Updated Project Proposal outline

A) Exploring the Effectiveness and Implementation of Software Supply Chain Security Methods in Mitigating Risks: A Case Study of Software Signing and Sigstore

Research Questions

- RQ1a: What are the prevalent software supply chain risks faced by teams?
- RQ1b: How do the implemented methods, especially Software Signing and Sigstore, address or contribute to risk mitigation?
- RQ2: How do software teams implement Software Signing as a security measure to mitigate software supply chain risks?
- RQ3: What factors influence the selection of the adopted tool (Sigstore) over others in the context of mitigating software supply chain risks?

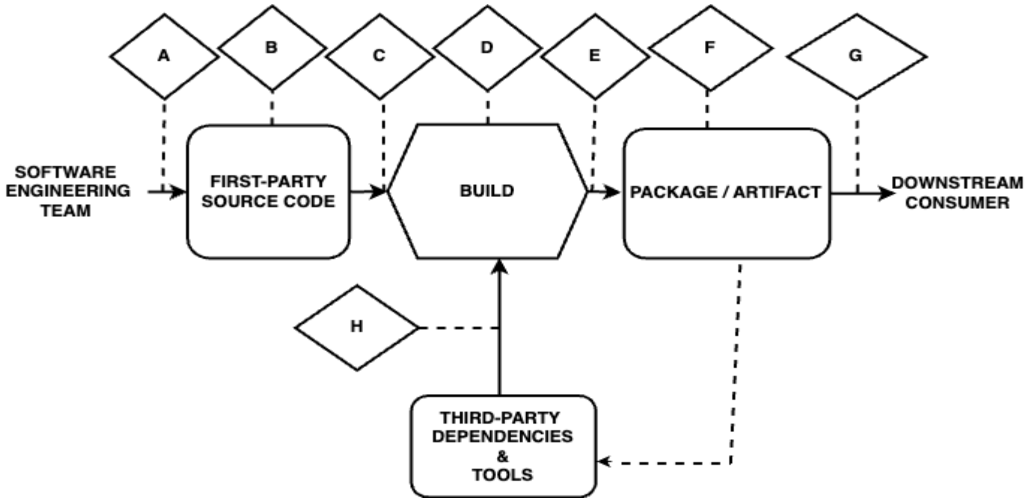
REFINED RQS/Research AIMS

RQ1 What are the perceived prevalent software supply chain risks faced in practice? – **perceived risk**

RQ2 What is the perceived importance of Software Signing in mitigating perceived risks? **Importance of Software Signing**

RQ3 How do software teams implement Software Signing as a security measure to mitigate software supply chain risks? – **software signing implementation**

RQ4 What factors influence the selection/adoption of specific signing tools over others in the context of mitigating software supply chain risks? – **Signing Implementation Adoption**



A. Demographic

1. What best describes your role in your team? (Security engineer, Infrastructure, software engineer, etc)
2. What is your seniority level? How many years of experience?
3. What is the team size?
4. What are the team's major software products/artifacts? **(Moved To Demographic from Prevalent SSC risk section After Pilot)**

B. Prevalent software supply chain risks faced by teams.

This set of questions aims to quantify how common certain risks exist in practice compared to others.

1. Describe briefly the team's process from project conception to product release and maintenance.
2. **(ADDED AFTER PILOT):** What do you consider a Software Supply chain attack/incident to be?
3. Can you describe any specific software supply chain risks (or incidences with third-party dependencies, code contributors, open source, etc.) that your team has encountered during your software development process?
    a. How were these addressed? (1B)
4. What are your team's greatest source of risk to the product?

   a. Project component
   b. Software Process

C. Software Supply Chain Risks and Software Signing in Mitigating them

This set of questions serves to understand with greater clarity how the team uses supply chain security methods/software signing to secure their source code, open-source code contributions and contributors, third-party dependencies, build scripts, deployed environments(containers), build infrastructure, and other infrastructure tools.

1. If no incident has been recorded (depending on the answer from Q1 in section I): why did the team choose to implement software signing?
   a. Is software signing the team's major strategy to secure its supply chain? Any other complimentary security efforts and methods?
   b. Are these strategies influenced by regulations, and standards?
   c. What challenges or obstacles did your team face while integrating Software Signing into your supply chain security practices?
   d. How do different team members contribute to the implementation of Software Signing? What roles and responsibilities are involved? **(After Practise)**
   e. Do you think Software Signing on its own merit (if implemented right) is good enough to completely secure a supply chain? **(Added after Pilot)**

THIRD-PARTY DEPENDENCIES:

2. What are the team's peculiar selection strategies for Third-party dependencies?
   a. How does the presence of a signature influence this decision?
   b. How is the authenticity of this signature verified?
   c. Any other methods/practices to ensure the trustworthiness of the third-party dependencies before integrating them into your projects?
   d. Can you describe these methods/practices?
3. What influences the team to discontinue the use of a package?

    a. How does the Signature on the dependency influence this decision?

4. How does the team manage its third-party dependencies' security?
    a. How do you maintain awareness of potential vulnerabilities or threats related to third-party components?

FIRST-PARTY SOURCE CODE

5. Is software signing a requirement for team developers (insider threats) and open-source contributors (if any)?
    a. how does the team use software signing to protect its source code (what parts of the process is signing required e.g. Commit signing)?

BUILD PROCESS

6. (How) Does the team utilize signing in its build process?

PACKAGE ARTIFACT

7. How is Signing used to protect the following from compromise?
    a. artifact's build binaries/ deployment pipeline
    b. artifact's repository

8. How does the team evaluate the effectiveness of their Security processes/Signing Implementations? Any Feedback mechanism

D. Tool/Sigstore Selection

This set of questions is meant to particularly extract information about how the team ensures the integrity of its product, with respect to the selection of Sigstore by the team.

1 What factors did the team consider before adopting this tool/method (Sigstore) over others?

2 What was the team's previous signing practice before the introduction of Sigstore?

3 How does your team implement Sigstore (which components of sigstore does the team mostly use)?

   a. Are there any areas where you believe further enhancements could be made in your current implementation of these methods?

4 Have you encountered any challenge(s) using this tool of choice? How have you coped with this limitation(s)?

5 Have you/your team considered switching this tool for another?

   a. What other tools have been considered or are currently being considered?

Change Timeline

After Practice Interviews:

   a. How do different team members contribute to the implementation of Software Signing? What roles and responsibilities are involved? **(Added After Practise)**
   b. What project component constitute greatest risk to ssc security

After Pilots (first 2 interviews):

   a. Do you think Software Signing on its own merit (if implemented right) is good enough to completely secure a supply chain? **(Added after Pilot)**
   b. What are the team's major software products/artifacts? **(Moved To Demographic from Prevalent SSC risk section After Pilot)**
   c. **What type of Organization/company – Removed from demographics**
   d. What are your team's greatest source of risk to the product? (Moved from signing to ssc risks)
      i. Project component
      ii. Software Process
   e. **(ADDED AFTER PILOT):** What do you consider a Software Supply chain attack/incident to be?

AT the 8Th interview

2. How does the team evaluate the effectiveness of their Security processes/Signing Implementations? Any Feedback mechanism