An Idea of an Independent Validation of Vulnerability Discovery Models *

Viet Hung Nguyen and Fabio Massacci

Università degli Studi di Trento, I-38100 Trento, Italy
{vhnguyen,fabio.massacci}@disi.unitn.it

Abstract. Having a precise vulnerability discovery model (VDM) would provide a useful quantitative insight to assess software security. Thus far, several models have been proposed with some evidence supporting their goodness-of-fit. In this work we describe an independent validation of the applicability of these models to the vulnerabilities of the popular browsers Firefox, Google Chrome and Internet Explorer. The result shows that some VMDs do not simply fit the data, while for others there are both positive and negative evidences.

1 Introduction

The vulnerability discovery process normally refers to the post-release stage where people identify and report security flaws of a released software. Vulnerability discovery models (VDM) operate on the known vulnerability data to do a quantitative estimation of the vulnerabilities present in the software. Successful models can be useful hints for both software vendors and users in allocating resources to handle potential breaches, and tentative patch update. For example, we do not exactly know the day of major snow falls but cities expect it to fall in winter and therefore plan resources for road clearing in that period.

In this paper we consider six proposed VDMs. The first model is Anderson's Thermodynamic(AT) [5]. Rescorla proposed two other models [11]: Quadratic (RQ) and Exponential (RE). The fourth model considered here is Alhazmi & Malaiya's Logistic (AML) model [2]. The fifth is directly derived from a software reliability model, Logistic Poisson (LP) (a.k.a Musa-Okumoto model). The last model is the simple linear model (LN).

Among these models, the AML model has been subject to a significant experimental validation: from operating systems [1–4] (*i.e.,* Windows NT/95/98/2K/XP, Redhat 6.2/7.1 and Fedora) to browsers [14] (*i.e.,* IE, Firefox, Mozilla), and web servers [15] (*i.e.,* ISS, Apache). The results reported in the literature show that there is not enough evidence to neither reject nor accept AML. Three browsers were considered: one is strongly accepted by AML (Mozilla), one is strongly rejected (IE), and another one is unknown (Firefox).

* This work is supported by the European Commission under projects EU-FET-IP-SECURECHANGE.

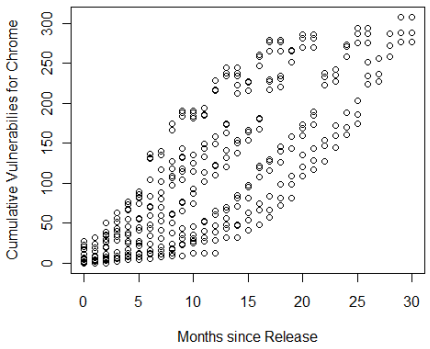


Fig. 1. Google Chrome Firework of Vulnerability Discovery Trends

These inconsistent results may be caused by a combination of factors. First, the authors did not clearly mention what a vulnerability is. For example, the National Vulnerability Database (NVD) reports a number of vulnerabilities which the security bulletin of the vendors do not classify as such. By considering different database we could get different trends.

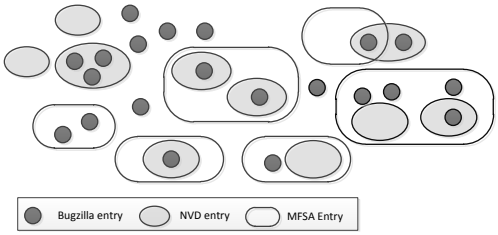
The second problem is that the authors considered all versions of software as a single application, and counted vulnerabilities for this "application". Massacci *et al.* [8] has shown that each Firefox version has its own code base, which may differ by 30% or more from the immediately preceding one. Therefore, as time goes by, we can no longer claim that we are counting the vulnerabilities of the same application. To explain visually this problem, Fig. 1 shows in one plot the cumulative vulnerabilities of the different versions of Chrome in which we restart the counters for each version. It is immediate to see that there is not a single "trend" but a "firework" effect where each version determines its own trajectory.

1.1 Contribution of this Paper

This paper presents an independent validation experiment on the goodness-of-fit of six existing VDMs against the three most popular browsers: Firefox, Google Chrome and Internet Explorer.

- We show that some model (AT) does not work completely. Some (LN, RE, RQ, LP) might not work, and some (AML) may work.
- We find an interesting phenomenon that cumulative vulnerabilities of a life long software may have many saturation points (where the number of vulnerabilities is stable) which might falsify all of existing VDMs.

The rest of the paper is organized as follows. In the subsequent section (§2), we describe our research questions and how to find out the answers. Next we briefly discuss existing VDMs and their formulae (§3). After that, we discuss the



This illustrates different abstract levels of vulnerability: from technical level (Bugzilla) to abstract level (MFSA, Bugzilla). Bugzilla entry denotes technical programming issues (both security and non-security ones). Security bugzilla are ones reported in an MFSA, or referenced by an NVD. The overlaps between notations denote that an report might reference to another report.

Fig. 2. The vulnerability space of Firefox.

methodology to conduct the experiment, and a discussion about the result in our experiment (§4). Finally, we present potential threats (§5) to the validity of our work and conclude the paper with future work (§6).

2 Research Questions

The primary question is "does this model fit the observed data?". Frequently when a new VDM is proposed, the authors have done some experiment to validate the applicability of this VDM. Mostly, in their reports the proposed VDMs often have good goodness-of-fit measures. As time goes by, the goodness-of-fit may improve or deteriorate as more data become available (either in terms of data point for the same software or new software to be considered as an instance). This motivate our first research question:

RQ1 *Are existing VDMs able to fit cumulative numbers of vulnerabilities of the popular browsers (i.e., IE, Firefox, and Chrome)?*

To find the answer, we touched another, major and almost foundational issue: "what is a vulnerability?". Most related work did not explicitly discuss this question. Normally, a vulnerability is a security report describing a particular problem of a particular application, for instance: a report in Mozilla Foundation Security Advisories (a.k.a an MFSA entry), or an NVD report of NIST (NVD entry). In the wisdom of many people, an NVD entry is a vulnerability, but there are many other definitions [6, 7, 12].

Fig. 2 illustrates the vulnerability space of Firefox, in which different 'kinds' of Firefox vulnerabilities are coexisted at different level of abstraction.

- *Mozilla Bugzilla* (or *bug*): contains very technical reports for vulnerabilities, but also other normal programming bugs.
- *NVD*: holds high level third-party security reports for several applications, including Firefox. Many NVD entries (gray ovals) mentioning Firefox maintain references to Bugzilla (black circles inside ovals).

Table 1. Formal definitions of six VDMs in the study.

Model	*Formula*
Alhazmi-Malaiya Logistic (AML)	$\Omega(t) = \frac{B}{\frac{B}{K}Ce^{-ABt} + 1}$
Anderson Thermodynamic (AT)	$\Omega(t) = \frac{K}{\gamma}\ln(t) + C$
Linear (LN)	$\Omega(t) = At + B$
Logistic Poisson (LP)	$\Omega(t) = \beta_0 \ln(1 + \beta_1 t)$
Rescorla Exponential (RE)	$\Omega(t) = N(1 - e^{-\lambda t})$
Rescorlar Quadratic (RQ)	$\Omega(t) = \frac{At^2}{2} + Bt$

– *MFSA*: are set of vendor's high level security reports for Mozilla's products. Each MFSA entry (rounded rectangle) always references to one or more bugs (black circles inside) responsible for this security flaw. MFSA also holds links to corresponding NVD entries (overlapped ovals).

Depend on the judgement of analysts, different numbers of vulnerabilities are observed and collected. Here, in Fig. 2, if we define a vulnerability is an MFSA, or NVD, or Bugzilla, these numbers are respectively six, ten and fourteen. The fact that we can have a large variance in numbers raise another research problem *"How do different definitions of vulnerability impact the VDMs' goodness-of-fitness?"*. However, we do not present it here due to the limit of space.

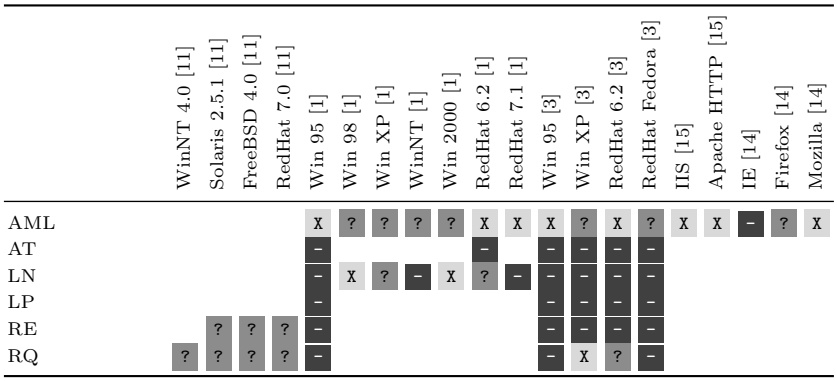
3 Vulnerability Discovery Models

This section provides a quick glance about six VDMs. As denoted in [3], these VDMs are main features of the vulnerability discovery models. Here, only the formulae of these six models are discussed. The detail rationale of models as well as the meaning of each parameter can be found in the original work or in [3]. All these parameters are estimated using non-linear regression on observed data.

– *Alhazmi-Malaiya Logistic (AML)*: proposed by Alhazmi & Malaiya [1], inspired by the s-shape curve.
– *Anderson Thermodynamic (AT)*: the application of this model to vulnerabilities is proposed in [5]. The term *thermodynamic* originates by the analogy from thermodynamics, in which γ accounts for the lower failure rate during beta testing compared to higher rates during alpha testing.
– *Linear model (LN)*: this is the simplest model, and well known by most people. Linear model is often used to express the trend line of data.
– *Logistic Poisson (LP)*: is originated from the field of reliability engineering, also known as Musa-Okumoto model.
– *Rescolar Exponential (RE)*: is proposed by Rescorla [11] while attempting to identify trends in the vulnerability discovery using statistical tests.
– *Rescolar Quadratic (RQ)*: this model is also a work of Rescorla [11], inspired by the Goel-Okumoto in software reliability engineering.

Table 2. The goodness-of-fit of VDMs in other studies.

The table reports goodness-of-fit from previous studies. Columns are applications of which vulnerabilities are fitted. The number next to each application is citation to the corresponding study.



4 Validation of VDMs

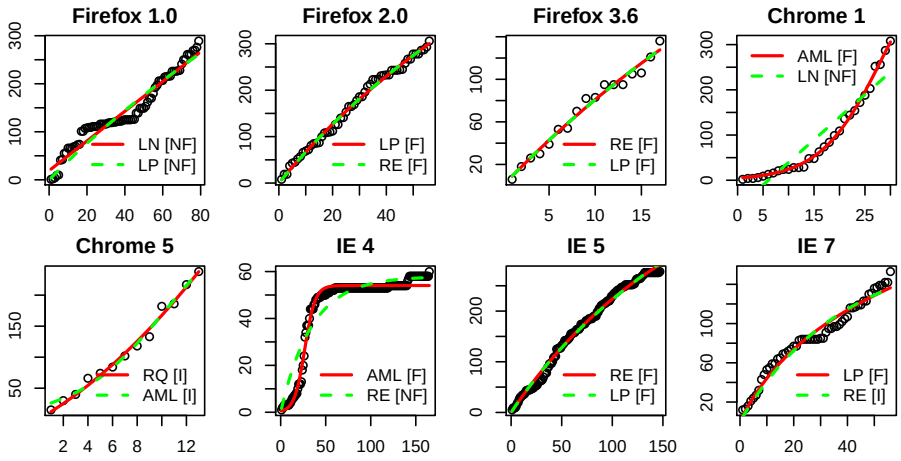
The validation is quire straight forward. We fit VDMs into the data set using R [10] tool. The differences between expected values of each generated model and observed values are calculated and tested by the chi-square (χ^2) goodness-of-fit test. This test is based on χ^2 statistics calculated as follows.

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} \tag{1}$$

O_i and E_i orderly denote the observed values and expected values generated by VDMs. The smaller χ^2, the higher goodness a VDM gains. In practice, a VDM is acceptably fitted if the χ^2 is less than a critical value, given a significant level (α) and degrees of freedom. The *p-value* here represents the significance of the differences between observed values and expected values. If the *p-value* is small, differences are significant, not by chance. Thus, the smaller *p-value*, the stronger evidence a VDM does not fit the data. Hence, we interpret the goodness-of-fit based on the ranges of *p-value* as follows

- *Not Fit*: $p - value \in [0 \sim 0.05)$, the difference are not by chance. So, this evidence is strong enough to reject the model.
- *Good Fit*: $p - value[0.95 \sim 1.0]$, the difference, in opposite with the previous, is significant small. It is a strong evidence to accept the model.
- *Inconclusive Fit*: $p - value \in [0.05 \sim 0.95)$, there is not enough evidence neither to reject nor accept this model.

Applying goodness-of-fit interpretation discussed above, Table 2 show the goodness-of-fit of VDMs in other studies. In these studies, Windows 95/XP and RedHat 6.1 have been tested in two studies (*i.e.,* [1,3]) of the same authors, but they seem to be duplicated. Thus we will consider these two experiments as one.



This figure illustrates feature goodness in Table 3. The circles indicate cumulative vulnerabilities at a certain time. The horizontal axis (X) is time-in-market measured by the number of months since officially released. The vertical axis (Y) is the cumulative vulnerabilities.

Fig. 3. Goodness of VDMs on browsers in database NVD.

According to the table, AML is the one that has been tested in various kinds of applications, *e.g.,* operating systems, web servers, and browsers. Most of the cases, AML shows its outstanding performance (only 1 *Not Fit* over 13 tests). On the contrary, AT model also did not work in all cases. For the other models, the ratio between *Not Fit* and *Inconclusive + Good Fit* is more or less fifty-fifty. Therefore, we cannot conclude anything about the performance of these models.

We run our experiment of five VDMs on seventeen releases. The experiment produces 102 curves, which are impossible to show all of them. Fig. 3 shows the some fitted plots of VDMs for releases using NVD data set. For Firefox v1.0, the cumulative number of vulnerabilities has more than one linear periods. This trend is against the recently three-phase model (*i.e., learning, linear* and *saturation*) proposed by Alhazmi *et al.* [1]. So none of VDM is able to fit (either *Good Fit* or *Inconclusive Fit*) this version. This trend of Firefox vulnerabilities is caused by a large portion of v1.0's code base is inherited in later releases. Thus lots of vulnerabilities applied to Firefox v1.0 are discovered in the newer releases [8]. This phenomenon slightly appears in IE v4.0, but the stable period of this release is long enough[1] for the AML model to obtain a *Good Fit*, nonetheless. For Firefox v3.6 and Chrome v3.5, these releases are still young (less than 16 months old), the vulnerability discovery is in the linear phase. So any VDM that supports linear modeling (or nearly linear), *i.e.,* AML, LN, LQ, RQ, RE, has a chance to fit the data.

Table 3 reports the goodness-of-fit for 102 curves. Here, instead of reporting a big table of numbers, Table 3 shows the interpretation of *p-value* of the χ^2

[1] A long stable period has larger degree of freedom in the χ^2 test, thus there is more chance that the *p-value* is less than the significant level 0.05.

Table 3. The goodness-of-fit of VDMs using data set NVD.

The goodness of fit of a VDM is based on *p-value* in the χ^2 test. *p-value* < 0.05: not fit (–), *p-value* ≥ 0.95: good fit (X), and inconclusive fit (?) otherwise.

	Firefox						Chrome						IE				
Model	v1.0	v1.5	v2.0	v3.0	v3.5	v3.6	v1.0	v2.0	v3.0	v4.0	v5.0	v6.0	v4.0	v5.0	v6.0	v7.0	v8.0
AML	–	–	?	?	?	?	X	?	?	?	?	?	X	?	?	–	X
AT	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	?	–
LN	–	–	X	–	X	?	–	–	–	?	–	–	–	–	–	?	?
LP	–	–	X	?	X	X	–	–	–	–	?	?	–	X	–	X	?
RE	–	–	X	?	X	X	–	–	–	–	?	?	–	X	–	?	?
RQ	–	–	–	?	?	X	–	–	?	?	?	?	–	–	–	–	X

tests. This presentation also helps to study at higher abstract level than the raw *p-value*s. Basically, our result is consistent with others. In this table, there are 47 times VDMs can either well fit or inconclusively fit the data, and 55 times they do not work. Roughly speaking, the chance of not fit is about 50%. If we look at each VDM particularly, the AML model appears to be the best one as it obtains more positive results than others. In contract, the AT model seems to be the worst because it could only fit one release (IE v7.0). Meanwhile, other models are equivalent in number of times being rejected and accepted, except the LP model which is likely a bit better. Even though our result confirms the conclusion in previous studies, we still could not claim any strong argument about the goodness-of-fit of these VDMs since the goodness-of-fit might change overtime as more data will be available. We can only say that at the time when we collect data, AML is the best model that can fit most releases in our study; AT model apparently does not applicable; and other models work in haft way.

5 Threats to Validity

Bias in data collection. This work employs the same technique discussed in [9] to parse HTML pages of MFSA, and process the XML data of NVD and Bugzilla. Even though the collector tool has been checked for multiple times, it might contain bugs affecting to data collection.

Error in curve fitting. We estimate the goodness-of-fit of VDMs by using the Nonlinear Least-Square technique implemented in R (`nls()` function). This might not produce the most optimal solution. That essentially impacts the validity of this work. To mitigate this issue, we additionally employ a commercial tool *i.e.,* CurveExpert Pro[2] to cross check the goodness-of-fit.

6 Conclusion and Future Work

In this work we validated the goodness-of-fit of several VDMs on the three most popular browsers: IE, Firefox and Chrome. Our validation took into account the definition of vulnerability which is not adequately considered in previous

[2] http://www.curveexpert.net/, site visited on 16 Sep, 2011

studies. However we have not enough room to report the result. Even though our experiment is consistent with other studies, but all the experiments so far have only reported the goodness-of-fit of these VDMs at certain time points of a software life cycle. Meanwhile, we need to analyze the evolution of each model in a long period and see how the goodness-of-fit evolves to have a better insight.

Additionally, we have shown the potential impact of different understanding about what a vulnerability is. Hence, it would be interesting to study which one is more appropriate for VDMs in general.

References

1. O. Alhazmi and Y. Malaiya. Modeling the vulnerability discovery process. In *Proc. of the 16th IEEE Int. Symp. on Software Reliab. Eng. (ISSRE'05)*, 2005.
2. O. Alhazmi and Y. Malaiya. Quantitative vulnerability assessment of systems software. In *Proc. of RAMS'05*, 2005.
3. O. Alhazmi and Y. Malaiya. Application of vulnerability discovery models to major operating systems. *IEEE Trans. on Reliab.*, 57(1):14–22, 2008.
4. O. Alhazmi, Y. Malaiya, and I. Ray. Security vulnerabilities in software systems: A quantitative perspective. *Data and App. Sec. XIX*, 3654:281–294, 2005.
5. R. Anderson. Sec. in open versus closed systems - the dance of Boltzmann, Coase and Moore. In *Proc. of Open Source Soft.: Economics, Law and Policy*, 2002.
6. A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions On Dependable And Secure Computing*, 1(1)):1133, 2004.
7. I. Krsul. *Software Vulnerability Analysis*. PhD thesis, Purdue University, 1998.
8. F. Massacci, S. Neuhaus, and V.H. Nguyen. After-life vulnerabilities: A study on firefox evolution, its vulnerabilities and fixes. In *Proc. of ESSoS'11*, 2011.
9. F. Massacci and V.H. Nguyen. Which is the right source for vulnerabilities studies? an empirical analysis on mozilla firefox. In *Proc. of MetriSec'10*, 2010.
10. R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, 2011. ISBN 3-900051-07-0.
11. E. Rescorla. Is finding security holes a good idea? *IEEE S&P*, 3(1):14–19, 2005.
12. Fred B. Schneider. Trust in cyberspace. *National Academy Press*, 1991.
13. J. Sliwerski, T. Zimmermann, and A. Zeller. When do changes induce fixes? In *Proc. of the 2nd Int. Working Conf. on Mining Soft. Repo. MSR('05)*, 2005.
14. S. Woo, O.Alhazmi, and Y. Malaiya. An analysis of the vulnerability discovery process in web browsers. In *Proc. of 10th IASTED SEA'06*, 2006.
15. S. Woo, H. Joh, O. Alhazmi, and Y. Malaiya. Modeling vulnerability discovery process in apache and iis http servers. *C&S*, 30(1):50 – 62, 2011.