UNIVERSITY OF NORTH CAROLINA
CHARLOTTE

Vulnerability Perceptions and Practices in Software Development Teams

Arpita Ghosh, Lipsa Sahoo, Heather Lipford
University of North Carolina at Charlotte

Motivation

Ensuring robust security in software development is crucial due to high risks from vulnerabilities. Despite recommendations for greater focus on security, many organizations lack robust practices. This study examines teams with standard vulnerability management processes to identify common practices, and variations, aiming to improve prevention and mitigation strategies.

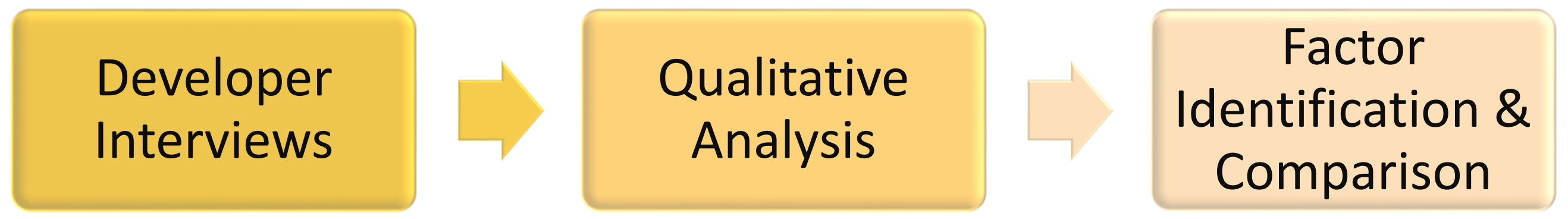
Research Goals

Beyond implementing a secure development life cycle process, what other factors can organizations address to improve vulnerability prevention and mitigation?

We are investigating the following within development teams:

- Team structure and roles
- Team perceptions and attitudes
- Adherence to standard procedures
- Perceptions and incentives of the management team
- Developer training and experience

Methodology


Developer Interviews → Qualitative Analysis → Factor Identification & Comparison

Recruitment: 35 software developers from enterprise organizations primarily in financial sector

Interview Topics:

- Vulnerability detection processes of the individual and team
- Threat modeling and perceptions
- Individual perceptions of security
- Perceptions of the team's and management security performance
- Security training of themselves and the team
- Desires and motivation for vulnerability support

Qualitative Themes

- **SDLC Practices**
  - Common: Regular static analysis, primarily limited to code check-in
  - Common: Testing focused on functionality, little security testing
  - Distinctive: A few teams utilized a variety of tools for continuous testing

> *"We use Splunk to check logs... looking at Splunk logs to see if there's any type of vulnerability" - P25*

- **Code Review and Management**
  - Common: Peer code reviews required, but little security focus
  - Distinctive: Teams follow coding standards, and use code review checklists to ensure consistent security focus

> *"We do peer review even before we dive into the development. To start the developer test, we do the peer review, and ensure that the code is doing what it is supposed to do. And it is not leaking anything like, you know, sensitive data, or it is not going to cause an issue to the existing app…We, as leads at different levels, look at that, and we make sure that the security is validated." – P11*

- **Vulnerability and Threat Management**
  - Common: Scanning reports are lengthy and filled with false positives, making it difficult for teams to prioritize which vulnerabilities to mitigate
  - Distinctive: Some teams designate individuals to manage these reports and the response to detected vulnerabilities

> *" Well, somebody is tasked to oversee everything especially related to vulnerability, may be scanning reports. Then the word goes out to everyone, check all the code, and everybody scanned the code for smaller vulnerabilities." - P31*

- **Security Training**
  - Common: Organizational generic security training required regularly
  - Common: Training theoretical, not hands-on
  - Distinctive: A few teams focused on additional training and knowledge transfer (KT), particularly for new team members

> *"So there are some training about the attacks on the code. Like whatever the spammers do, whatever had attackers do, how to avoid that. Such type of training we have usually in after 2 to 3 months." - P34*

- **Team Dynamics**
  - Common: Teams with stable compositions, strong interpersonal relationships, and active managerial involvement lead to stable security practices
  - Contrasting: High turnover and disengaged management lead to fragmented and inconsistent security practices
  - Distinctive: Manager emphasis on security critical for adherence

> *"He [manager] wanted it to be secure... he was very clear that he did not expect us to find vulnerabilities when we went into production.." - P31*

- **Incentives**
  - Common: Functionality and speed rewarded, little motivation for secure practices
  - Distinctive: Team recognition for security fostered a security-focused culture

> *"getting a shout-out by doing a good job in preventing any vulnerability in the group scrum call" – P33*

Takeaways

- Effective software security extends beyond just technical measures and tools, necessitating a holistic approach that includes rigorous SDLC practices, team dynamics, and continuous security training.
- Teams with security-oriented cultures reported heavier focus on training, structured peer reviews, and organized responses to vulnerabilities.
- Effective security is linked to aligning team incentives and managerial support with security objectives.

Future Work

- Additional interviews to diversify organizations and teams
- Complete analysis, including exploring relationships between codes and themes
- Develop guidance for organizations to improve practices
- Develop survey to help organizations understand and track team practices over time

This research was funded through the NSF IUCRC Center for Cybersecurity Analytics and Automation