Security Risk
Assessment and Management as
Technical Debt

Kalle Rindell

kakrind@utu.fi

UNIVERSITY OF TURKU

Johannes Holvitie

johannes.holvitie@tyks.fi

TYKS Turku University Hospital

Index

1. Software Security
   - What, when, who, why
2. Technical Debt and Security Risk
   - Concepts and management
3. Security Debt
   - Types, consequences
4. Conclusions and future work

Software security

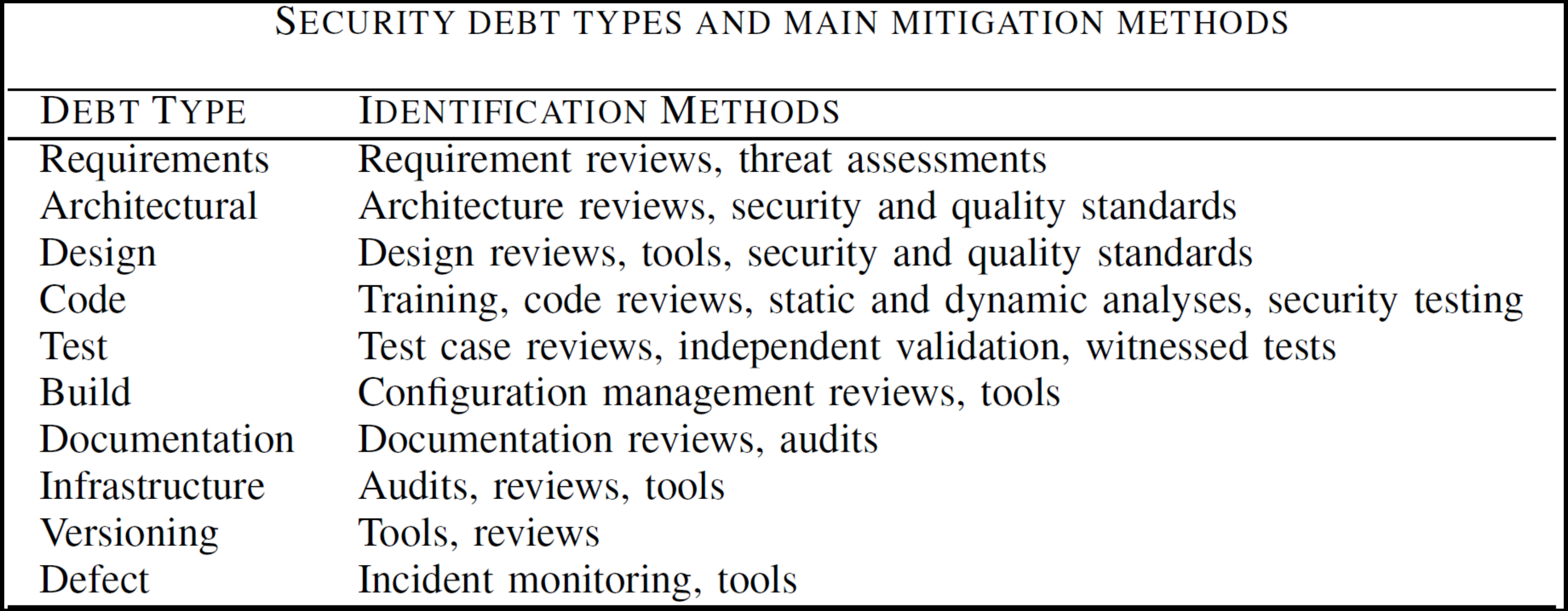
- Information security → application security → software security
- Development-time activities to ensure the following:
  a) software security requirements are adequate
  b) software meets the requirements
  c) sufficient security assurance is produced
- Secure design and coding practices
- Security functionality and features (e.g. authentication, encryption)
- Security testing: validation and verification of the above

Technical debt

- TD captures trade-offs between development-driving aspects (e.g. release time vs code structuring)
- (In)voluntary development-time decisions affecting software's internal quality
- Reduced internal quality may yield short-term benefits, but it can slow and potentially stop future development
- **Principal**: resources required for refactoring or redesign
- **Interest**: future development needs to adhere to lowered internal quality
- **Risk**: in literature, only direct labor costs and business risk
- How does security risk change the appraisal of TD?
- How can TD management help managing the security risk?

Security Debt; Security risk sensitive TD

- Notable challenges in TD management include identification, management, and tools for the previous
- Security engineering has the tools and techniques, but has issues in management: measurement/appraisal is a central problem



SECURITY DEBT TYPES AND MAIN MITIGATION METHODS

DEBT TYPE	IDENTIFICATION METHODS
Requirements	Requirement reviews, threat assessments
Architectural	Architecture reviews, security and quality standards
Design	Design reviews, tools, security and quality standards
Code	Training, code reviews, static and dynamic analyses, security testing
Test	Test case reviews, independent validation, witnessed tests
Build	Configuration management reviews, tools
Documentation	Documentation reviews, audits
Infrastructure	Audits, reviews, tools
Versioning	Tools, reviews
Defect	Incident monitoring, tools

Theoretical model and benefits

- TD prominently borrows from economics: the portfolio approach (Markovitz, 1952) applied to TD by Guo (2011)
- A portfolio is a list of TD items: extended by including security risk
- TD item in a SW artefact (depending on debt type) may consist of
  - Identifying information (ID, location)
  - Work amount estimate (LoC, complexity, velocity-dependent time estimate)
  - Changes to related components (code, design, tests, documentation…)
  - Security risk (affected process/asset, impact, estimated probability)
- Increase the visibility and manageability of security risk, decrease the overall security risk

Key issues

- **Prioritization**: after identifying and making the estimates, how does a TD item get placed into the work queue? How is the queue updated?
- **Compound effect**: big security risks get typically immediate attention. How about the small ones? How do they add up over TD items?
- **Interest**: TD has direct cost-related consequences, but its payback can typically be planned ahead. Security flaw or bug (i.e. external quality) is intentionally triggered, and may lay dormant indefinitely.
- **Collateral damage**: amount of work not only to repair the SW but to deal with the spillover and indirect effects.

Conclusion

- Identification and repayment are "trivial" – assessment is not
- Tool support needed – e.g. SonarQube
- Security risk is a theoretical concept with concrete consequences
    - Probability: a guess based on current security information.
    - Impact: another guess based on business projections.
    - Work estimate: an educated guess based on whichever model is used (possibly input with historical data).
- Like for any conceptual model, *empirical data* is required
    - Domain, technology and organization – down to team and role level?

Thank you.

Kalle Rindell

kakrind@utu.fi

UNIVERSITY
OF TURKU

Johannes Holvitie

johannes.holvitie@tyks.fi

TYKS
Turku
University
Hospital