

# Measuring and Enhancing Prediction Capabilities of Vulnerability Discovery Models for Apache and IIS HTTP Servers

Omar H. Alhazmi and Yashwant K. Malaiya

Colorado State University  
omar|malaiya@cs.colostate.edu

## Abstract

*The prediction of the number of vulnerabilities in an HTTP server can allow us to evaluate the security risk associated with its use. Vulnerability discovery models have recently been proposed which can be used to estimate the future number of vulnerabilities expected to be discovered. A detailed analysis of the prediction capabilities of two models termed AML and LVD for the vulnerabilities in the two major HTTP servers is presented. Four complete data sets for Apache and IIS are used, representing an open source and a commercial web server respectively. Both long term predictions involving several years and short term predictions for the following year are considered. Potential methods for enhancing the prediction accuracy are considered. The results show good predictive capabilities of the AML model when constraints are used for estimating the model parameters. The LVD model works well in some special cases when saturation has not yet set in. The results can be used by both developers to plan the test and maintenance effort needed and by users to assess the potential security risks associated with a specific server.*

## 1. Introduction

To assure an acceptable degree of security for a web server the developers need to determine how much testing for security vulnerabilities is needed. Moreover, developers need to be able to project the post-release vulnerability discovery rate to plan the maintenance and patch development effort needed. The available data can be fitted to the Vulnerability discovery models (VDMs) to project the trend that the vulnerabilities discovery process is likely to follow.

There has been considerable discussion of the security of web servers in recent years. However, investigations have focused on qualitative issues related to detection and prevention of individual vulnerabilities. Quantitative data is sometimes cited, but without any significant critical analysis. Methods need to be developed to allow security related risks to be evaluated

quantitatively in a systematic manner. A study by Ford et al. [1] has compared several servers, the number of vulnerabilities and the associated severity levels. This study identifies a need to develop tools for estimating the risks posed by vulnerabilities.

The two major software components of the Internet are an HTTP (Hyper Text Transfer Protocol) server, also termed a web server, and the browser, which serves as the client. Both of these were first introduced in 1991 by Tim Berners-Lee of CERN and they have now become indispensable parts of both organizational and personal interactions. The early web servers provided information using static HTML pages. The web server now provides dynamic and interactive services between the server and client using database queries, executable script, etc. The web server is able to support functions such as serving streaming media and mail. An HTTP server has thus emerged as a focal point for the Internet.

In this paper we examine the vulnerabilities in the two most widely-used HTTP servers, the Apache server and the Microsoft IIS (Internet Information Service), both introduced in 1995. While Apache has a much larger overall market share, roughly 70%, IIS may have a higher share of the corporate websites. The market share for other servers is small and thus they are not examined here. IIS is the only HTTP server that is not open-source. Both Apache and IIS are generally comparable in features; however, IIS runs only under the Windows operating systems, whereas Apache supports all the major operating systems.

IIS 2.0 was supplied as part of the NT 4.0 operating systems. The next version, IIS 3.0, was followed by IIS 4.0, the most popular version. After that, IIS 5.0 was embedded in Windows 2000; recently, IIS 6.0 was separated from Windows 2003 and is available as an independent application.

The security of systems connected to the Internet depends on several components of the system. These include the operating systems, the HTTP servers and the browsers. Some of the major security compromises arise because of vulnerabilities in the HTTP servers. A *vulnerability* is defined as “a defect which enables an

attacker to bypass security measures” [2]. Vulnerabilities are a special class of defects that can permit circumvention of security measures. The vulnerabilities found are disclosed by the finders using some of the common reporting mechanisms available in the field. The databases for the vulnerabilities are maintained by organizations such as the National Vulnerabilities Database [3], MITRE [4], Bugzilla [5], BugTraq [6], as well as by the developers of the software. The exploitations of some of the server vulnerabilities are well known. The Code Red worm [7], which exploited a vulnerability in IIS (described in Microsoft Security Bulletin MS01-033, June 18, 2001), appeared on July 13, 2001, and soon spread world-wide in unpatched systems.

All the computing systems connected to the network are subject to some security risk. While there have been many studies attempting to identify causes of vulnerabilities and potential counter-measures, the development of systematic quantitative methods to characterize security has begun only recently. There has been considerable debate comparing the security attributes of open source and commercial software [8]. However, for a careful interpretation of the data, rigorous quantitative modeling methods are needed. The likelihood of a system being compromised depends on the probability that a newly discovered vulnerability will be exploited. Thus, the risk is better represented by the not yet discovered vulnerabilities and the vulnerabilities discovery rate rather than by the vulnerabilities that have been discovered in the past and remedied by patches. Possible approaches for a quantitative perspective of exploitation trends are discussed in [9], [10]. Probabilistic examinations of intrusions have been presented by several researchers [11]. In [12], Rescorla has studied vulnerabilities in open source servers. The vulnerabilities discovery process in operating systems has just recently been examined by Rescorla [13] and by Alhazmi and Malaiya [14][15][16].

Servers are very attractive targets for malicious attacks because they represent the first line of defense that, if bypassed, can compromise the integrity, confidentiality and availability attributes of the enterprise security. Thus, it is essential to understand the threat posed by both undiscovered vulnerabilities and recently discovered vulnerabilities for which a patch has not been developed or applied. In this paper we address questions such as: How can we predict the vulnerabilities not yet discovered? How accurate are these estimations? We also consider methods for enhancement of the prediction capabilities.

At this time, despite the significance of security in the HTTP servers, very little quantitative work has been done to model the vulnerabilities discovery process for the servers. Such work would permit the developers and

the users to better estimate future vulnerabilities discovery rates. Use of reliability growth models is now common in software reliability engineering [17][18]. SRGMs take into account the fact that as bugs are found and removed, fewer bugs remain. Therefore, the bug finding rate gradually drops and the cumulative number of bugs eventually approaches saturation. Such growth models are used to determine when a software system is ready to be released and what future failure rates can be expected.

Some vulnerability discovery models were recently proposed by Anderson [8], Rescorla [12], Ozmont and Schechter [19] and Alhazmi and Malaiya [14]. The applicability of these models to several operating systems was examined in [20]. The results show that while some of the models fit the data for most operating systems, others do not fit well or provide a good fit only during a specific phase.

The next section previews the vulnerability discovery models used. We then consider the total number of vulnerabilities in the two versions of two HTTP servers and examine how well the models fit the available data. Afterwards, we will examine the goodness of fit of the two models on all data sets. In the following sections we will evaluate the prediction capabilities using those datasets and examine some enhancements that can improve the accuracy of predictions. Lastly, we will discuss the major observations and present the conclusions.

## 2. The Vulnerability Discovery Models

Here we investigate the applicability of two vulnerability discovery models for HTTP servers. The models used are a logistic model and a linear model proposed by Alhazmi and Malaiya [14][17]. These two models have been found to fit datasets for several of the major Windows and Linux operating systems, as determined by goodness of fit and other measures. The logistic model considers the three phases of vulnerabilities discovery, including the effect of the rising and declining market share on the software. The second model can be viewed as a simplification of the logistic model with fewer parameters and easier applicability.

The Alhazmi-Malaiya Logistic Model (AML) assumes that the rate of change of the cumulative number of vulnerabilities  $\Omega$  is governed by two factors, as given in Equation 1 below [14]. The second factor declines as the number of remaining undetected vulnerabilities declines. The first factor increases with time to take into account the rising share of the installed base. The saturation effect is modeled by the first factor. It is possible to derive a more complex model; however, this model provides a good fit to the data, as shown

below. Let us assume that the vulnerabilities discovery rate is given by the differential equation:

$$\frac{d\Omega}{dt} = A\Omega(B - \Omega), \quad (1)$$

where  $\Omega$  is the cumulative number of vulnerabilities,  $t$  is the calendar time, and initially  $t=0$ .  $A$  and  $B$  are empirical constants determined from the recorded data. By solving the differential equation, we obtain

$$\Omega(t) = \frac{B}{BCe^{-ABt} + 1}, \quad (2)$$

where  $C$  is a constant introduced while solving Equation 1. Equation 2 gives us a three-parameter model given by the logistic function. In Equation 2, as  $t$  approaches infinity,  $\Omega$  approaches  $B$ . Thus, the parameter  $B$  represents the total number of accumulated vulnerabilities that will eventually be found. As Figures 1 to 4 show, the model has an s-shape and it displays three phases, the initial learning phase, the middle linear phase and the final saturation phase.

Available data for some systems [14] [15] suggest that the early learning phase is sometimes negligible. This may be due to the similarity between the new release and its prior version, which results in a shorter learning phase for the testers to become familiar with the new release. Moreover, it has been observed that during the saturation phase, the vulnerabilities discovered in the next version include vulnerabilities shared with the modeled version. This can prolong the linear trend even though an increasing market share is taken by the newer version.

The AML model has three parameters causing some instability of the estimated parameter values. This instability can be reduced by applying a constraint requiring the duration between the two transition points within some limit. This computational approach termed *AML constrained (AML-C)*. The duration has been shown to be  $2.63/AB$  for the AML model [20]. Hence, we can limit  $2.63/AB$  between some minimum and maximum values chosen using the values from previous software systems. This constraint assumes that the transition points are within the time-frame of the expected lifetime of the software, thereby enforcing the S-shape and anticipating the two transition points.

The Linear Vulnerability Discovery model (LVD) is another example of time-based vulnerability discovery models. The LVD model assumes that the rate of change is constant. It is simpler than AML because the LVD model has one main parameter, slope  $S$ , which represents the vulnerabilities discovery rate, in addition to a constant  $k$ .  $\Omega$  is given by [16].

$$\Omega(t) = k + St \quad (3)$$

The LVD can also be considered an approximation of the AML model, when there is no saturation and the learning phase is negligible.

A few other vulnerability discovery models have been proposed, however these two have been found in the past to fit the data best [16], thus only these two models are considered here.

### 3. Modeling Vulnerabilities in HTTP Servers

In this section, the datasets for the total vulnerabilities of the Apache and Microsoft IIS web servers are fitted to the two models and the goodness of fit is evaluated to determine how well the models reflect the actual vulnerabilities discovery process. The vulnerability dataset were extracted from the National Vulnerabilities Database [3] maintained by NIST and from Mitre Corporation [4], whereas the market share data is from Netcraft [21]. We have used data for Apache 1 and 2 representing open source software and two versions of IIS 4 and 5 representing two versions of closed source software. The total number of vulnerabilities found in the four versions is shown in Table 1. It should also be noted that the number of vulnerabilities, either found or estimated as remaining, should not be the only measurement of a security threat. Factors such as patch development, application delays and vulnerabilities' exploitation rates also need to be considered.

**Table 1. Market share, Number of vulnerabilities and release dates of some web servers**

	Apache		IIS		SJSWS (SunOne)	Zeus
<b>First Release</b>	1995		1995		2002	1995
<b>Market Share</b>	69.7%		20.92%		2.53%	0.78%
<b>Version</b>	1.x	2.x	4.0	5.0	Up to 6.1	Up to 4.3
<b>Vulnerabilities</b>	58	45	85	73	3	5

#### 3.1 Web server market share

Market share is one of the most significant factors impacting the effort expended in exploring potential vulnerabilities. Higher market share offers a greater incentive to explore and exploit vulnerabilities because attackers will obviously find it more profitable or satisfying to spend their time focusing on a software system having a greater market share.

Table 1 above presents data obtained from NVD and Netcraft, showing the current web server market share and total number of vulnerabilities found to date. As we can see from the table, for servers with a lower percentage of the market, such as Sun Java System Web

Server (SJSWS) and Zeus, the total number of vulnerabilities found is low. This does not mean that these systems are vulnerabilities-free, but merely that only limited effort has gone into detecting their vulnerabilities. A significant number of vulnerabilities have been found in both Apache and IIS, illustrating the impact of the market share on the motivation for exploring and finding undiscovered vulnerabilities.

Apache and Microsoft IIS dominate the web server market, while other web servers such as Sun Java System Web Server and Zeus occupy a very small share of the market, as shown in the Table 1 above .

The Apache HTTP server was first released in mid 1995. Since then it has gained wide popularity and is used by over 50 million web server systems. Apache dominates the market, probably because it is an open source system that is free. Apache may also have benefited from not having been exposed to serious security issues such as the Code Red [7] or Nimda worms that were faced by IIS in 2001.

In this section, we fit the vulnerabilities data for Apache to the Alhazmi-Malaiya Logistic model (AML) and to a linear model (LVD). Figure 1 gives the vulnerabilities data obtained from NVD [3] for the period between March 1996 and December 2005.

### 3.2 Goodness of fit analysis for Apache and IIS

Figure 1 shows Apache 1.x (which includes subversions 1.0 through 1.3) vulnerabilities fitted to the AML model and the LVD model, clearly showing that the AML follows the date very closely and has fitted with much less error than the LVD model. However, in, Figure 2 the Apache 2.x data set is shown to fit both models equally well. Figure 3 and Figure 4 show the fit for the two models for IIS 4 and 5 respectively.

Apache 2 (including subversions through 2.1) appears to be in the linear phase, since the number of vulnerabilities still appears to be growing linearly. Despite having been on the market for several years, Apache 2 and even Apache 1 have not reached a clear saturation phase, possibly because of their larger market share. Moreover, the number of systems using the Apache web server is still increasing; indicating that vulnerabilities discovery for Apache can be expected to continue at a significant pace in near future.

On the other hand, the IIS web server appears to have reached the saturation phase. During the past several months, the vulnerabilities discovery rate for IIS has dropped to a very low point. A possible explanation for this can be that the number of IIS web servers installed appears to be stationary, unlike the Apache server which is still gaining in terms of new installations. Another possibility is that the number of remaining undiscovered vulnerabilities may actually have dropped significantly.

Figures 1 to 4 show the data fitted to the model using the least square fit. For Chi-squared goodness of fit test, we chose an alpha level of 5%;  $\chi^2$  test results are shown in Table 2 and Table 3. This table shows that all data sets have fitted the AML model with P-value > 0.991. The values of the parameter  $A$  range from 0.0008 to 0.002, and for  $C$  they range from 0.18 to 0.70. The parameter  $B$  corresponds approximately to the number of vulnerabilities.

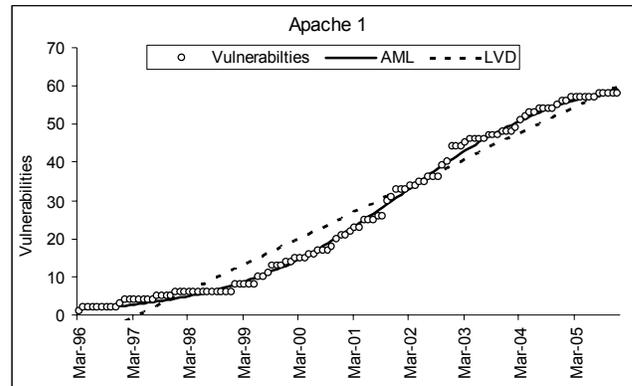


Figure 1. Apache 1.x Vulnerabilities Data

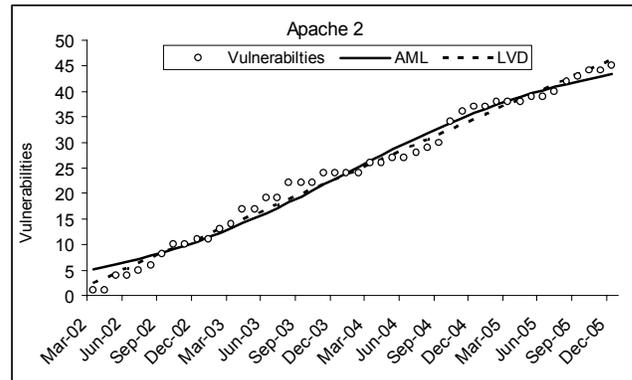


Figure 2. Apache 2.x Vulnerabilities Data

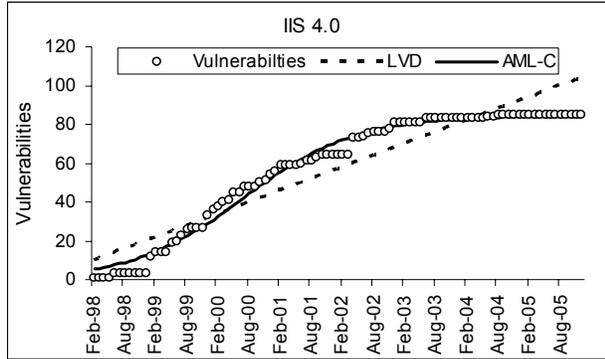
Table 2. Goodness of Fit Tests Results for AML

	A	B	C	$\chi^2$	$\chi^2_{critical}$	P-value
Apache 1	0.00083	64.160	0.705	39.7	138.8	1
Apache 2	0.002014	48.05	0.192	14.6	61.65	0.999995
IIS 4	0.001106	84.313	0.191	65.44	118.75	0.991
IIS 5	0.001488	72.379	0.181	15.19	91.67	1

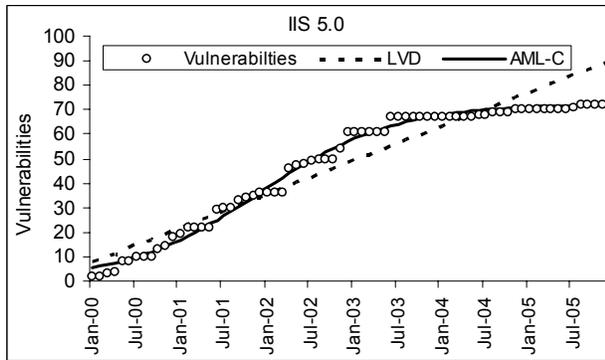
The goodness of fit for the LVD model for the later versions Apache 2 and IIS 5 is significant, implying that the vulnerability discovery has not yet reached the saturation phase; whereas the data for the older versions Apache 1 and IIS 4 did not fit the LVD model. By observing the *slope* parameter we can see that the rate of vulnerabilities discovery given by the LVD model is higher for Apache 2 than Apache 1, and IIS 5 rate is higher than IIS 4.

**Table 3. Goodness of Fit Tests Results for the LVD**

	Slope(S)	Constant(k)	$\chi^2$	$\chi^2_{critical}$	P-value
Apache 1	0.575	-7.986	199.48	138.8	0.0000001
Apache 2	0.976	1.386	6.2	61.65	1
IIS 4	0.999	9.342	239.11	118.75	$3.99 \times 10^{-11}$
IIS 5	1.157	6.664	79.58	91.67	0.227



**Figure 3. IIS 4.0 vulnerabilities data fitted to the models**



**Figure 4. IIS 5.0 vulnerabilities data fitted to the models**

#### 4. LONG TERM PREDICTION

The analysis in the previous section shows that the AML model fits all datasets while the LVD model has fitted only some of them. However, we should really judge the models by examining the accuracy of future projections about the vulnerabilities using the data available in hand. A model with good predictive capabilities can be used to estimate the resources needed for maintenance and the risk associated with a particular web server.

We evaluate predictability by examining the accuracy of prediction by simulating real life situations using early partial data. We apply the model under consideration for  $n$  time instants ( $t_1, t_2, t_3 \dots t_n$ ) with equal calendar time interval periods between estimations. For each  $t_i$ , the partial data available at  $t_i$  is fitted to the model using regression analysis to determine the best

values for the parameters. The parameters are used to obtain the estimated number of the total vulnerabilities ( $\Omega_i$ ) at the end of the time period examined. The estimates of the numbers of vulnerabilities ( $\Omega_1, \Omega_2, \Omega_3 \dots \Omega_n$ ) are compared with the actual number of vulnerabilities ( $\Omega$ ) to evaluate the normalized estimation error  $(\Omega_i - \Omega) / \Omega$ . We then take the average of the normalized error magnitude values to obtain the measure of average error ( $AE$ ). Average bias ( $AB$ ) is similarly obtained when the sign of the error is also considered [22]. The average error ( $AE$ ) and average bias ( $AB$ ) are given by:

$$AE = \frac{1}{n} \sum_{i=1}^n \left| \frac{\Omega_i - \Omega}{\Omega} \right| \tag{4}$$

$$AB = \frac{1}{n} \sum_{i=1}^n \frac{\Omega_i - \Omega}{\Omega} \tag{5}$$

where  $\Omega$  is the actual number of vulnerabilities, while  $\Omega_i$  is the number of vulnerabilities predicted at time  $t_i$ .

We examine the accuracy of the predictions made using three approaches AML, AML-C and LVD. AML-C is the AML model with the constraint ( $21 < 2.63/AB < 42$ ) used during numerical parameter estimation, the constraint restricts the linear phase to be within 21 to 42 months which enforces the S-shaped characteristic of the AML model. The constraint was able to help in avoiding extreme parameter values during estimation using AML especially when the available dataset was still small [17]. The constraint is generally automatically satisfied when the dataset gets larger.

In the next two subsections, the plots of normalized error values are given against normalized time in Figures 5 to 12. The x-axis gives time as the percentage of the overall time, and the y-axis give the normalized error.

#### 4.1 Long term prediction accuracy test

The normalized error values for the estimates of the total number of vulnerabilities for Apache 1 are shown in Figure 5. AML-C shows significant improvement over AML by avoiding extreme values, during the early part. LVD has shown stable performance.

The error values for Apache 2 illustrated by Figure 6 shows that LVD has performed well, followed by AML-C, then AML. **Error! Reference source not found.** summarizes the Average error and Average bias of the estimations.

IIS 4 and 5 estimation errors, illustrated by Figure 7 and Figure 8 respectively, show that AML-C has the most accurate estimations, followed by AML. LVD was generally shown to have the higher average error in most cases (see Table 4 **Error! Reference source not found.**). This suggests that the LVD model was unable to model the saturation phase for the vulnerabilities in IIS datasets.

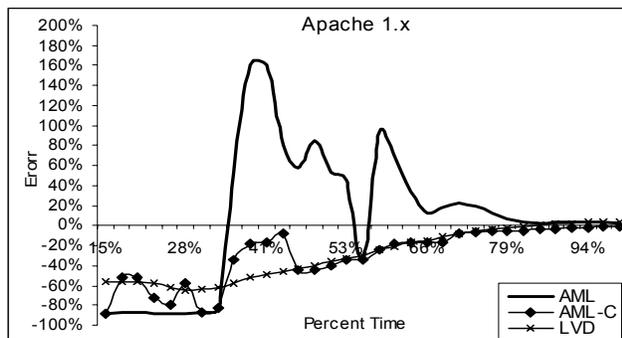


Figure 5. Accuracy of the models in estimating Apache 1.x vulnerabilities data

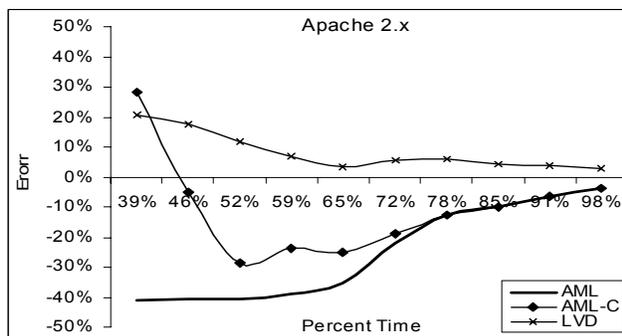


Figure 6. Accuracy of the models to estimate Apache 2.x vulnerabilities data

Table 5 below shows the average error and average bias values calculated when at least 65% of the data was available for making the projections. We chose 65% because it represents the time period when enough early data is available and significant part of the duration still remains. After about 65% of the time, the AE value drops to single digit in many cases. This table demonstrates that LVD projections still result in high error values for the two IIS versions. The AML-C results are generally the best with AE ranging from 2.1 to 10.4% whereas LVD can result in AE values of 41% for the two IIS versions.

The AB values in the tables suggest that the projections are biased. In the next subsection, an adaptive estimation technique which attempts to reduce the bias is discussed and evaluated.

Table 4. Average Error and Average Bias for Prediction of  $\Omega$

Model Server	AML		AML-C		LVD	
	AE	AB	AE	AB	AE	AB
Apache 1.0	52.5%	8.3%	29.8%	-29.8%	29.9%	-29.1%
Apache 2.0	25.2%	-25.2%	16.3%	-10.7%	8.2%	8.2%
IIS 4.0	20.3%	-20.3%	18.9%	-18.9%	60%	60%
IIS 5.0	17.4%	-17.1%	17.4%	-17%	47%	47%

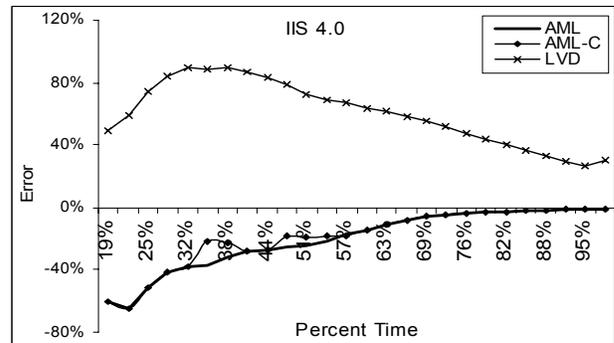


Figure 7. Accuracy of the models for estimating IIS 4.0 vulnerabilities data

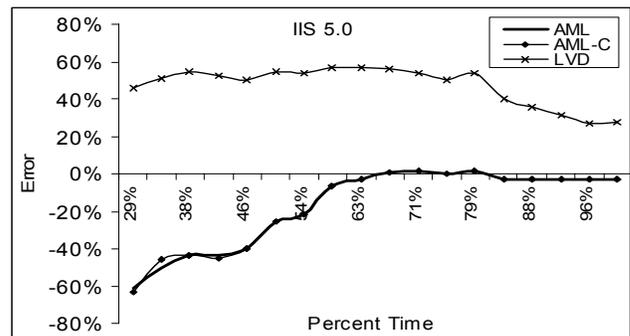


Figure 8. Accuracy of the models for estimating IIS 5.0 vulnerabilities data

Table 5. Average Error and Average Bias for Prediction After 65% of Time Elapsed

Model Server	AML		AML-C		LVD	
	AE	AB	AE	AB	AE	AB
Apache 1.0	8.4%	8.4%	6.2%	-6.2%	4.7%	-2.4%
Apache 2.0	11%	-11%	10.4%	-10.4%	4.4%	4.4%
IIS 4.0	3.2%	-3.2%	3.2%	-3.2%	41.2%	41.2%
IIS 5.0	2.1%	-1.4%	2.1%	-1.4%	41.8%	41.8%

## 4.2 Adaptive long term prediction accuracy test

Adaptive techniques or recalibration have been applied to software reliability growth models [23] in order to improve prediction. The approaches adjust the predictions by observing the errors of past estimations. This adaptive technique also referred to as recalibrating, was found to improve the accuracy of the estimations of software reliability growth models.

The main consideration in adaptive technique is finding the optimal size of prediction error to eliminate by subtraction or division. There is a trade-off in choosing a larger or a smaller error value used for correction. Larger error correction can give better results when there is a consistent bias in estimation; however, it may increase in error when the bias is unstable. With adaptive techniques the differences among the models tend to get reduced. Choosing to compensate for a

smaller error can better preserve the overall model characteristic; however, the improvement will be small.

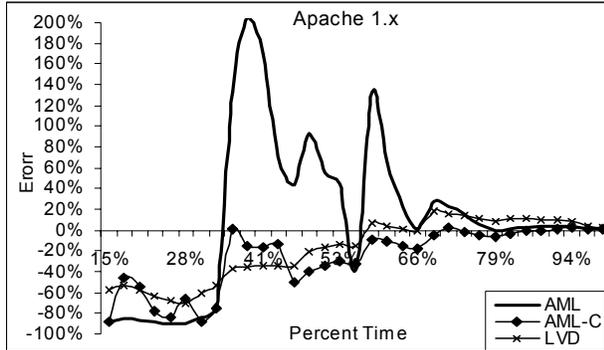


Figure 9. Accuracy of the adaptive models in estimating Apache 1.x vulnerabilities data

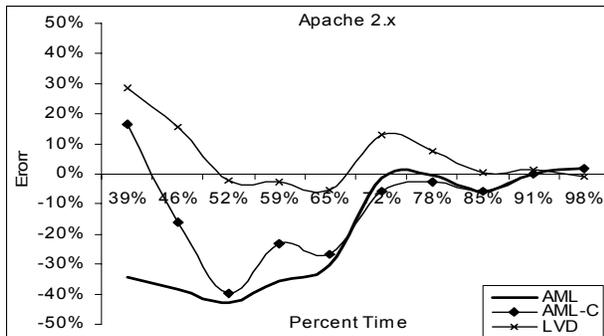


Figure 10. Accuracy of the adaptive models in estimating Apache 2.x vulnerabilities data

Figure 9 to Figure 12 show errors in estimations using adaptive techniques, they demonstrate significant improvement over the original estimations shown in Figure 5-Figure 8.

The recalibration was done by calculating the ratio between observed and estimated values for the past three quarters, and then dividing the next estimate by the ratio to adjust it. Recalibration can be done by using several alternative approaches; the effectiveness of recalibration can vary depending on the size of adjustments used.

The resulting plots shown in Figure 9 to Figure 12 and summarized in Tables 6 and 7, were they demonstrate some improvements over the non-adaptive approach's results.

Table 6. Average Error and Average Bias for Prediction of  $\Omega$  (Adaptive Estimation)

Model Server	AML		AML-C		LVD	
	AE	AB	AE	AB	AE	AB
Apache 1.0	56.1%	12.1%	27.2%	-12%	26.1%	-17.7%
Apache 2.0	19.1%	-26.1%	15.3%	-10%	10.5%	5.6%
IIS 4.0	16.1%	-15.6%	32.9%	-20%	50.8%	50.8%
IIS 5.0	16.9%	-14.7%	21.1%	0.1%	40.9%	40.9%

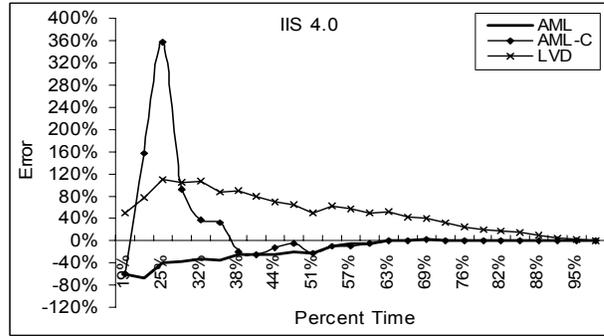


Figure 11. Accuracy of the adaptive models in estimating IIS 4.0 vulnerabilities data

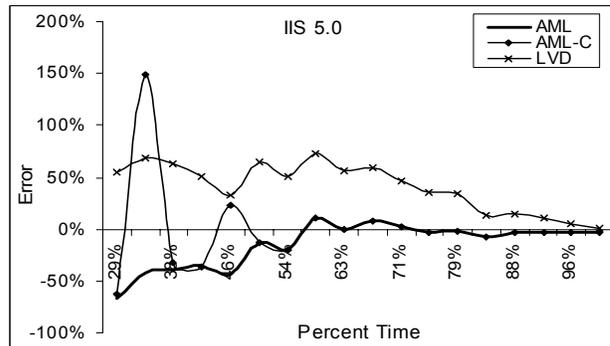


Figure 12. Accuracy of the adaptive models in estimating IIS 5.0 vulnerabilities data

Table 7. Average Error and Average Bias for Prediction of  $\Omega$  after use of 65% of the Data Is Used (Adaptive Estimation)

Model Server	AML		AML-C		LVD	
	AE	AB	AE	AB	AE	AB
Apache 1.0	6.8%	6.7%	3.7%	-2.4%	9.7%	9.7%
Apache 2.0	1.9%	-1.2%	8.2%	-2.5%	5.5%	4.3%
IIS 4.0	0.6%	0.3%	0.6%	0.3%	19%	19%
IIS 5.0	3.9%	-1.9%	3.9%	-1.9%	24.4%	24.4%

Table 7 above gives the error values when the projections are made after only 65% of the data has become available. The errors in general are small for AML and AML-C, ranging from 0.6% to 8.2% and from 5.5% to 24.4% for the LVD model.

## 5. Estimating Vulnerabilities for the Following Year

In the previous section, we have seen that the AML models and the LVD Model can be used to predict the total number of vulnerabilities with good accuracy especially when suitable methods are used to enhance the predictive capability. However, many applications such as risk assessments applications require the estimation of the vulnerability discovery rate for a shorter period of time in near future. It is therefore

essential to test the accuracy of the shorter term estimations so a practitioner can predict the number of vulnerabilities expected, say during the next year, and be aware of any limitations of the estimates made.

Similar to the methodology used for the long term projections, here the projection has been made at each of the  $n$  instants ( $t_1, t_2, t_3 \dots t_n$ ) with three months intervals. For each  $t_i$ , the partial data up to  $t_i$  has been used to fit the model using regression analysis to determine the best values for the parameters. The parameters are used to project the number of vulnerabilities  $\theta_i'$  expected to be discovered within the next year and then compared with the actual number of vulnerabilities ( $\theta_i$ ) to determine the estimation error ( $\theta_i - \theta_i'$ ). We then take the average of the error values to obtain the measure of the absolute average error ( $AAE$ ) and the absolute average bias ( $AAB$ ) which are defined as follows:

$$AAE = \frac{1}{n} \sum_{i=1}^n |\theta_i - \theta_i'| \quad (6)$$

$$AAB = \frac{1}{n} \sum_{i=1}^n \theta_i - \theta_i' \quad (7)$$

Note that it is not possible to normalize the  $AAE$  and  $AAB$  measures because in the later periods, the actual value can sometimes be zero, causing the divide by zero problem if normalization is used.

Next, the estimation approaches will be previewed and the predictive capabilities of these approaches will be evaluated. In this section we suggest an alteration to direct application of the models in order to improve the accuracy of prediction. Estimations can be done using direct model estimation, or using an adaptive approach.

### 5.1 Estimating the following year's vulnerabilities by direct model application

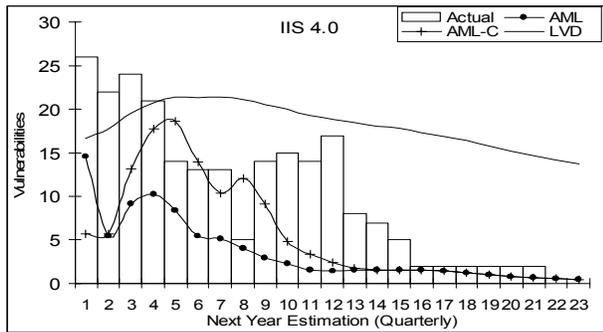


Figure 13. Estimations of vulnerabilities for the following year (quarterly)

In this approach, the prediction is done using a subset of the data to fit the model using the least square method, and then the actual number of vulnerabilities of the following year is compared to the predicted number of vulnerabilities.

The results are shown in Figures 13 to 16 and the average values are given in Table 8. AML and AML-C has shown to have the higher accuracy than LVD in IIS, while LVD has shown better accuracy for Apache 2, possibly because that Apache 2 dataset was very linear. For IIS 5 LVD accuracy is close to AML and AML-C. All of the models performed close to each other in terms of average error, although AML and AML-C did slightly better for the IIS data sets and LVD did slightly better for the Apache data sets. This difference in applicability of models can be explained by the fact that AML and AML-C model the saturation affect where as LVD may fits data well when saturation has not occurred.

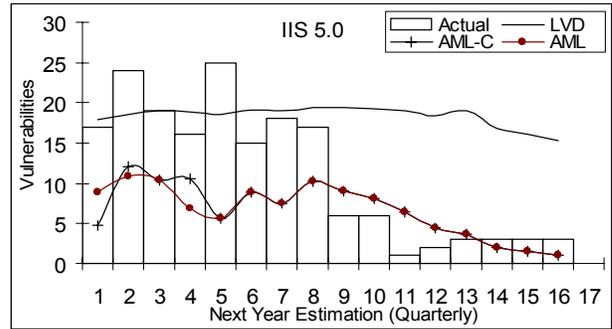


Figure 14. Estimations of vulnerabilities for the following year (quarterly)

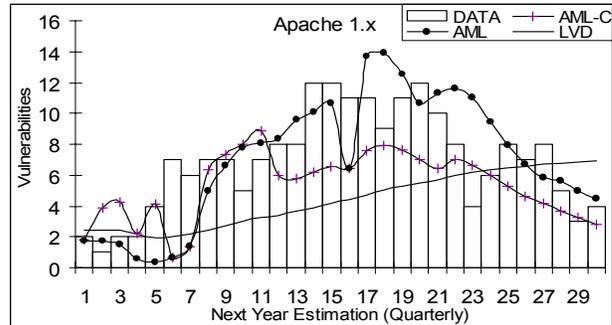


Figure 15. Estimations of vulnerabilities for the following year (quarterly)

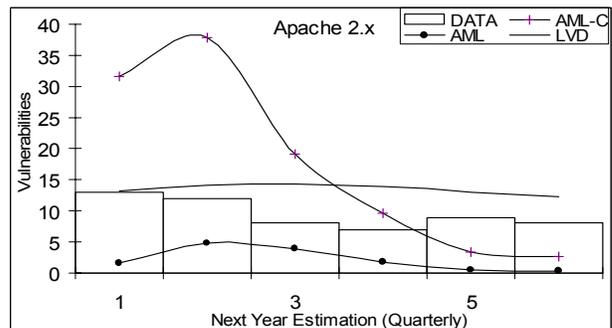


Figure 16. Estimations of vulnerabilities for the following year (quarterly)

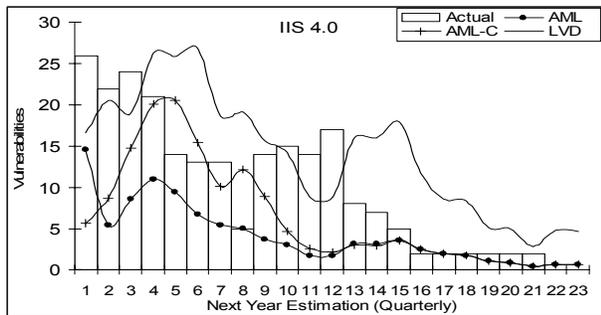
**Table 8. Absolute Average Error and Absolute Average Bias for Non-Adaptive Short Term Prediction (in vulnerabilities per year)**

Model Server	AML		AML-C		LVD	
	AAE	AAB	AAE	AAB	AAE	AAB
Apache 1.0	2.16	.11	2.48	-1.57	3.42	-2.51
Apache 2.0	7.32	-7.32	11.57	7.89	4.01	4.01
IIS 4.0	6.51	-6.43	5.57	-4.4	9.7	8.13
IIS 5.0	8.15	-1.97	7.15	-2.87	8.45	-7.13

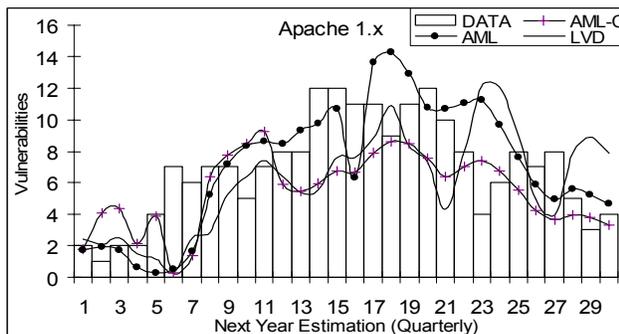
## 5.2 Estimating the following year's vulnerabilities using adaptation

Here the prediction takes place using a subset of the data to fit the model using least squared method similar to the direct approach; however, half of the error of previous estimations is subtracted from the estimation and a window of the past three errors is used.

Figure 17 and Figure 18 show the enhancements achieved by using adaptive techniques. The error was chosen to be the average of the past three errors for the LVD model. The same was used with AML and AML-C; however, only with the difference that the error size was divided by 2 to make it milder. Table 9 below shows improvements over The non-adaptive approach by showing smaller AAE and AAB numbers. Further improvements in finding an optimal choice of error size may yield better estimation accuracy.



**Figure 17. Adaptive estimations of vulnerabilities for the following year (quarterly)**



**Figure 18. Adaptive estimations of vulnerabilities for the following year (quarterly)**

**Table 9. Absolute Average Error and Average Bias for Adaptive Short Term Prediction (in vulnerabilities per year)**

Model Server	AML		AML-C		LVD	
	AAE	AAB	AAE	AAB	AAE	AAB
Apache 1.0	2.27	0.12	2.53	-1.4	3.09	-1.4
Apache 2.0	6.54	-6.54	7.31	11.59	3.76	3.76
IIS 4.0	5.98	-5.82	5.24	-3.69	6.57	3.96
IIS 5.0	8.30	-1.74	7.1	-2.42	8.05	6.18

## 6. Discussion

Goodness of fit results for the three-phase AML model show significant fit to all datasets. When using the model for long term predictions, adding a constraint to AML was found to be useful in filtering early extreme estimations. However, the constraint does not significantly improve the estimation of the number of vulnerabilities expected in the next year. Overall, the long term prediction for AML-C has proved to be the most accurate. Moreover, the accuracy of AML/AML-C improves significantly as more data is available, especially with about 65% or more of the data. Recalibration has improved the estimations for both long term prediction and short term prediction. However, in this study the recalibration adjustment was kept small because sometimes there was no consistent pattern in the bias and therefore larger adjustment would make the estimations unstable.

Goodness of fit results for LVD shows that LVD fits when the data set is less mature, suggesting that LVD can be better in short term prediction than long term prediction. However, results for LVD for the next year's estimation show that overall performance is close to AML and AML-C. For long term prediction LVD has larger Average Error; it demonstrates consistent bias, underestimation for the Apache data and overestimation for IIS. For the next year's prediction, heavier recalibration has improved LVD performance better than milder recalibration. The overall results suggest that recalibration improves the estimations for all models. The results are similar to those for software reliability growth models [22].

## 7. Conclusions

This paper examines the applicability of two vulnerability discovery models, AML and LVD, to the data for two separate versions of Apache and IIS each. The fit of the models was evaluated by computing goodness of fit for each case. The fit was always significant for the AML model. However for LVD, the fit was not significant in some of the cases.

The paper also examines the prediction capability which was tested for both long term predictions (total number of vulnerabilities) and for short term predictions

(number of vulnerabilities that will be discovered in the next year). For long term prediction, an adaptive technique was shown to improve estimations. After 65% of the data, the prediction error becomes small. The best results were observed with the AML-C approach which uses a more stable estimate of the parameter values. For short term prediction, the models were applied directly and using recalibration. Recalibration again showed some improvement over direct model application.

AML and LVD models can be integrated into the development process to create more secure software systems [24]. An approach recently proposed by Sahinoglu [25] needs such an assessment of the vulnerabilities for estimating risk and the cost of loss. Short term prediction can be used to evaluate the estimated vulnerability discovery rates which would become part of the risk evaluation.

The work can be extended to measure the accuracy of predicting vulnerabilities of a specific category or severity level, since the AML model has shown to fit such data [26].

Further work is needed to improve and optimize the adaptive technique, such as applying neural network approaches to it. Also, it may be possible to utilize the vulnerability density as a static measure to improve estimation accuracy by stabilizing some of the parameter values.

## Acknowledgment

We would like to acknowledge discussions with Sung-Whan Woo in the Computer Science Department at Colorado State University.

## References

- [1] R. Ford, H. Thompson, and F. Casteran, F. *Role comparison report—web server role*. Technical Report, Security Innovation, 2005.
- [2] E. E. Schultz, D. S. Brown, and T. A. Longstaff, *Responding to computer security incidents*. Lawrence Livermore National Laboratory (July 1990).
- [3] National Vulnerability Database. <http://nvd.nist.gov/>.
- [4] Mitre Corp, Common Vulnerabilities and Exposures, <http://www.cve.mitre.org/>.
- [5] Apache Software Foundation Bug System, <http://issues.apache.org/bugzilla/>.
- [6] Security focus, <http://www.securityfocus.com/>.
- [7] D. Moore, C. Shannon, and K. C. Claffy, Code-red: a case study on the spread and victims of an internet worm. In *Internet Measurement Workshop* (2002), pp. 273–284.
- [8] R. Anderson, Security in open versus closed systems—the dance of Boltzmann, Coase and Moore. In *Conf. on Open Source Software: Economics, Law and Policy* (2002), 1–15.
- [9] S. Brocklehurst, B. Littlewood, T. Olovsson, and E. Jonsson, On measurement of operational security. *Proc. 9<sup>th</sup> Annual IEEE Conference on Computer Assurance* (1994), 257–266.
- [10] J. Hallberg, A. Hanstad, and M. Peterson, A framework for system security assessment. *Proc. 2001 IEEE Symposium on Security and Privacy* (May 2001), 214–229.
- [11] H. K. Browne, W. A. Arbaugh, J. McHugh, and W. L. Fithen, A trend analysis of exploitations. In *IEEE Symposium on Security and Privacy* (2001), 214–229.
- [12] E. Rescorla, Security holes... who cares? *Proc. 12<sup>th</sup> USENIX Security Symposium* (2003), 75–90.
- [13] E. Rescorla, Is finding security holes a good idea? *IEEE Security and Privacy* 03, 1 (2005), 14–19.
- [14] O. H. Alhazmi and Y. K. Malaiya, Quantitative vulnerability assessment of system software. *Proc. Annual Reliability and Maintainability Symposium* (Jan. 2005), 615–620.
- [15] O. H. Alhazmi, Y. K. Malaiya, and I. Ray, Security vulnerabilities in software systems: A quantitative perspective. *Proc. Ann. IFIP WG11.3 Working Conference on Data and Information Security* (Aug. 2005), 281–294.
- [16] O. H. Alhazmi and Y. K. Malaiya, Modeling the vulnerability discovery process. *Proc. 16<sup>th</sup> International Symposium on Software Reliability Engineering* (Nov. 2005), 129–138.
- [17] J. Musa, *Software Reliability Engineering*. McGraw-Hill, 1999.
- [18] M. R. Lyu, *Handbook of Software Reliability*. McGraw-Hill, 1995.
- [19] A. Ozment and S. E. Schechter, Milk or Wine: Does Software Security Improve with Age? *Proc. 15<sup>th</sup> USENIX Security Symposium* (July-August 2006) pp. 93-104
- [20] O. H. Alhazmi and Y. K. Malaiya, Prediction capability of vulnerability discovery process. *Proc. Reliability and Maintainability Symposium* (Jan. 2006), 86–91.
- [21] Netcraft, <http://news.netcraft.com/>.
- [22] Y. K. Malaiya, N. Karunanithi, and P. Verma. "Predictability of software reliability models," *IEEE Transactions on Reliability*, Vol. 41, No. 4, December 1992, 539–546.
- [23] N. Li, Y. K. Malaiya, "Enhancing accuracy of software reliability prediction", *Proc. 4<sup>th</sup> International Symposium on Software Reliability Engineering* (Nov. 1993), 71–79.
- [24] R. Seacord, *Secure Coding in C and C++*. Addison Wesley, 2005.
- [25] M. Sahinoglu, Quantitative risk assessment for dependent vulnerabilities. *Proc. Reliability and Maintainability Symposium* (Jan. 2006), 82-85.
- [26] W. Sung-Whan, O. H. Alhazmi, and Y. K. Malaiya, Assessing Vulnerabilities in Apache and IIS HTTP Servers, *Proc. IEEE International Symposium on Dependable Autonomic and Secure Computing (DASC'06)*, (Sept.-Oct. 2006).