Applying Software Security Growth Model to Web-Browser Software

Saikath Bhattacharya, Munindar Singh, Laurie Williams
Department of Computer Science, North Carolina State University

Motivation: Program managers need to track the software security growth and identify if the product has mitigated enough security vulnerabilities before its release.

Research Question 1: Analyzing if software reliability growth models(SRGM) can be measure software security growth?
Research Question 2: What SRGM metrics effectively measure software security growth?

Software Reliability Engineering

- Software failure function $m(t)$ having model parameters $\{a, b, c\}$ indicates the expected numbers of the unique vulnerabilities detected during testing time interval (0, t].

Table 1: SRGM Models

Model	Software Failure Function $(m(t))$
Inflection S-Shape Model	$m(t) = \frac{a(1-e^{(-bt)})}{1+ce^{-bt}}$
Delayed S-Shaped Model	$m(t) = a\left(1-(1+bt)e^{-bt}\right)$
Goel-Okumoto Model	$m(t) = (1-e^{-bt})$

Case Study- Google Chromium Vulnerabilities

- We consider the bug-Security issues to be the vulnerabilities reported in Chromium Monorail.
- A total of 1986 security vulnerabilities were reported from 1st Jan. 2021 to 31st Dec. 2021.

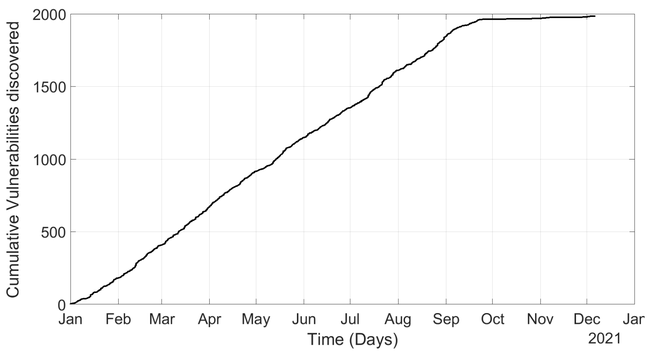

Fig 1: One-year cumulative vulnerability

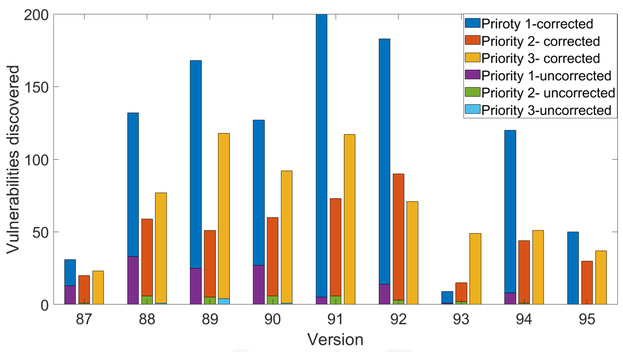

Fig 2: Bar graph of vulnerabilities discovered over various versions

Software Security Growth Model

- Based on Table 1, software security is $S(t_d) = e^{(-(m(t_n+t_d)-m(t_n))}$
- Optimal release time $t^\star$ is estimated as $S\ (t_d^\star) = e^{-(m(t^\star+t_d)-m(t^\star))}$

Table 2: SRGM Comparison of Chromium Version

Version	Model	a	b	c	R_{sq}	RMSE	Optimal Release
	ISS	1015.494	0.011	1.790	0.995	3.194	903.237
	DSS	**200.404**	**0.070**	**0.000**	**0.980**	**6.680**	**137.000**
88	GO	3628.386	0.001	0.000	0.995	3.394	6757.000
	ISS	275.587	0.056	3.290	0.997	3.040	185.708
	DSS	**242.567**	**0.065**	**0.000**	**0.990**	**5.614**	**148.000**
89	GO	5045.921	0.001	0.000	0.994	4.205	8446.000
	ISS	644.074	0.006	-0.115	0.996	2.837	1256.095
	DSS	**161.389**	**0.088**	**0.000**	**0.979**	**6.357**	**109.000**
90	GO	595.698	0.007	0.000	0.996	2.839	1065.000
	ISS	1168.192	0.005	0.673	0.992	5.367	1638.823
	DSS	**224.539**	**0.056**	**0.000**	**0.975**	**9.332**	**167.000**
91	GO	2736.585	0.001	0.000	0.992	5.353	5637.000

Conclusion

- Delayed S-Shape predicted the number of vulnerabilities remaining and calculated the minimum optimal release from 109 to 167 days.

The main contribution of the paper includes:

- Evaluation of software reliability growth models and reliability metrics in the context of software security.
- Identification of optimal release policies considering multiple software versions with software security growth.

9TH ANNUAL
HOT TOPICS in the SCIENCE OF SECURITY
April 5-7, 2022 | Virtually Hostedby UNIVERSITY OF ILLINOIS, URBANA-CHAMPAIGN
http://hot-sos.com

Realsearch
Real People · Real Projects · Real Impact

NC STATE UNIVERSITY