The Economics of Secure Software Development

2019 Annual Research Review

Elaine Venson

March, 2019
Secure Software Development Life Cycle (S-SLDC)

Software Security Touchpoints

Costs of Software Security

• Practices to address software security have been increasingly introduced in the software development lifecycle.
• However, current models for software effort estimates do not consider the security factor, causing inaccurate estimates results, which affects the planning of required resources for software project development.
Systematic Mapping

- Classify and analyze literature related to the impact of security in software development costs.
  - Categories/themes of papers
  - Major sources of cost
  - Existing cost estimation models for secure sw development
  - Security standards and security engineering process used
• IC1 – Study about software security that considers effort/cost impacts.
• IC2 – Study about effort/cost estimation or measurement that considers software security issues.
Categories of Papers

- Economics of SecSW development: 9
- Countermeasure identification/prioritization: 9
- Vulnerability prediction/detection: 9
- Software cost estimation model: 7
- Security in agile development: 6
- Security development method: 4
- Security effort analysis: 4
- Quantification of software security: 2
- Security practices in software development: 2
- Security requirements identification/prioritization: 2
Economics of SWSec Development

- Cost issues are still a barrier to the introduction of security practices in software projects.
- Papers that analyze software security in the context of business, management and finances:
  - Effectiveness of applying security practices early in the SDLC
  - Planning for software security
  - Secure software engineering investment models
Effectiveness of SWSec

• Most security problems are found during testing or in the field.
  – Pattern: create -> discover -> remove vulnerabilities
• Race between attacker (find faults -> easy) and defender (fix faults -> hard).
• Affordable secure software -> SW built with few vulnerabilities.
• Security can be improved and total cost of a software can be reduced by appropriate allocation of resources to the early stages.
  – Vulnerabilities found and fixed as close as possible to their point of introduction.
Effectiveness of SWSec

• Does early effort invested in security contribute to the success of software?
• Chehrazi et al. (2016) analyzed OSS projects:
  – Security in early stages positively related to success
  – Late consideration of security negatively related to success
Planning for SWSec

- In Agile Development, abuser stories with expected cost, instead of business value.
  - Cost = loss x probability of attack.
  - Abuser stories related to User stories (many-to-many).
  - Planning needs to optimize net value (adding functionality can enable new attacks).
  - Users stories may have negative value.
Planning for SWSec

- Integration of economic aspects as attributes of misuse cases/abuse cases.
- Quantifiable basis to link business and engineering practices.

SWSec Investment Models

- Financial benefits (relative cost savings and ROSI) to elicit support for secure development initiatives.
- Risk analysis weight up-front costs of protection for vulnerabilities.
- Value of Protection:

\[ \text{VoP} = (P(V) \times L) - M \]

- P(V): prob. successful attack on vulnerability V
- L: loss resulting from the attack
- M: mitigation countermeasure

SWSec Investment Models

• Information Security investment models:
  – Enterprise level.
  – Help to determine the optimal amount to invest to protect a given set of information.
  – Focus on post-development optimization of reactionary approaches to security.

• Secure Software Engineering (SSE) investment models:
  – Understand SSE investment across a system’s life cycle in order to maximize gains.
The Iterated Weakest Link (IWL) Model

- Investment model from Information Security field.
- Dynamic and adaptive investments over time.
- Process of attack and defense of successive weakest links.
- Uncertainty about the weakest component.
- The ordering of the threats is known by the attacker and unknown to the defender:
  - This misalignment is capture as $\sigma$.
- Each successive threat ($1, \ldots, n$) has an increase in costs of $\Delta x$.

(a) Attack profile with certainty ($\sigma = 0$).
(b) Attack profile with uncertainty ($\sigma > 0$).
The Iterated Weakest Link (IWL) Model

- The defender starts by specifying an initial defensive configuration.
  - 1, ..., n expected threats
  - 1, ..., k defenses, with k <= n
- Attacker may choose to exploit the component with least true cost, not covered by the defensive configuration (k+1).
  - It happens when the benefits exceed the cost of the attack.
- Defender can:
  - Ignore the attack and absorb the loss;
  - Disinvest, business is non-viable;
  - Specify an new defensive configuration.
- Repeat until secure state is reached, when the threats below the reservation cost of attack are protected.
The goal of secure software development is in having fewer economically vulnerabilities (by increasing attack costs) with less uncertainty (nature of the vulnerabilities present).

IWL-SSE
Defender Investment

\[ I_{\{AD,IT\}} = (i \cdot c) + i(\text{eff} \cdot e), \]

- \( I \): cost of secure software engineering activity conducted at time \( t \).
- \( i \): set of iterations within that activity.
- \( c \): cost of the activity.
- \( \text{eff} \): effectiveness of the activity:
  - the probability of finding a flaw via review (\( \alpha \)),
  - or the probability of finding a bug or flaw via test (\( \beta \)).
- \( e \): cost to fix the flaw or bug.
IWL-SSE
Defender Payoff

• Reduction in the overall uncertainty faced by the defender:
  – Overall uncertainty is given by:
    • $\sigma = \sigma_{AD} + \sigma_{IT}$
    • $\sigma_t = \frac{\sigma_{max}}{2} \cdot \frac{1}{eff_i}$
      $eff \in \{\alpha, \beta\}$
      $i \in \{i_{AD}, i_{IT}\}$
      $t \in \{AD, IT\}$
• Increase of the gradient of attack:
  $\Delta x = \sqrt{(1 + \alpha i_{AD} + \beta i_{IT})}$
Return on Secure Software Process (ROSSP)

• Return on Security Investment (ROSI)

\[
ROSI = \frac{ALE_0 - ALE_1 - \text{average security investment}}{\text{average security investment}}
\]

– ALE = Expected rate of loss * Value of loss
– ALE_0 is the ALE without security investment
– ALE_1 is the ALE with security investment

• \( ROSSP = ROSI_{SSE} - ROSI_{NOSSE} \)
**IWL**


---

**IWL-SSE**

Conclusion

- Allocation of security effort over the SDLC impacts on software costs.
- Balancing between features and security as an optimization problem.
- No investment in security may be a rational choice.
- Intensity and effectiveness of security practices affects security level (benefits) and costs.
- Investment models can guide the amount and allocation of security expenditure.
- Software security investments in the context of information security investments.
References


