Software Security Readiness and Deployment

Saikath Bhattacharya, Munindar P. Singh, Laurie Williams
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206, USA
Email: sbhatt26@ncsu.edu, mpsingh@ncsu.edu, lawilli3@ncsu.edu

***Abstract*—Software companies need to assess the security of their products before release. Software reliability engineering tracks and predict software failures using statistical models and metrics. The goal of this paper is to aid software companies in tracking software security growth and identify release readiness through software reliability engineering. Enhancement of software reliability engineering concepts, metrics, and techniques to software security can yield a fresh perspective to secure software release readiness and deployment process.**

***Index Terms*—Software Security, Software Reliability, Release Readiness, Vulnerability Discovery.**

I. Introduction

Software security is an important concern because vulnerabilities in software could lead to cyber-attacks, violate user privacy, or cause economic harm. Recent supply chain attacks, such as the Solarwinds cyber-attack [1], have prompted President's executive order on criteria to evaluate software security [2]. To mitigate cyber-attacks, companies need to assess the security of their products before release. Software companies tracks software test data and identify its release readiness based on operational performance indicators. However, software release is subjective based upon software companies, stakeholders, software vendors and customer business needs. An early or late release may yield negative production cost, return of investment, customer satisfaction, and overall product success.

Software security growth tracks the increase in software security by extrapolating the decrease in vulnerabilities during testing. Software security readiness considers the security growth and identifies optimal release time. To analyze software security, companies often utilize specialized software security testing tools and allocate additional testing time, thereby delaying software release. Providing a secure software release within a stipulated time is a challenge for developers, managers, and software testers.

Software reliability engineering (SRE) assesses software product, tracks software failures, and estimate operational service requirements such as software availability. Based on SRE techniques, organizations predict the number of failures in software and allocate necessary resources to mitigate them. SRE incorporates metrics, statistical models, and techniques to aid organizations in tracking and achieving the desired degree of reliability and availability in a software system [3].

The paper discusses translation of SRE concepts into software security growth models for secure software release. This work will answer the following research questions:

- **RQ1**: How does software reliability growth models (SRGM) maps to software security and software security growth?
- **RQ2**: What metrics effectively measure software security growth, ensuring secure software readiness before its release?

The rest of the paper discusses the development of the research questions (RQs) and techniques to incorporate SRGM for software security engineering (SSE) and security growth models.

II. Integrating software reliability with software security engineering

Software reliability growth models (SRGMs) predict software failures and analyze software reliability growth during testing [3]. SRGM considers a non-homogeneous Poisson arrival rate for software failures. Researchers [4] have studied non-homogeneous Poisson process(NHPP) based SRGMs and their applications to optimal release policies considering software failure intensity. Pham et al. [5] considered the effect of imperfect debugging on optimal software release time. Nagaraju et al. [6] developed an open-source software reliability tool with six SRGM models in their closed forms. Researchers have applied SRGM concepts to predict software security and vulnerability as discussed in section III.

Through our research, we will develop security growth model based on SRGM. Open source software product bug reports will be analyzed for software failure, vulnerability and severity. To develop and evaluate the translation of SRE concepts into security objectives and deployment readiness, we consider the following steps:

1) *Identify the rate of vulnerabilities discovered over software testing and post release.*
   Software bug reports include (i) a summary description of the bug; (ii) a unique bug ID (iii); timestamp information about open and closed bug report; (iv) bug severity; (v) software version; and (vi) type of bug: ”security” or ”non-security”. By capturing the timestamp and bug severity over time, the expected rate of vulnerabilities discovered along with its severity could be calculated to develop the software security growth function.
   Software security growth models will be developed to predict and meet product security and deployment readiness objectives. Meta-heuristic methods will be used

to estimate model parameters in a stable and efficient manner.

2) *Determine security metrics to prioritize testing efforts.* The presence or absence of a vulnerability patch date can infer the status of software security growth. Similarly, identifying vulnerability severity and priority could help to determine optimal resource allocation and testing efforts. Metrics such as arrival rate of vulnerabilities and mean-time-to-recover will be tracked to identify malicious, accidental misuses, and trace compromised event timeline.

III. Software security growth models

Vulnerabilities are special class of software defects which could be exploited for malicious misuse by hackers. Statistically, software vulnerabilities are stochastic counting processes similar to software failures experienced during testing phase. Based on the counting process, software security could be expressed as a probability that no exploit occurs during a certain time interval. Statistical security growth models could lead software engineers to predict product security and determine its release readiness.

Researchers [7], [8] have considered extending software reliability modeling approaches to security. Similarly, software reliability growth has been used for vulnerabilities discovery models (VDM) [9]. Johnston et al. [8] worked on integrating vulnerability discovery within the software security lifecycle. Often, the vulnerability discoveries across data have no discernible patterns even though products have similar features [7]. Software developers and product managers track different time-dependent vulnerability variables, such as vulnerability report time, resolved time, and patch release time which may influence security prediction differently.

Traditional software reliability models limit consideration of the reliability metrics to software testing efforts only. Whereas vulnerability discovery models considers post-release security data and apply SRGM concepts to future software versions. However, vulnerability data are sparse and is a subset of all undiscovered vulnerabilities, resulting in overestimation of the vulnerability prediction in the next software release.

To overcome this problem, *we will consider Bayesian analyzes generalized over non-homogeneous Poisson process for software security growth model.* Bayesian non-homogeneous SRGM could encapsulate time-dependent software security testing variables, identify secured release readiness checkpoints, and update deployment policies.

IV. Discussion and Conclusion

The proposed research integrates reliability modeling with software security using empirical vulnerability data to improve software and release readiness. The research will consider software reliability growth models and metrics such as arrival rate of vulnerabilities as the basis for software security growth models. Based on the software security growth models and software security engineering, secure release readiness and deployment policies will be determined.

The research will consist of the following research steps:

1) Identify security metrics such as the arrival rate of vulnerability and mean-time-to-recover from open-source software bug reports and vulnerability database.
2) Develop software security growth model using Bayesian inference considering time-dependent vulnerability parameters and NHPP software reliability growth model.
3) Develop multi-point software release policy incorporating security metrics.
4) Perform tradeoff analysis and identify optimal resource allocation between team efforts, resilience strategies, and critical security components.

Software security engineering processes will assist software engineers in tracking security testing, testing efforts, and secured software release readiness. Similarly, SSE could guide the validation and verification (V&V) efforts toward those parts of the software that influence security and resilience the most.

However, limitations of incorporating SRGM includes solving computationally expensive higher-order complex mathematical forms which might not capture the actual testing and development process. Complex SRGM models could also mislead the program managers to misinterpret information. Also, during software testing covariates such as effort-based reporting and software change-points are not recorded which might influence effort-based SRGMs. Similarly, the rate of vulnerability discovered depends upon testing efforts and the skills of those exerting it.

Acknowledgment

We thank the NSA's Science of Security Lablet at NC State University for funding the research.

References

[1] "These big firms and US agencies all use software from the company breached in a massive hack being blamed on Russia." [Online]. Available: https://www.businessinsider.in/tech/news/these-big-firms-and-us-agencies-all-use-software-from-the-company-breached-in-a-massive-hack-being-blamed-on-russia/articleshow/79725169.cms

[2] "Improving the Nation's Cybersecurity," May 2021. [Online]. Available: https://www.govinfo.gov/content/pkg/FR-2021-05-17/pdf/2021-10460.pdf

[3] M. R. Lyu, *Handbook of software reliability engineering*. McGraw-Hill, Inc., 1996.

[4] J. D. Musa and K. Okumoto, *Application of basic and logarithmic poisson execution time models in software reliability measurement*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988, pp. 68–100.

[5] H. Pham, L. Nordmann, and Z. Zhang, "A general imperfect-software-debugging model with s-shaped fault-detection rate," *IEEE Transactions on Reliability*, vol. 48, no. 2, pp. 169–175, 1999.

[6] V. Nagaraju, V. Shekar, J. Steakelum, M. Luperon, Y. Shi, and L. Fiondella, "Practical software reliability engineering with the software failure and reliability assessment tool (sfrat)," *SoftwareX*, vol. 10, pp. 1–6, 2019.

[7] A. Ozment, "Software security growth modeling: Examining vulnerabilities with reliability growth models," in *Quality of Protection*. Springer, 2006, pp. 25–36.

[8] R. Johnston, S. Sarkani, T. Mazzuchi, T. Holzer, and T. Eveleigh, "Multivariate models using mcmcbayes for web-browser vulnerability discovery," *Reliability Engineering & System Safety*, vol. 176, pp. 52–61, 2018.

[9] A. Omar, M. Yashwant, and R. Indrajit, "Measuring, analyzing and predicting security vulnerabilities in software systems," *Computers & Security*, vol. 26, no. 3, pp. 219–228, 2007.