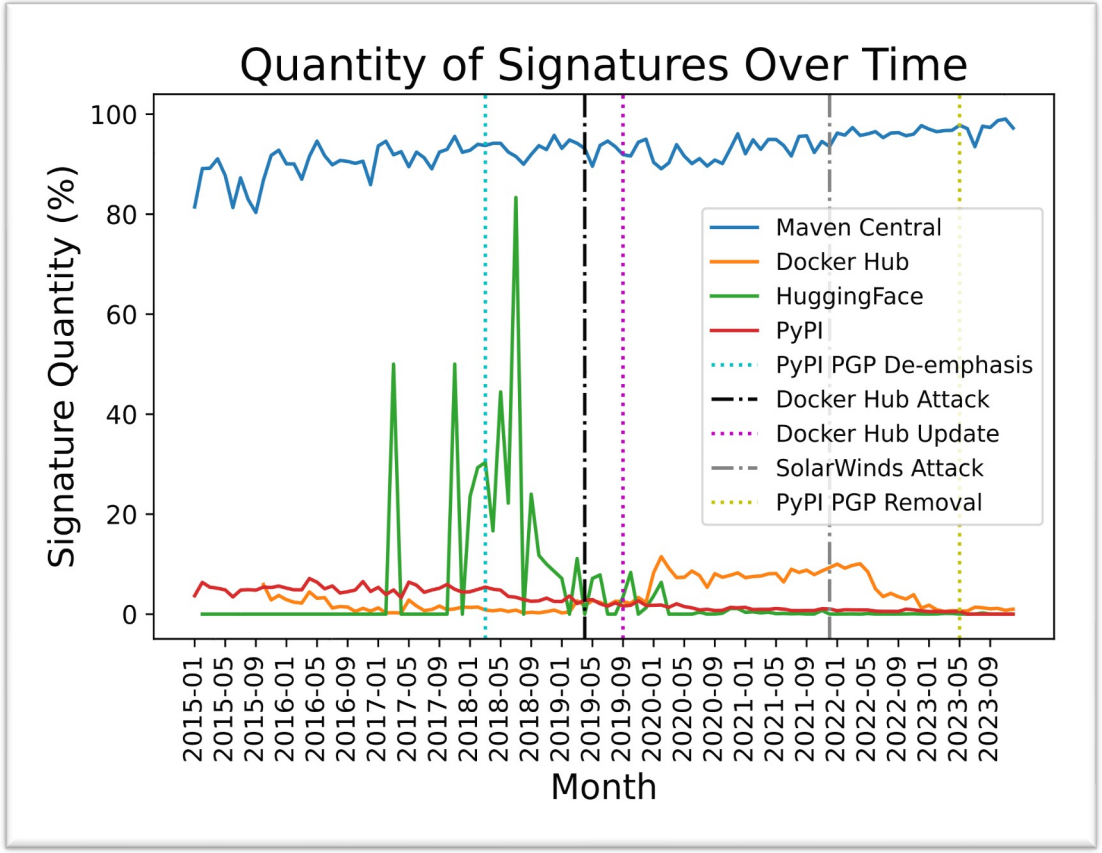
SIGNATURE CREATION

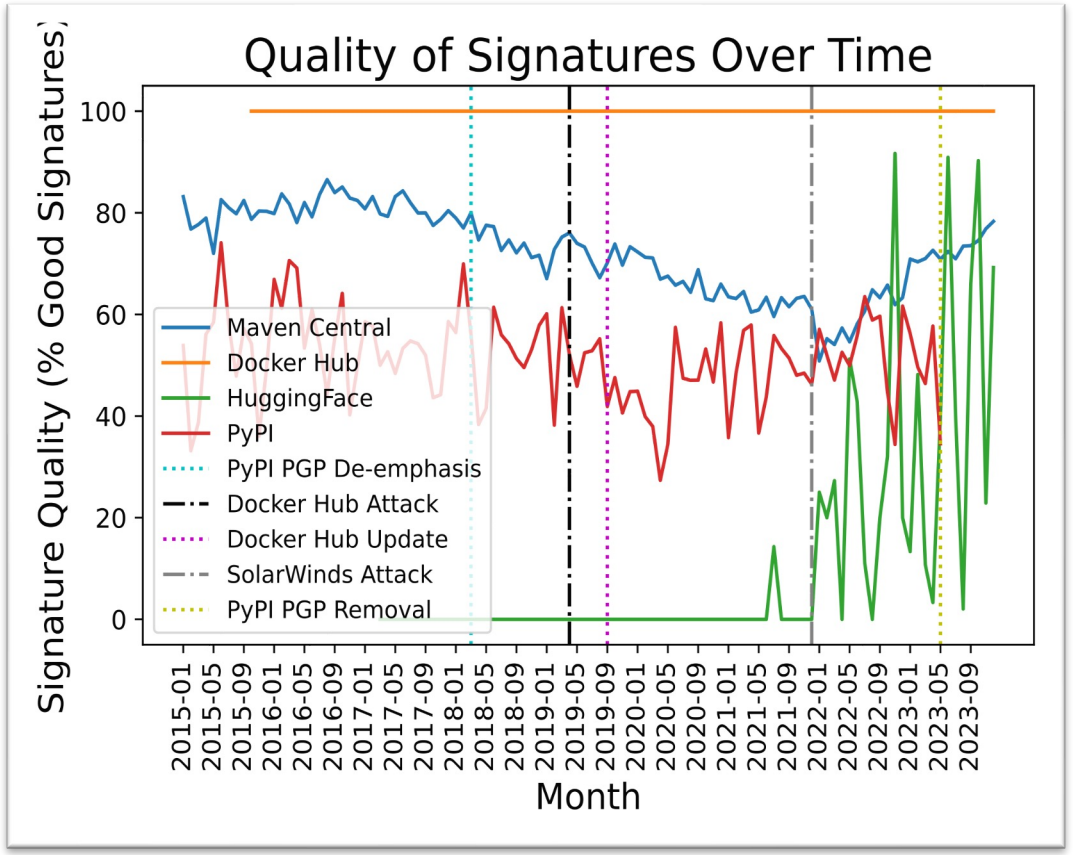
```
signature packet: algo 1, keyid F6D4A1D411E9D1AE
```

SIGNATURE VERIFICATION

```
gpg: Signature made Mon Feb 28 16:18:57 2022 EST
gpg:                using RSA key BDB5FA4FE719D787FB3D3197F6D4A1D411E9D1AE
gpg: Good signature from "Christopher Povirk <cpovirk@google.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner
Primary key fingerprint: BDB5 FA4F E719 D787 FB3D  3197 F6D4 A1D4 11E9 D1AE
```

Software Signing – Current State





Low Quality & Adoption (Schorlemmer et al, IEEE S&P 2024)

Software Signing – Current State

1. Low Quality & Adoption (Schorlemmer et al, IEEE S&P 2024)

2. **Attack Vector**

Example: Microsoft CVE-2020-0601, Adobe Attacks (2012)

3. **Unclear Implementation Guidance**

- Use code signing to help protect the integrity of executables.

Example Recommendation: NIST SP 800-218 (*Pg 9*)

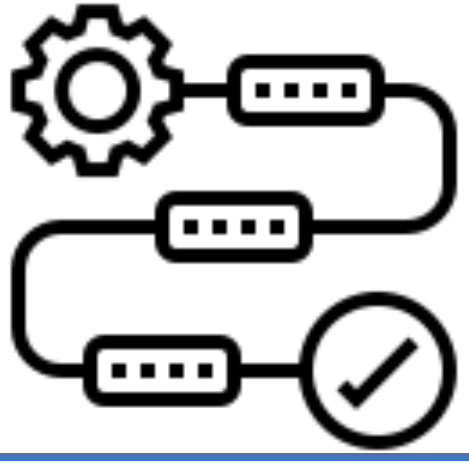
Research Questions

Research Questions

- **RQ1** — Where and **how is software signing implemented** by software teams?
- **RQ2** — What are the **challenges** that affect the implementation and use of software signing?
- **RQ3** — What is the **perceived importance of Software Signing** in mitigating risks?
- **RQ4** — How do **internal and external signing events influence** the adoption of software signing?

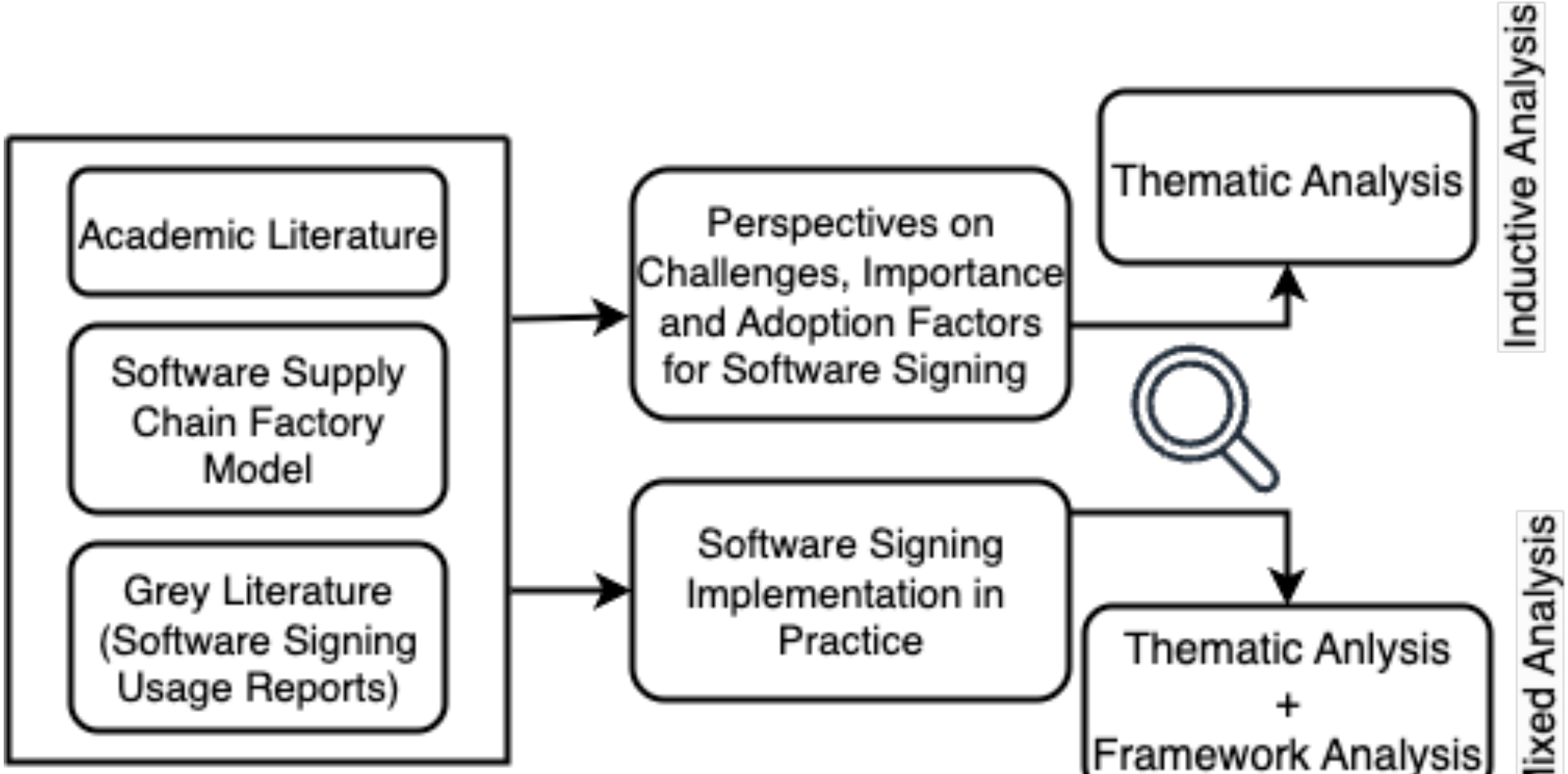
Research Questions

- **RQ1** — Where and **how is software signing implemented** by software teams?
- **RQ2** — What are the **challenges** that affect the implementation and use of software signing?
- **RQ3** — What is the **perceived importance of Software Signing** in mitigating risks?
- **RQ4** — How do **internal and external signing events influence** the adoption of software signing?



Methodology

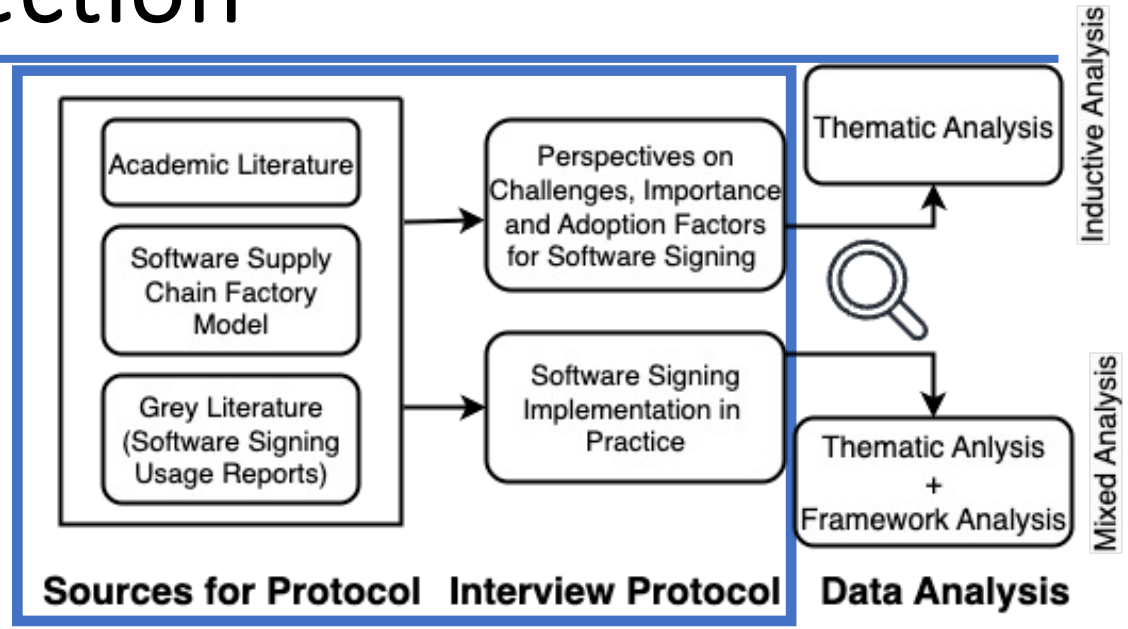
Methodology – Overview



Sources for Protocol **Interview Protocol** **Data Analysis**

Methodology – Data Collection

- Research Instrument – **Semi-structured interviews**.
- **18** Experienced Security Practitioners - 13 Orgs.
- Median years of experience – **13 years.**
- Median Interview Lasted for **50 minutes**.



Methodology – Data Collection

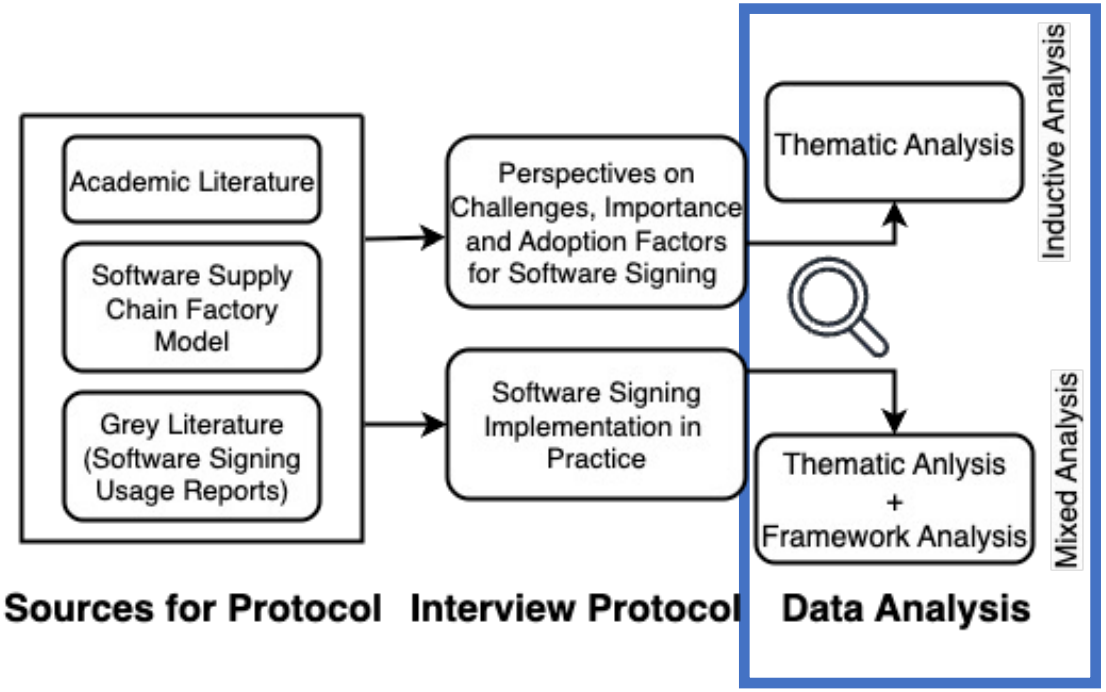
ID	Role	Experience	Software Type
S1	Research leader	5 years	Internal POC software
S2	Senior mgmt.	15 years	SAAS security tool*O
S3	Senior mgmt.	13 years	SAAS security tool*O
S4	Technical leader	20 years	Open-source tooling
S5	Engineer	2 years	Internal security tooling
S6	Technical leader	27 years	Internal security tooling*O
S7	Manager	6 years	Security tooling*O
S8	Technical leader	8 years	Internal security tooling*
S9	Engineer	2.5 years	SAAS security*
S10	Engineer	13 years	SAAS security*
S11	Technical leader	16 years	Firmware*
S12	Technical leader	4 years	Consultancy
S13	Senior mgmt.	16 years	Internal security tool
S14	Research leader	13 years	POC security Software*
S15	Senior mgmt.	15 years	Internal security tooling
S16	Technical leader	26 years	Security tooling
S17	Senior mgmt.	15 years	SAAS
S18	Manager	11 years	Security tooling

Methodology – Data Collection

Type	Breakdown (#Organizations|#Subjects)
Organizational Size (Employee Size)	Small (<100) (4/6), Medium (<1500) (3/4), Large (>1500) (6/8)
Product Area	Digital technology (1/3), SSC Security (2/4), Social technology (1/1), Dev tools (1/1), Telecommunications (1/1), Cloud security (2/3), Aerospace Security (1/1), Internet services (2/2), Cloud + OSS Security (1/1), Cloud + Dev toools (1/1)
Subject Distribution	A (3), B (3), C (1), D (1), E (2), F (1), G (1), H (1), I (1), J (1), J (1), K (1), L (1)

Methodology – Data Collection

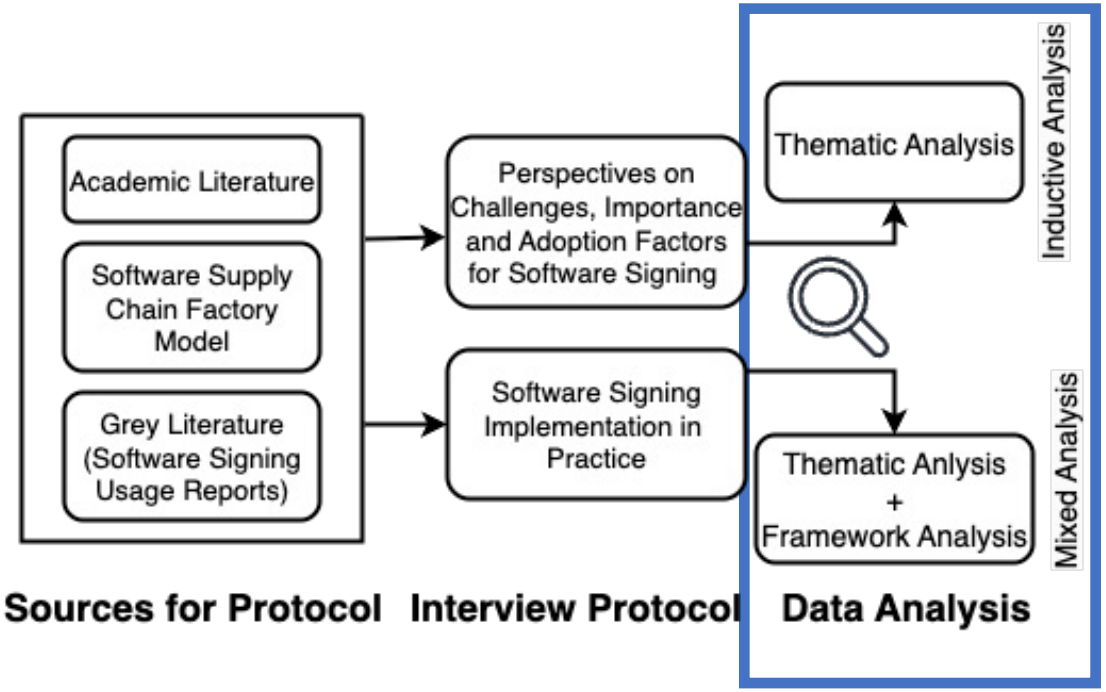
- Analysis Framework – Software Supply Chain Model

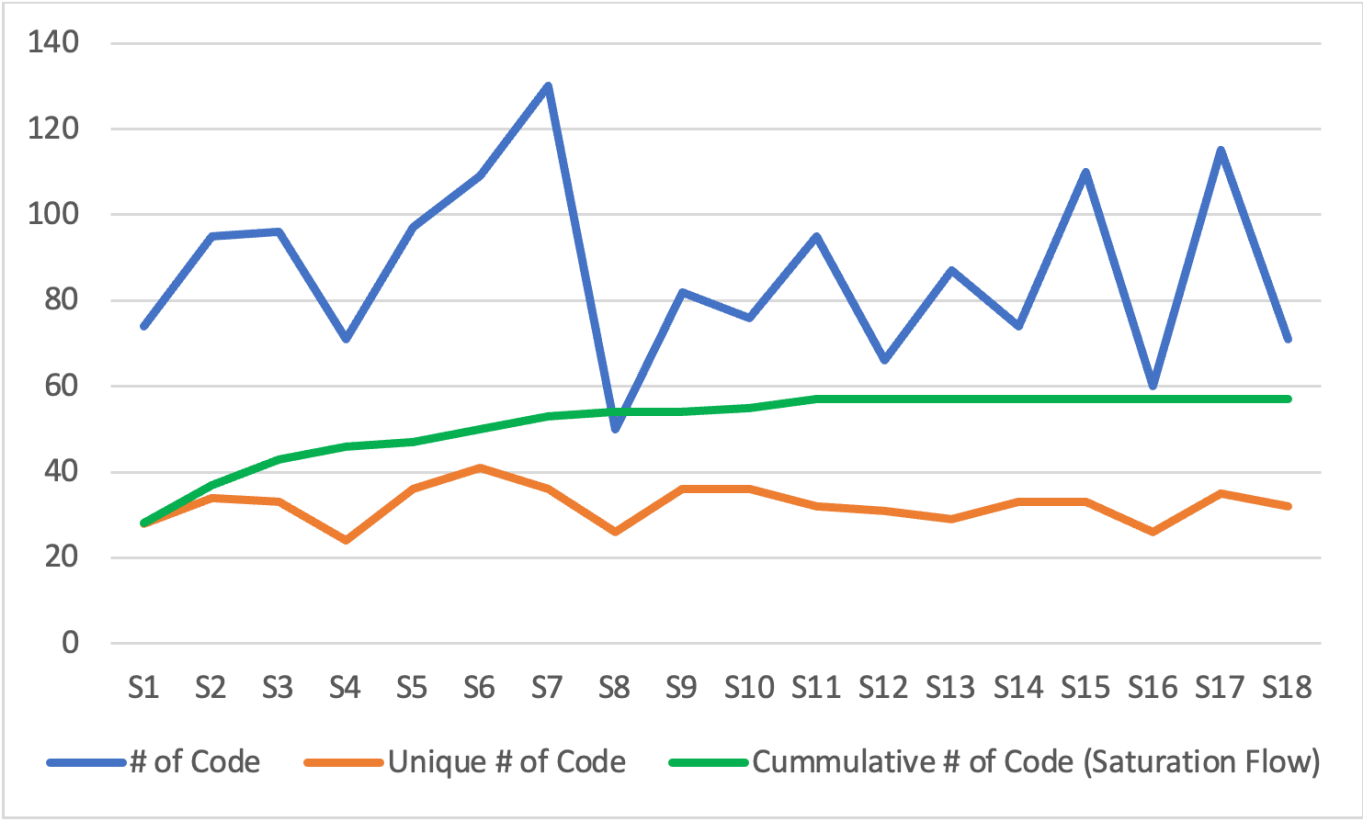


Methodology – Data Collection

- Analysis Framework – Software Supply Chain Model

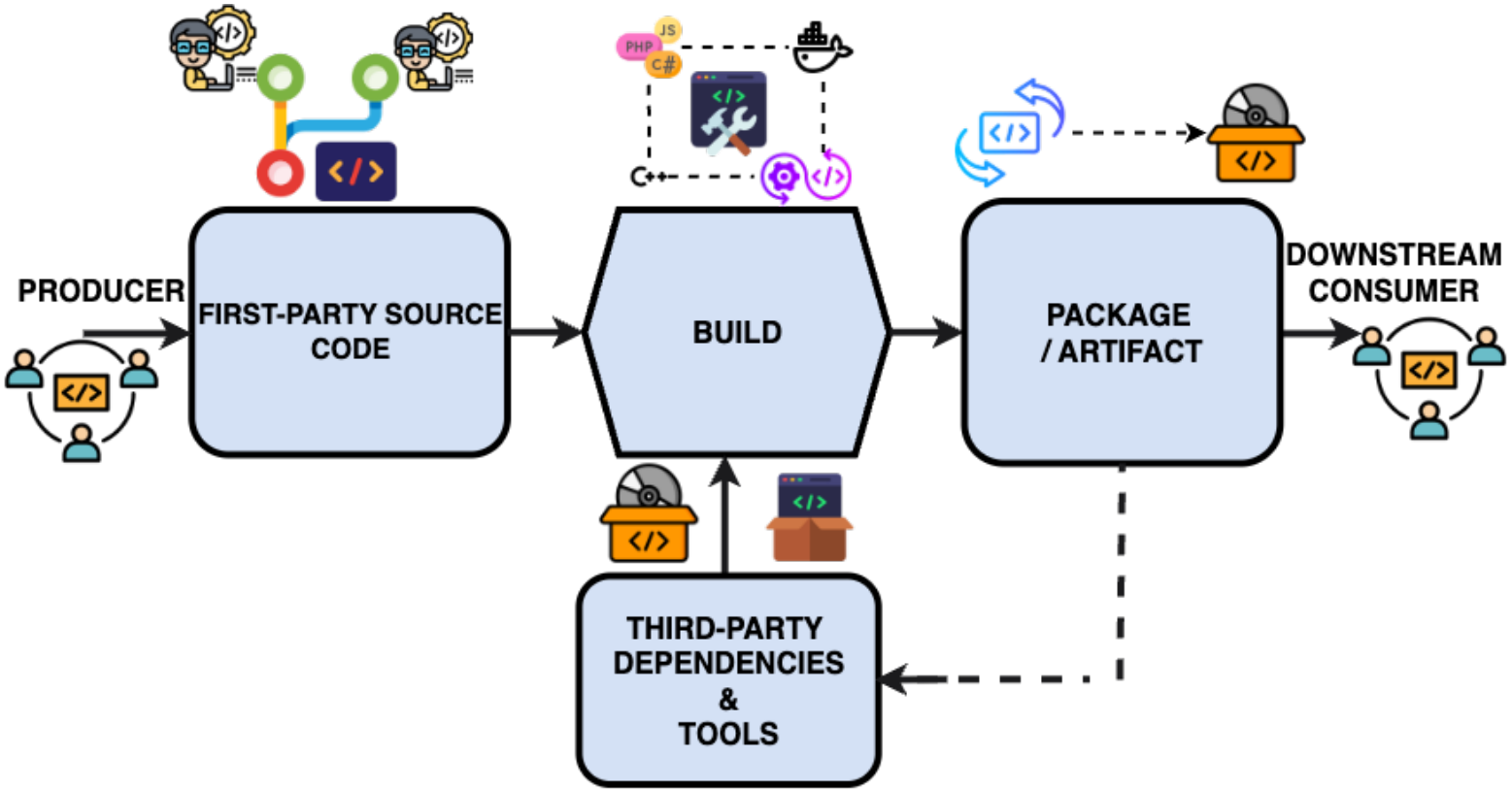
- Achieved **Saturation at the 11th Interview**



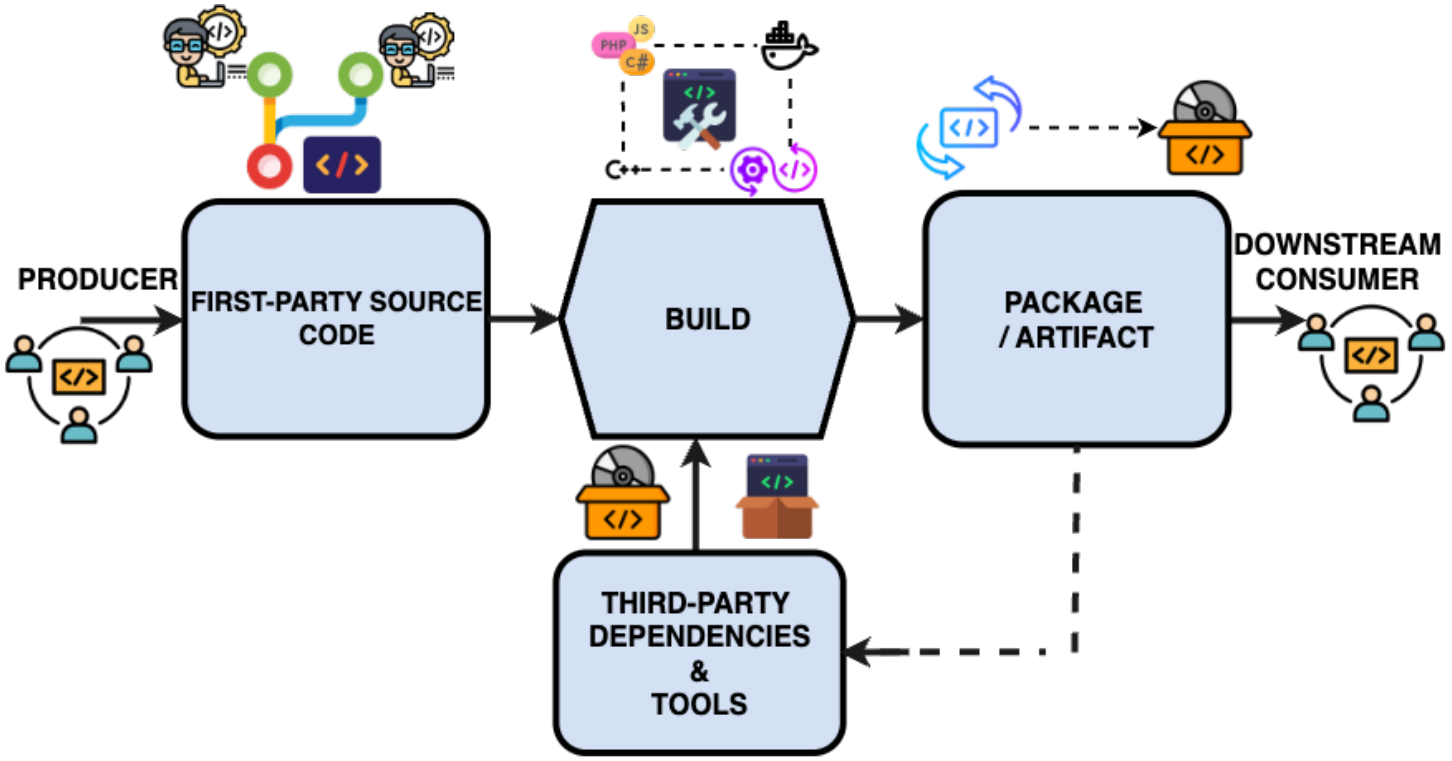


Results

Results (RQ1 – *Signing Implementation in Practice)*

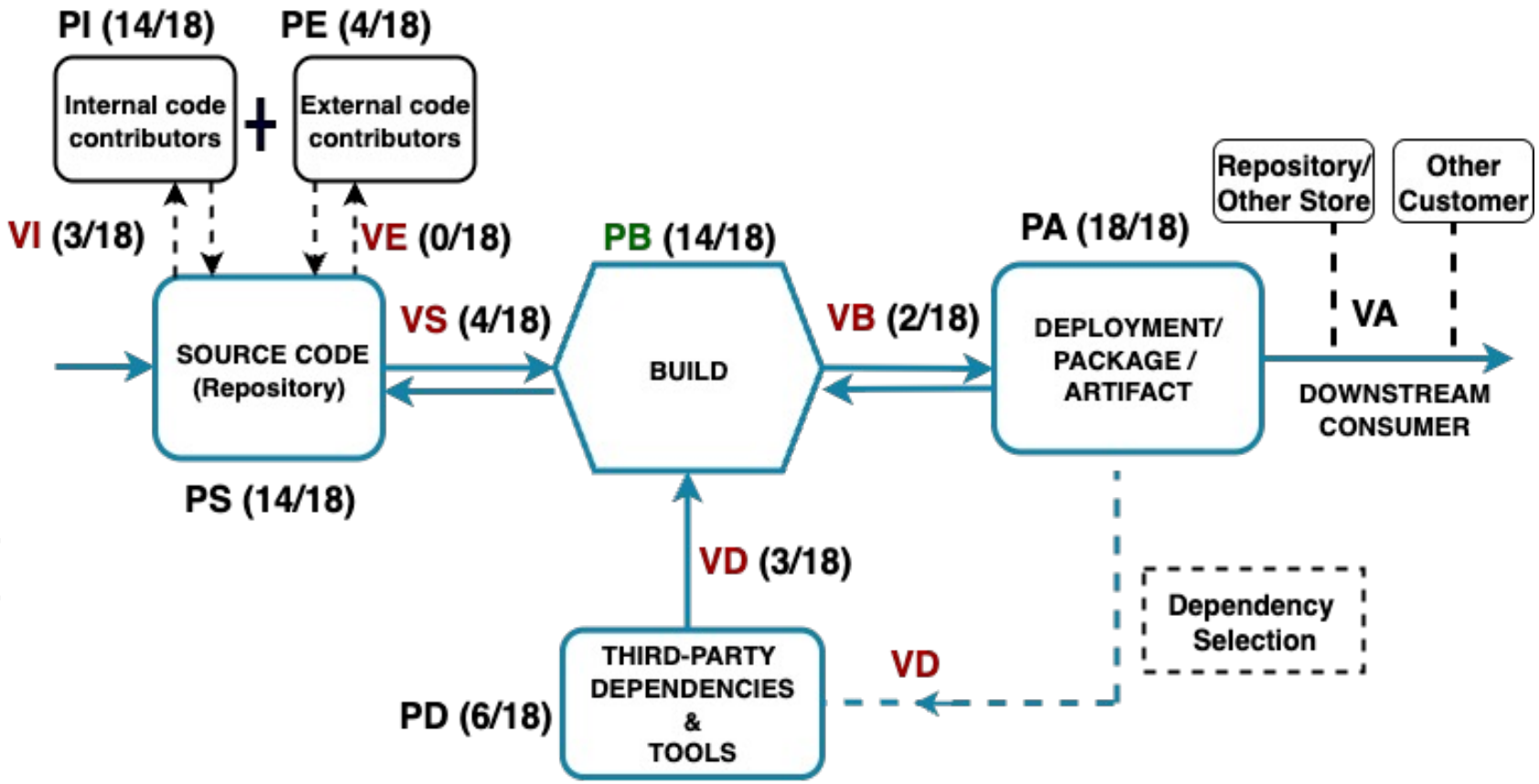


Results (RQ1 – *Signing Implementation in Practice)*



Results (RQ1 – *Signing Implementation in Practice)*

	Description
	SOURCE CODE
PI	Internal code contributors sign commits
PE	External code contributors sign commits
VI	Verify signatures from Internal contributors
VE	Verify signatures from external contributors
PS	Signing after code reviews/audits
	BUILD
VS	Verify source code signatures before build
PB	Signing of build output
	DEPLOYMENT/PACKAGE/ARTIFACT
VB	Verify build signature before deployment
PA	Signing of Final Software product
	THIRD-PARTY DEPENDENCY
PD	Signing after verification and certification of dependencies
VD	Verify signatures from external dependencies
VA	Verification by final customers



Results (RQ2 – *Challenges in software signing implementation*)

Observed Challenges	#Subjects	#Orgs	Subjects' Proposed Solutions
Technical			
Key Management	10	9	Use of Keyless Signing (*e.g.*, Sigstore)
Compatibility Issues	6	6	—
Lack of Verification of Signatures	6	5	Signed Metadata, Component Data Management
Ease of Use/Usability	4	4	Usable Signing Tools (*e.g.*, Sigstore), Documentation
No Unifying Standard	2	2	—
Organizational			
Operationalization of the Signing Process	4	4	Automating Signing
Resources to Set up Signing	3	3	—
Creating Effective Signing Policy	2	2	Regular Process Feedback Mechanisms
No Management Incentive to Sign	2	2	—
Bureaucracy	1	1	—
Human			
Expertise in setting setup and use of signing	5	4	—
Developer Attitude to Signing	3	3	Automating Signing
Lack of Demand from Customers	1	1	—

Results (RQ2 – *Challenges in software signing implementation*)

Observed Challenges	#Subjects	#Orgs	Subjects' Proposed Solutions
Technical			
Key Management	10	9	Use of Keyless Signing (*e.g.*, Sigstore)
Compatibility Issues	6	6	—
Lack of Verification of Signatures	6	5	Signed Metadata, Component Data Management
Ease of Use/Usability	4	4	Usable Signing Tools (*e.g.*, Sigstore), Documentation
No Unifying Standard	2	2	—
Organizational			
Operationalization of the Signing Process	4	4	Automating Signing
Resources to Set up Signing	3	3	—
Creating Effective Signing Policy	2	2	Regular Process Feedback Mechanisms
No Management Incentive to Sign	2	2	—
Bureaucracy	1	1	—
Human			
Expertise in setting setup and use of signing	5	4	—
Developer Attitude to Signing	3	3	Automating Signing
Lack of Demand from Customers	1	1	—

Results (RQ2 – *Challenges in software signing implementation*)

Observed Challenges	#Subjects	#Orgs	Subjects' Proposed Solutions
Technical			
Key Management	10	9	Use of Keyless Signing (*e.g.*, Sigstore)
Compatibility Issues	6	6	—
Lack of Verification of Signatures	6	5	Signed Metadata, Component Data Management
Ease of Use/Usability	4	4	Usable Signing Tools (*e.g.*, Sigstore), Documentation
No Unifying Standard	2	2	—
Organizational			
Operationalization of the Signing Process	4	4	Automating Signing
Resources to Set up Signing	3	3	—
Creating Effective Signing Policy	2	2	Regular Process Feedback Mechanisms
No Management Incentive to Sign	2	2	—
Bureaucracy	1	1	—
Human			
Expertise in setting setup and use of signing	5	4	—
Developer Attitude to Signing	3	3	Automating Signing
Lack of Demand from Customers	1	1	—

Results (RQ2 – *Challenges in software signing implementation*)

Observed Challenges	#Subjects	#Orgs	Subjects' Proposed Solutions
Technical			
Key Management	10	9	Use of Keyless Signing (*e.g.*, Sigstore)
Compatibility Issues	6	6	—
Lack of Verification of Signatures	6	5	Signed Metadata, Component Data Management
Ease of Use/Usability	4	4	Usable Signing Tools (*e.g.*, Sigstore), Documentation
No Unifying Standard	2	2	—
Organizational			
Operationalization of the Signing Process	4	4	Automating Signing
Resources to Set up Signing	3	3	—
Creating Effective Signing Policy	2	2	Regular Process Feedback Mechanisms
No Management Incentive to Sign	2	2	—
Bureaucracy	1	1	—
Human			
Expertise in setting setup and use of signing	5	4	—
Developer Attitude to Signing	3	3	Automating Signing
Lack of Demand from Customers	1	1	—

Results (RQ3 – *Perceived Importance of Software Signing*)

- **Subjects differed on this point.**

Results (RQ3 – *Perceived Importance of Software Signing*)

- **Subjects differed on this point.**

- **<u>As Crucial to Provenance and Integrity:</u>**
  - **S4 —**“I think signing is massively important. Signing for open source is probably the single most important thing for software supply chain security”.

Results (RQ3 – *Perceived Importance of Software Signing*)

- **Subjects differed on this point.**

- **As Crucial to Provenance and Integrity:**
  - **S4 —** "I think signing is massively important. Signing for open source is probably the single most important thing for software supply chain security".
- **<u>As a Secondary Security technique:</u>**
  - **S2 —** "I think the most important thing is accurate metadata collection and centralization. And even if that data's not signed, we can still get a lot of usefulness out of it. Signatures, in my opinion, they offer an additional layer of security on top of what we should be doing, which is metadata collection"

Results (RQ3 – *Perceived Importance of Software Signing*)

- **Subjects differed on this point.**

- **As Crucial to Provenance and Integrity:**
  - **S4 —** "I think signing is massively important. Signing for open source is probably the single most important thing for software supply chain security".
- **As a Secondary Security technique:**
  - **S2 —** "I think the most important thing is accurate metadata collection and centralization. And even if that data's not signed, we can still get a lot of usefulness out of it. Signatures, in my opinion, they offer an additional layer of security on top of what we should be doing, which is metadata collection"
- **<u>As a Requirement-Driven Practice:</u>**
  - **S8 —** "...for signed binaries and code signing ...a lot of those [requirements] are driven...by platform requirements...to prevent us from being blocked from...these platforms, code signing...was largely driven by [these platforms]."

Recommendations & Future Work

Recommendations

- Improving Signature & Artifacts verification in Signing workflows – *Improved Tooling.*
- Addressing Incentives challenges in software signing.

Recommendations & Future Work

Recommendations

- Improving Signature & Artifacts verification in Signing workflows – *Improved Tooling.*
- Addressing Incentives challenges in software signing.

- **Future Work**
  - What constitutes *"Improved Tooling"?*
  - *Research on operationalization of trust in the open-source space*

Thank you for your attention!

Our work on formative usability evaluation of signing tools here:

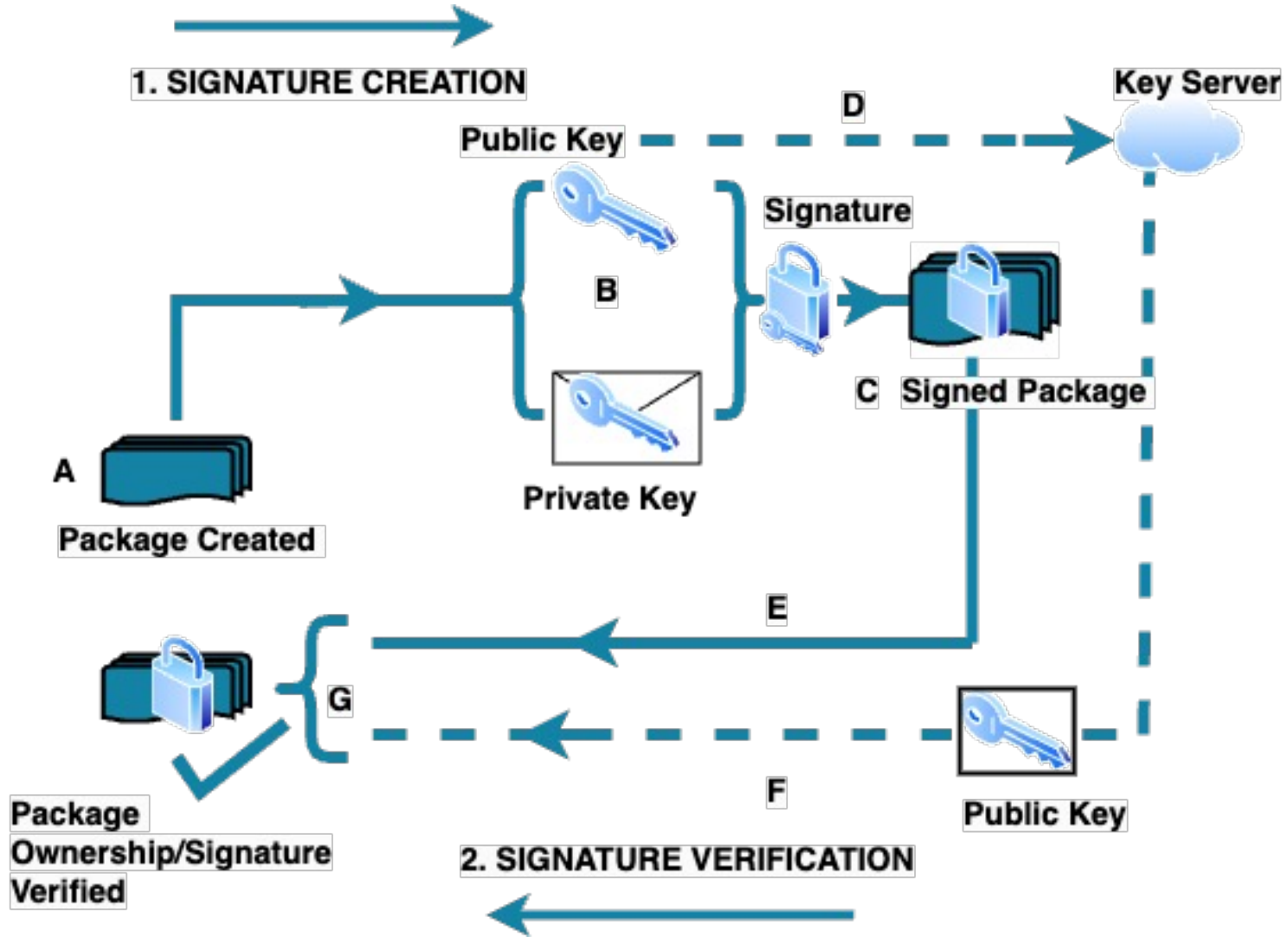
Kelechi G. Kalu
kalu@purdue.edu

Find our full paper here:

PURDUE UNIVERSITY

BONUS SLIDES

Software Signing Workflow



Interview Protocol

Topic (# Questions)	Sample Questions
A. Demographics (4)	What is your role in your team?
B. Perceived software supply chain risks (4)	Can you describe any specific software supply chain attacks (*e.g.*, incidents with 3^{rd}-party dependencies, code contributors, OSS) your team has encountered?
C. Mitigating risks with signing (8)	How does the team use software signing to protect its source code (what parts of the process is signing required)?
D. Signing tool adoption (5)	What factors did the team consider before adopting [TOOL/METHOD] over others?

Results (*RQ4 - Impact of Supply Chain Security Awareness on Software Signing Practices*)

Type	# Subjects (# Orgs)	# Direct fix	# Change in General Security process	# Change in Signing process
Vulnerability	9 (6)	9	2	1
Incident	5 (5)	5	2	0
Attack	4 (4)	4	2	0

- **Only one participant reports an influence of supply chain failures on the signing implementation.**
  - Software supply chain failures usually require immediate fixes to the malfunctioning or compromised components.
  - Large-scale failures typically trigger a comprehensive reassessment, leading to changes in the security strategy of a team or organization.

Results (*RQ4 - Impact of Supply Chain Security Awareness on Software Signing Practices*)

Type	# Subjects (# Orgs)	# Direct fix	# Change in General Security process	# Change in Signing process
Vulnerability	9 (6)	9	2	1
Incident	5 (5)	5	2	0
Attack	4 (4)	4	2	0

- **Only one participant reports an influence of supply chain failures on the signing implementation.**
  - Software supply chain failures usually require immediate fixes to the malfunctioning or compromised components.
  - Large-scale failures typically trigger a comprehensive reassessment, leading to changes in the security strategy of a team or organization.

- **Security regulations, frameworks, and standards does not generally drive the adoption of software signing.**
  - Only **2 participants** reports it influenced their adoption of signing.
  - **9 participants** opined that it influenced their adoption of other security techniques mainly SBOMs.