ISSN:2320-0790

An International Journal of Advanced Computer Technology

A Study on Cloud Security Growth Model using Non Homogenous Poisson Process

Mahtab Alam[1], Owais Shah[2]
Department of Computer Science, Noida International University, Greater Noida, U.P., INDIA,
mahtab.alam@niu.edu.in
Department of Electronics and Communication, Noida International University, Greater Noida, Uttar Pradesh
Owais.shah@niu.edu.in

Abstract: Software Security is one of the most important attributes for securing reliable and trustworthy software but unfortunately it is being least attentive among all others attributes. There are a number of models available for all the attributes like reliability, maintainability but security model is not yet developed. In this paper we propose security growth model based on log logistic that will provide a critical analysis of the underlying postulation and analyze the applicability of the proposed model during the entire process of software development. The security growth model (SGM) will play important role to provide quantitative approach to assess software in security concerns. SGM has been used to describe best performance in terms of predictability, probability and goodness-of-fit and so forth.

Keywords: Security Growth Model, Non Homogenous Poisson Process (NHPP), Cumulative Distribution Function (CDF), Probability Distribution Function (PDF), Expected Vulnerability

I. INTRODUCTION

Security means to protect the software system from unauthorized access of the software system. A formal approach of security assessment in the software life cycle is always considered best solution to protect corporate resources [1]. The cloud system that will work and response under threat environment within a specified time is one of the major attribute of security. A large number of models and methods have been proposed to improve the security of the cloud systems for the user so far. To make secure cloud system intensive and careful planning of designing phase and accurate decision-making is required. This careful planning and decision-making requires the use of cloud system security analysis model or security growth model (SGM). Security Growth Models usually have the form of random process that describes the behavior of attacks with respect to time. It specifies the general form of the dependence of the attack process on the principle factors that affect it: vulnerability detection and its removal, and the operational environment for example, Security Growth Model is done to estimate the form of the attack rate function by statistically estimating the parameters associated with a selected mathematical model. At any particular time it is possible to observe a history of the attack rate (attacks per unit time) of cloud system. Vulnerability generally forces the attack rate of a cloud system to decrease with time. The purposes of modeling are:

- To estimate the remaining time required to achieve a specified objective.
- To estimate the expected attack of the cloud system when the system is in use.

Measurement of cloud security comprises of the determination of cloud robustness. Security Growth Models have many underlying assumptions that are often violated in practice, but empirical evidence has shown that many are quite robust despite these assumption violations [2]. Because of assumption violations, it is often difficult to know which models to apply in practice. The model presented here is the based on the time execution model or the Goel-Okumoto Model [3].

II. SECURITY GROWTH MODEL

The primary objective of a security growth model is to forecast attack behaviors of the cloud system that will be experienced when the system is operational. This expected behavior changes rapidly and it can be tracked during the period in which the system is operational.

A. Basic Assumptions of Security Growth Model

- The execution times between the attacks are exponentially distributed.

Time	Attack	$\frac{\lambda(t)}{E_V}$	$1-\frac{\lambda(t)}{E_V}$	$\log(1-\frac{\lambda(t)}{E_V})$	a
1	411	1.856229	2.856229	0.455793	0.455793
2	411	1.856229	2.856229	0.455793	0.227896
3	362	1.634927	2.634927	0.420769	0.140256
4	192	0.867143	1.867143	0.271178	0.067794
5	228	1.029733	2.029733	0.307439	0.061488
6	229	1.034249	2.034249	0.308404	0.051401
7	175	0.790365	1.790365	0.252942	0.036135
8	201	0.907791	1.907791	0.280531	0.035066
9	177	0.799398	1.799398	0.255127	0.028347
10	104	0.469703	1.469703	0.167229	0.016723
11	109	0.492285	1.492285	0.173852	0.015805
12	58	0.26195	1.26195	0.101042	0.00842
EV	221.4167				

- The cumulative number of attacks follows a Non Homogeneous Poisson process (NHPP) by its expected value function λ(t).
- For a period over which the cloud system is observed the quantities of the resources that are available are constant.
- The number of vulnerabilities detected in each of the respective intervals is independent of each other.
- The mean value function is such that the number of attacks occurrences for any time t to t+Δt is proportional to the expected number of undetected attacks at time t. It is also assumed to be bounded, non-decreasing function of time with lim t→∞ λ(t)= N <∞.
- Vulnerabilities causing attacks are prevented immediately; otherwise reoccurrence of those vulnerabilities is not counted.

B. *Security Growth Model*

$\lambda(t) = E_v (1-e^{-at})$, where $E_v \geq 0$, $a>0$ (1)

λ(t) = Predicted number of attacks at time t
E_v = Expected total number of vulnerabilities in the application in infinite time (it is usually finite)
a = Roundness factor = the rate at which the attack rate decrease.
t = Calendar time/ execution time/ number of test runs.

To get the value of "a "we use the following formula

From equation (1) we have

$$1 - e^{-at} = \frac{\lambda(t)}{E_V}$$

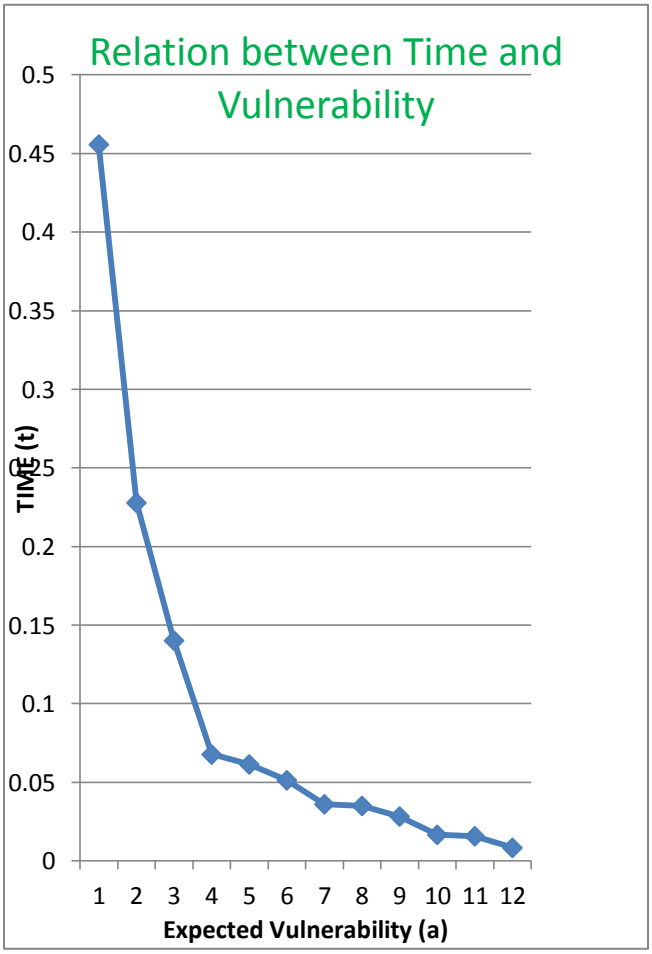
$$e^{-at} = 1 - \frac{\lambda(t)}{E_V}$$

Taking log we get

$$-at = \log\left(1 - \frac{\lambda(t)}{E_V}\right)$$

Or $$a = -\frac{1}{t}\log\left(1 - \frac{\lambda(t)}{E_V}\right)$$

Because it is a non-linear equation, the solution found may be local optimum rather global optimum. Therefore it is beneficial to define parameter values that are close to the final values [3]. The parameter values which are selected should provide a reasonable match to the existing data. But it is worthy only if the estimation is done already, or the analysis is done after the system is operational. If the result is obtained using the previous month's data, those parameter values are a good starting point to estimate the value of expected total number of attacks and the roundness factor.



III. COMPONENTS OF THE MODEL

1) Expected no. of attacks: In this model $\lambda(t)= E_vF(t)$. Here F(t) is a cumulative distribution function. F(0)=0, i.e number of attacks are 0 before the test starts, and F(∞)=1, therefore $\lambda(\infty)=E_V$ and E_E is the total number of vulnerabilities detected.

This model attempt to statistically correlate attack detection with other known functions like exponential functions. The model has a parameter that relates to the total number of vulnerabilities contained in the entire code. Residual flaws [4] can be found out if the entire number of vulnerability is detected and calculated as follows:

Residual flaws = Total number of flaws in the code - flaws detected and removed.

2) Roundness Factor: The roundness factor for a perfect circle has the value '1'and for shapes with increasing irregularity, the value tends to '0'. Other shape factors are sensitive especially for the presence of concave irregularities, whereas factors like the roundness factor can have the same value for shapes with many small concave irregularities and for elongated shapes without concave irregularities [5].

Test Time Data: For any security growth model, the appropriate measure of time must relate to the detection effort. There are three possible methods for measuring test time:

- Calendar time
- Number of detection run
- Execution (CPU) time.

Plot of Expected attacks in Security Growth Model is shown:

IV. ESTIMATION OF MODEL PARAMETERS

In the case of the above model, two parameters must be estimated: total expected attacks for infinite time (E_E) and the rate of reduction in the attack rate or the roundness factor (a).

- The parameters can be detected during two phases:
- During the detection phase. Statistical inference methods like Maximum Likelihood, Classical Least Square, and Alternative Least Square can be used to estimate the parameters in terms of calendar time.
- If the predictions are done at the operational period, then it is done through characteristics like size and complexity of the cloud system. Once the vulnerable system available in terms of execution time, these parameters may be estimated, using any statistical inference method [5]. The accuracy of the parameters generally increases with the size of the sample of attacks.

V- CONCLUSION

Software security is one of the most important features of software, but unfortunately it has been leas attended. A number of works has been done on software security but it's still lacking any model on which any one can conclude or provide any concrete ideas about security. In this particular paper I have proposed a unique model entitled Security Growth Model which describes the relation between expected vulnerability of software with time. This model is similar to Software Reliability Model.

VI. REFERENCES

[1] Mahtab Alam, "Software Security Requirement Checklist", Intl. Journal of Software Engineering, IJSE, Vol. 3, No. 1, January 2010, pp. 53-62.

[2] C Stringfellow, A Amschler Andrews "An empirical method of selecting software reliability growth models", Empirical Software Engineering, 7, 319–343, 2002.2 Kluwer Academic Publishers. Manufactured in The Netherlands.

[3] J.D.Musa, K. Okumoto, "A logarithmic Poisson execution time model for software reliability measurement", Proc. 7th International Conference on Software Engineering, Orlando, Florida, March 26-29, 1984, pp. 230-238.

[4] Alan Wood "Software Reliability Growth Models", Technical Report, Part Number 130056, September 1996

[5] Reinhold Nafe 1 , Wolfgang Schlote, "Methods for Shape Analysis of two-dimensional closed Contours - A biologically important, but widely neglected Field in Histopathology" Electronic Journal of Pathology and Histology Volume 8.2; June 2002.

[6] John D Musa, Kazuhira Okumoto "Application of basic and logarithmic poisson execution time models in software reliability measurement", Proceeding Software Reliability Modelling and Identification, Springer-Verlag London, UK ©1988, ISBN:3-540-50695-0