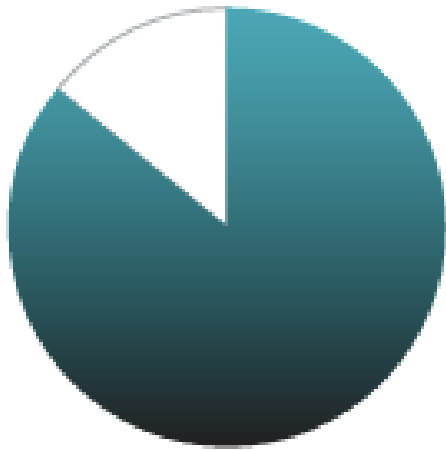


How Does Security Practice Adoption Impact Security Outcome Metrics Over Time?

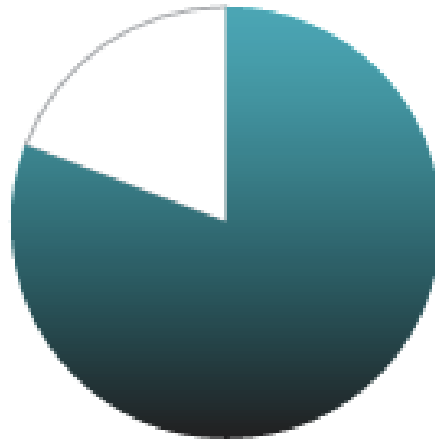
Jill Marley

Advisor: Dr. Laurie Williams

Vulnerable Dependencies



86% of
codebases
contained
vulnerable
open source
dependencies



81% of
codebases
contained
high- or
critical-risk
vulnerabilities

Software Security Frameworks



Secure Supply Chain Consumption Framework (S2C2F)

Microsoft continues improving the Framework in partnership and collaboration with the OpenSSF.



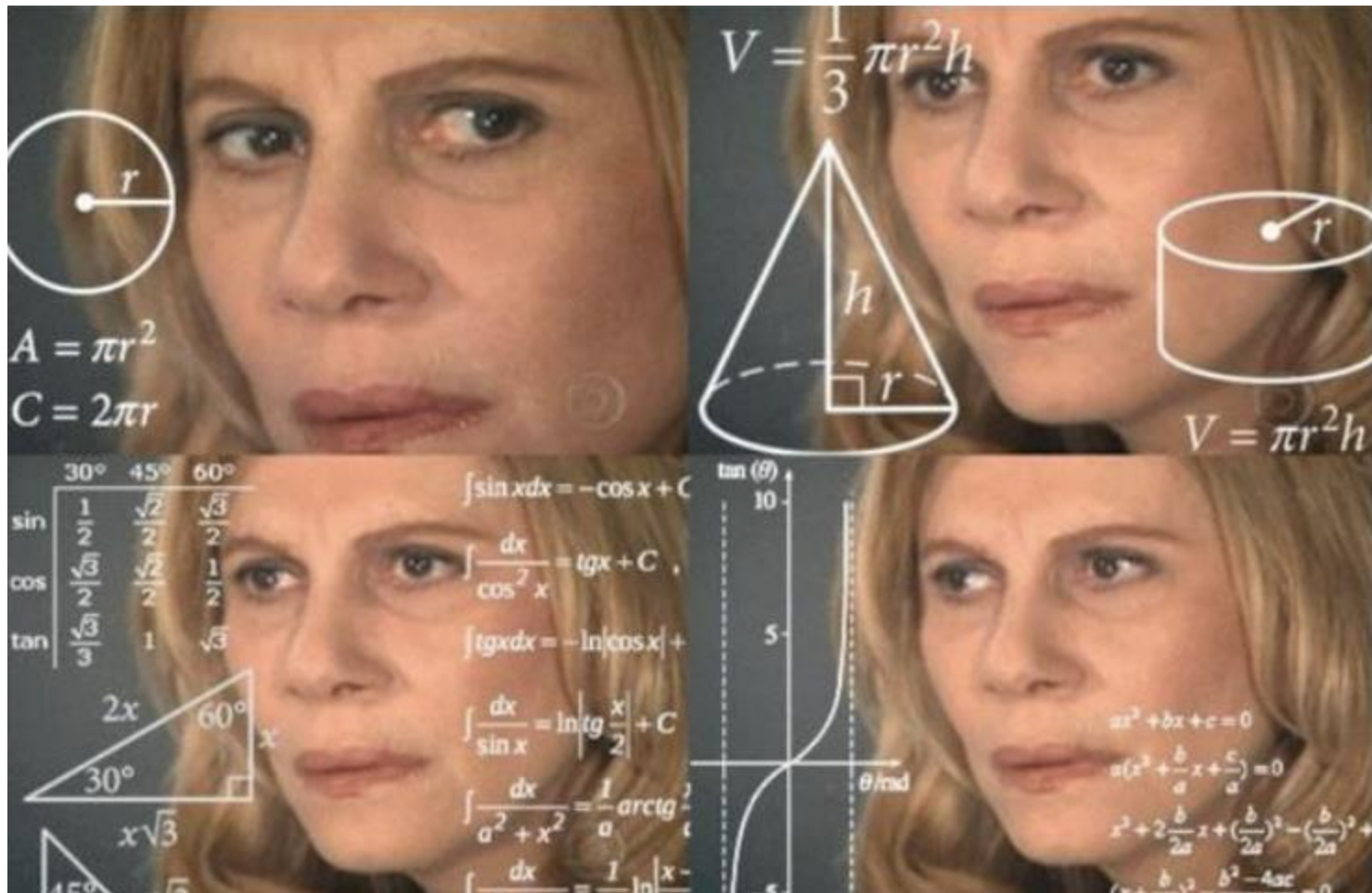
Secure Software Development Framework (SSDF)



Scorecard

Proactive Software Supply Chain Risk Management Framework

More security practices, more vulnerabilities?



Longitudinal study

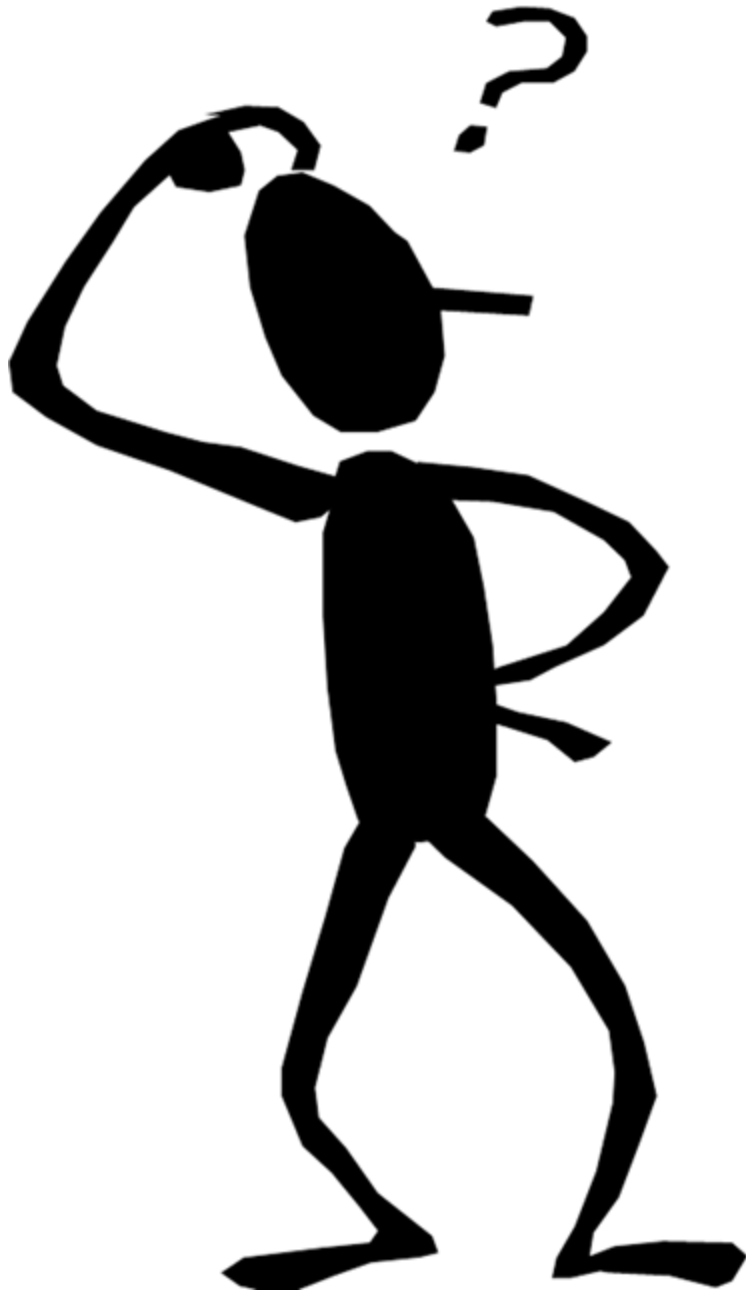


The diagram features a large, light red arrow that curves upwards from the bottom left towards the top right. Along the path of the arrow, there are three red circular markers. Below each marker is a year in large, bold, black font: 2000, 2015, and 2025. The text 'Longitudinal study' is written in a black, sans-serif font, following the curve of the arrow's path.

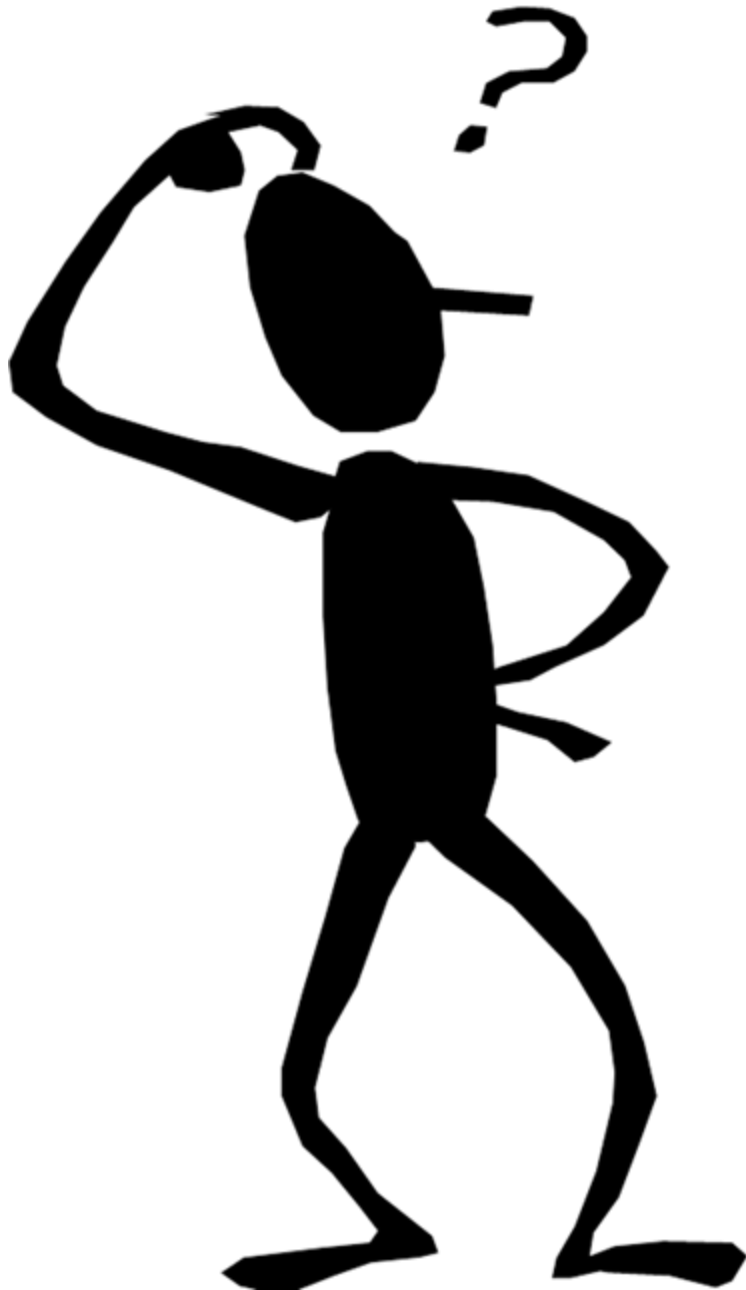
2000

2015

2025

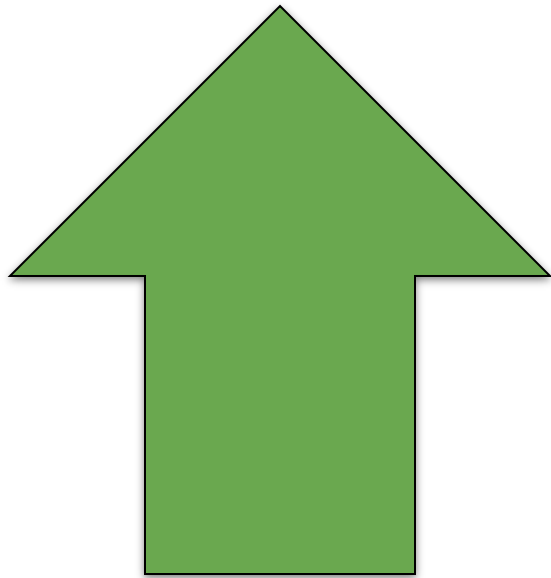


When software practitioners adopt security practices, do their security outcome metrics improve over time?



More specifically,
we want to
investigate if
adopting security
practices *causes*
security outcome
metrics to
improve through
causal analysis.

Hypothesis



Increase in
Practice Adoption

leads to



Better
Security Outcomes

Goal: Empirical Evidence



Measuring Practice Adoption: OpenSSF Scorecard



Measuring Practice Adoption: OpenSSF Scorecard



Examples of Scorecard Metrics

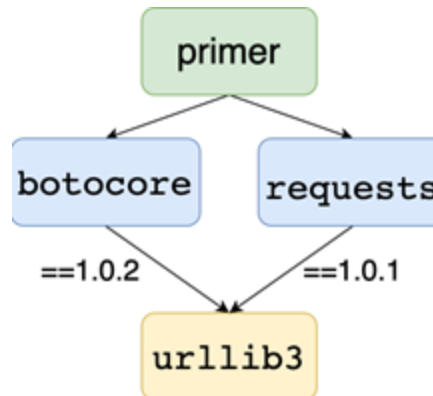


Code Review

Contributors



Contributors



Pinned Dependencies

Security Outcome Metrics

1. Vulnerability Count

The number of publicly disclosed or reported vulnerabilities within a project

Security Outcome Metrics

2. Mean Time To Update (MTTU)

The average aggregated time a package uses outdated direct dependencies in its lifetime

Security Outcome Metrics

3. Mean Time To Remediate (MTTR)

The average aggregated time a package uses outdated and vulnerable direct dependencies in its lifetime.

Dataset: Inclusion and Exclusion Criteria

Including :



- Packages with at least one dependent and dependency
- At least 2 years old

Excluding :

- Repositories that contain multiple packages

GitHub Release

Releases / Tags

Latest release

v0.17.2

def84ab

0.17.2

 **spraints** released this a month ago

- Use the git author as the TFS committer during `git tfs rcheckin` (#336) and `git tfs rcheckin --quick` (#357)
- Improve temporary workspace handling (#328, #372)
- Use libgit2sharp more and git-core less (#361)
- Bug fix for bare repositories (#352)
- Bug fix for crash during `git tfs clone` (#349)
- Bug fix for VS2008 (#362)
- Update libgit2sharp
- Improved release process (#333, #340)

[Full diff](#)

⬇ gltfs-0.17.2.zip

📄 Source code (zip)

📄 Source code (tar.gz)



Dataset Overview

- ~14k packages
- October 2023 - October 2025
- For each package release:
 - Scorecard metrics
 - Security outcome metrics
 - Control variables

What are the trends we can identify in Scorecard score changes?

How do changes in individual Scorecard metrics over time affect vulnerability count, MTTU, and MTTR?

Given practice adoption, how long does it take for the security outcome metric to improve, after controlling for (our control variables)?

Thank you!

Any thoughts? Contact me:

Jill Marley

jahmad5@ncsu.edu

