Proceedings of the 10th IASTED International Conference
SOFTWARE ENGINEERING AND APPLICATIONS
November 13–15, 2006, Dallas, TX, USA
ISBN Hardcopy: 0-88986-642-2 / CD: 0-88986-599-X

AN ANALYSIS OF THE VULNERABILITY DISCOVERY PROCESS IN WEB BROWSERS

Sung-Whan Woo, Omar H. Alhazmi and Yashwant K. Malaiya
Computer Science Department
Colorado State University, Fort Collins, CO, 80521 U.S.A
woo|omar|malaiya@cs.colostate.edu

ABSTRACT
New vulnerabilities discovered in a web browser put millions of users at risk, requiring urgent attention from developers to address these vulnerabilities. This paper presents a quantitative characterization of browser vulnerabilities which can be used to project the number of vulnerabilities to plan, test and development resources more efficiently. Vulnerability discovery data for the three major browsers, Internet Explore, Firefox and Mozilla, are examined and fitted to a vulnerability discovery model, and the goodness of fit is statistically examined. The results show that the datasets fit the model well, suggesting that this model can be used for making future projections. When the vulnerabilities are partitioned into categories based on their type, the data of individual categories also fit the model separately. When the vulnerabilities are partitioned into three severity levels, the model is found to be applicable to vulnerabilities with high and low severities. It is observed that the popularity of a browser itself leads to a higher discovery rate.

KEY WORDS
Vulnerability, Web Browser, Internet Explorer, Firefox, Mozilla

1. Introduction

The browsers serve as the clients' platform for several security-critical applications such as Internet banking, e-commerce and on-line trading. There has been growing concern about potential insecurity in web browsers due to vulnerabilities. While the vulnerabilities and exploits of Microsoft IE (Internet Explorer) have been frequently discussed [1][2], its alternatives are also not immune to serious vulnerability issues [3]. The existing studies on security vulnerabilities have been qualitative, focused on detection and prevention of vulnerabilities in web browsers. A number of security problems relating to the browsers are now being examined, such as spyware [4], phishing web page filtering [1][5], malicious pop-up windows, and e-commerce fraud. Many of these problems occur due to the presence of vulnerabilities in the browser software. Secure Science Corp. [6] reports a single phishing group collecting access information for 13,677 accounts in a single day by installing a malicious code through exploiting an unpatched vulnerability. Nimda, which used the buffer overflow vulnerability, affects all Windows versions of Microsoft Internet Explorer [7]. The exploitation techniques and tools utilized are no longer the exclusive possession of experts, since many such methods are now widely available and can be relatively easy to use.

Two of the major software components of the Internet are an HTTP (Hyper Text Transfer Protocol) server (also termed a web server) [8] and the browser. The first public version of IE was released in August 1995. Firefox and its predecessor Mozilla are based on Netscape Navigator, announced in October 1994, which emerged as the popular client web browser during the 1990s. IE currently has about 85% of the overall market share. However the popularity of Firefox has recently increased due to problems relating to IE vulnerabilities.

Some of the browser functions provide attackers or malicious users opportunities to exploit security holes since these processes require downloading, uploading and executing files. Browser vulnerabilities represent one of main sources of the spread of viruses or worms. A fraction of all software defects are security related and thus constitute vulnerabilities. In this paper we examine the vulnerability discovery rates for the three main web browsers and explore the applicability of a vulnerability discovery model to the aggregate vulnerability data as well as data partitioned by causes and severity.

Vulnerabilities are a special class of defects that can permit circumvention of security measures. Some vulnerability discovery models have recently been proposed by Anderson [9], Rescorla [10], and Alhazmi and Malaiya [11]. A comparison of proposed models was carried out by Alhazmi and Malaiya [12].

The next section introduces the vulnerability discovery model used and the significant factors that affect software vulnerability rates. We then consider the aggregate vulnerabilities in the three web browsers and examine how well the models fit the available data. The datasets are then partitioned into categories based on how such vulnerabilities arise the applicability of the models to individual categories is considered. Next, the vulnerabilities are divided according the severity of

impact and the fit provided by the model is again examined. Lastly, the major observations are discussed and the conclusion is presented.

2. Modeling the Vulnerability Discovery Process

Here we investigate the applicability of the Logistic Vulnerability Discovery model proposed by Alhazmi and Malaiya [13]. This model has 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 model considers calendar time as the independent variable and it incorporates the effect of the rising and declining market share on the software.

The Alhazmi-Malaiya Logistic model, referred to as the AML Model, assumes that the rate of change of the cumulative number of vulnerabilities is governed by two factors. The first factor declines as the number of remaining undetected vulnerabilities declines. The other factor increases with the time needed to take into account the rising share of the installed base. It gives the model as

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

where Ω is the cumulative number of vulnerabilities, t is the calendar time and A, B and C are empirical constants.

Equation 1 gives us a three-parameter model given by the logistic function; the equation shows that as t approaches infinity, y approaches B. Thus, the parameter B represents the total number of accumulated vulnerabilities that will eventually be found. In some cases when saturation has not set in, the available data corresponds to the linear part of the curve [11].

Many factors impact the vulnerability discovery rate. The three major factors, code size, software age and popularity, are examined below.

Code Size of Software: The studies show that the number of defects or errors increases as code size increases. A first order approximation assumes a linear relationship, which allows a measure defect density to be defined. Since vulnerabilities are a class of defects, we can similarly define a measure called vulnerability density [11]. Available data allows us to calculate the densities of the discovered vulnerabilities for two of the browsers, as given in Table 1.

Software Age: The software system passes through three different phases: the initial phase, termed learning phase, usually very few vulnerabilities are found. Next, as the knowledge about the software system increase, the rate of vulnerabilities discovery increases, this phase is the linear phase where a steady stream of vulnerability discoveries occurs. Finally, the third phase also called the saturation phase where the finding vulnerability discovery rate declines.

Market Share: A higher market share provides more incentive to explore and exploit vulnerabilities for both experts and non-experts, since both would find it more profitable or satisfying to spend their time on a software with a higher market share. The effect of the market share rise and fall is implicit in the AML model.

Table 1 present's data obtained the number of vulnerabilities from NVD [14] and the market share data from Net Applications [15] showing the current web browser market share and the total number of vulnerabilities found to date. The IE share ranges between 87% and 83%, while the Firefox share is between 12% and 8%. As we can see from the Table 1, for web browsers with a lower percentage of the market share, such as Mozilla and Safari, the total number of vulnerabilities found is low. This does not mean that these web browsers are more secure, but merely that only a limited effort has gone into finding their vulnerabilities. From the Table 1 we observe that the vulnerability discovery rate is related more to the market share than to the period of usage. Even though Mozilla was released earlier than Firefox and both source code sizes are similar, greater number of vulnerabilities has been found in Firefox than Mozilla because of the greater popularity of Firefox.

Table 1. Browsers' Market Share and Vulnerabilities

Web Server	IE	Firefox	Mozilla	Safari
Market Share	84.7%	10.05%	0.34%	3.19%
Vulnerabilities	286	134	39	30
Release Date	Aug 1995	Sep 2002	Dec 1998	Jan 2003
Latest Version	6.0	1.5.0.1	1.7.12	2.0.3
Vul. Density	N/A	0.0378	0.014	N/A

Table 1 gives of values for vulnerability density [11] for Firefox and Mozilla per thousand lines of source code. The code size for both was determined using the SLOCCount tool [16]. Since the source code size for IE is not available, its vulnerability density cannot be evaluated.

3. Aggregate Vulnerabilities in Browsers

In this section we use the data for the three major web browsers, IE, Firefox and Mozilla, and determine whether the vulnerability discovery trends are described by the AML model.

IE controls about 85% of the Internet browser market. This high market share has made it an attractive target for exploration and exploitation by malicious users. The problem is exacerbated by the integration of IE into Windows, unlike Firefox or Mozilla. IE integration

provides several benefits, such as faster start up and easier interface with other components of windows. However security analysts and experts consider the integration of IE to be a security disadvantage since IE connects with a variety of Windows core components. Another weakness of IE is the use of non-standard features, which do not follow the W3C standard. Even though IE is known for its many security flaws, numerous Internet users still prefer to use IE because many web sites are optimized for IE; moreover, Windows software is marketed with IE pre-installed.

Although Firefox was released in September 2002, it did not gain significant recognition until 2004; its popularity has increased because of its perceived better security, intuitive design and multi-tap features. Currently Firefox is more common in school or public computers and is expanding its market share. However, its popularity has led to a rising number of newly discovered vulnerabilities.

Figure 1 shows the cumulative vulnerabilities by month and the fitted AML model for IE, Firefox and Mozilla. The bold black line indicates the fitted AML model, while the other line shows the cumulative number of vulnerabilities by months. The AML model fit the data for IE and Firefox very well. At the beginning, the slope of the curve for IE rose gently until 2000, after which the slope has generally remained steady

From the point of the three phases of the vulnerability discovery process, IE currently still appears to be in the linear phase, since the number of vulnerabilities is growing linearly in spite of IE's having been on the market for several years. This may be because of its larger market share and possibly because it may have a higher number of potential vulnerabilities. This suggests that vulnerability discovery for IE may continue at a significant pace in the near future. It is expected that the next release of IE will have more security focus.

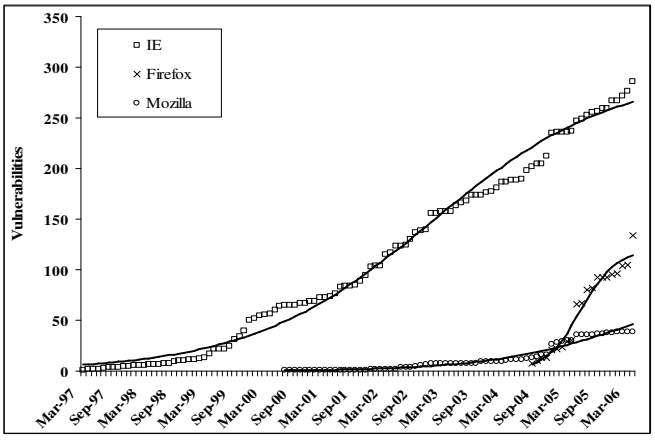

Figure 1. IE Vulnerabilities

Firefox is currently the second most popular web browser. The considerable market share gap between IE and Firefox is shrinking. Although Firefox is just four years old, and its market share is about one-eighth of the market, the fitted model suggests that Firefox is still in the linear phase. Consequently, we can expect that while more vulnerabilities will be found in the near future, the saturation phase is not likely to be reached soon.

Mozilla was first released at the end of 1998. However, since Mozilla never became very popular among Internet users, very few vulnerabilities have been found in Mozilla even though it was developed long before Firefox. Only 11 vulnerabilities were found through June 2004. We observe that the discovery rate of Firefox and Mozilla vulnerabilities suddenly increased in the later part of 2004. A large number of vulnerabilities were first found in Firefox, followed by a similar rise in the discovery of Mozilla vulnerabilities. This is likely to be due to the fact that significant parts of code are shared between Firefox and Mozilla, demonstrating that market share can be a more important contributing factor than software age. Moreover, Over 50% of Mozilla's vulnerabilities are vulnerabilities shared with Firefox, in fact most of the vulnerabilities found in Mozilla after October 2004 were actually found in the shared code. The popularity of Firefox has increased markedly at that time. Figure 1 suggests superimposition to two s-shapes, one due to vulnerabilities found in Mozilla itself and the second due to vulnerabilities found in Firefox.

Table 2 shows parameter values obtained by fitting the models used in Figure 1. For chi-squared goodness of fit test, we chose an alpha level 5%. Table 2 gives each browser's chi-square values, R^2 values and parameter values for the AML model. When the P-value value is close to 1, the model data fit is significant; moreover, R^2 close to 1 indicates strong correlation between the model and actual data. The table shows that the chi-square values are less than the critical value except for IE. P-values for Firefox and Mozilla are in the acceptable range. The chi-square value and P-value for IE are not significant with respect to the level chosen; however, the R^2 value is very close to 1, indicating a strong correlation between the model and actual data.

Table 2. χ^2 Goodness of Fit for Total Vulnerabilities

Browser	*A*	*B*	*C*	R^2	χ^2	$\chi^2_{critical}$	*P-value*
IE	.0002	295	.164	.9887	174.1	135.5	7.43E-05
Firefox	.0024	119.8	.190	.9570	20.8	32.6	.4055
Mozilla	.0007	95.9	1.53	.9572	32.078	90.531	.9999

4. Individual Vulnerability Categories

In this and the following sections we apply the AML model to two separate classification schemes for web browser vulnerabilities. Distinguishing among vulnerabilities is useful when we want to examine the nature and extent of the problem, as well as being helpful in determining what corrective actions would be most effective. A vulnerability taxonomy classifies the

vulnerabilities according to some significant characteristics. Several taxonomies have been proposed [17][18][19][20][21]. An ideal taxonomy should have such desirable properties as mutual exclusiveness, clear and unique definition, and coverage of all software vulnerabilities.

Vulnerabilities can be classified using schemes based on cause, severity, impact and source, etc. In this analysis, we use the classification scheme employed by the National Vulnerability Database of the National Institute of Standards and Technology. This classification is based on the causes of vulnerability. The eight classes are as follows [14][17]:

1. Input Validation Error (includes: boundary condition error, buffer overflow).
2. Access Validation Error
3. Exceptional Condition Error
4. Environmental Error
5. Configuration Error
6. Race Condition Error
7. Design Error
8. Others

Unfortunately, these eight classes are not completely mutually exclusive. Table 3 shows how vulnerabilities are distributed among categories for both the datasets studied. We define these vulnerabilities as overlap vulnerabilities. IE has 25 overlap vulnerabilities, Firefox has 12 and Mozilla has 3. In Table 3 three web browsers show the same pattern —i.e., that the number of design errors is much higher than other types of vulnerabilities, followed by input validation error. More than 60% of found vulnerabilities are related to design or input validation errors. When comparing web browsers to web servers (Apache and IIS) and operating systems (Windows 2000 and XP), we find a comparable pattern. Usually web servers and operating systems have a greater number of vulnerabilities in input validation error than in design error. Apart from these two categories, other classes show a similar priority order (exceptional condition error, access validation error, configuration error and other classified errors).

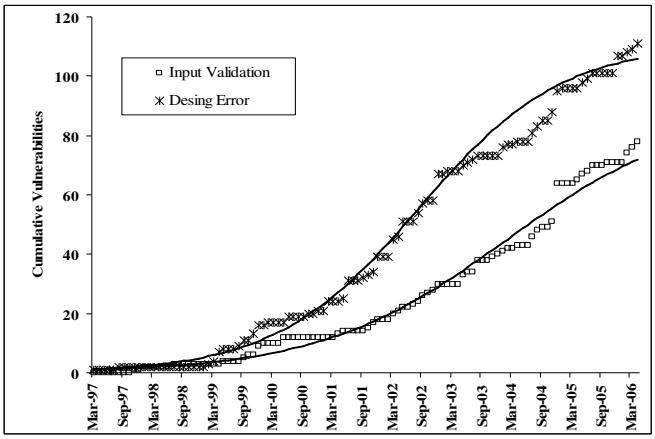

Figure 2. IE Vulnerabilities by Category

Figures 2 and 3 present the AML model fitting of each web browser's vulnerabilities by category. Here, we consider only the two major categories, design errors and input validation errors, since other categories have too small a number of vulnerabilities to fit to the model. In these three figures, the bold lines indicate the fitted AML model for each category, while the dotted lines and thin lines indicate cumulative vulnerability data for each category.

Table 3. Vulnerabilities Classified by Categories

	Total	I V	DE	CE	AVE	ECE	EE	RCE	Other
IE	286	78 27%	111 39%	11 4%	38 13%	41 14%	6 2%	3 1%	23 8%
Firefox	134	40 30%	68 51%	2 2%	14 10%	8 6%	0 0%	0 0%	14 10%
Mozilla	39	16 41%	17 43%	1 3%	3 8%	3 8%	0 0%	1 3%	1 3%

Figure 2 shows the AML model fitting for categorized IE vulnerabilities. From the beginning to the present, the AML model and the cumulative data demonstrate that design errors have been found more frequently than input validation errors, and that the gap between design error and input validation error is widening.

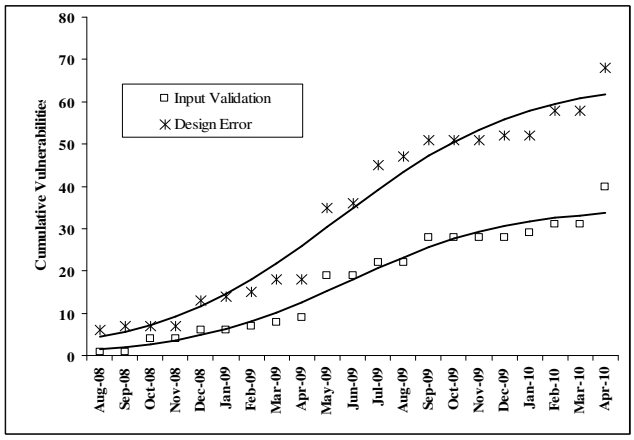

Figure 3. Firefox Vulnerabilities by Category

Figure 3 shows the AML model fitting for categorized Firefox vulnerabilities. Categorized Firefox vulnerabilities demonstrate a pattern similar to that of categorized IE vulnerabilities. Design errors have been found more frequently than input validation errors, and the gap between design error and input validation error is widening.

Table 4 shows the chi-square goodness of fit tests for IE, Firefox and Mozilla models by category. For each category, the χ^2 value is less than χ^2 Critical, and the P-values and R^2 values are close to 1. Thus, the fit for the two categories is significant for the AML model. It is interesting to note that the fit for the aggregate IE vulnerabilities considered in the previous section was not significant with respect to the significance level chosen.

We applied the AML model to two major vulnerability categories to determine whether there are observable patterns at the level of individual classes. Since we noted a similar pattern for the uncategorized vulnerabilities, a possible fit was examined. These individual classes reflect each web browsers' own total number of vulnerabilities.

Table 4. χ^2 Goodness of Fit for Total Vulnerabilities

Browser	Type	*A*	*B*	*C*	R^2	χ^2	$\chi^2_{critical}$	*P-value*
IE	**IVE**	.00059	89.7	.984	.98	43.9	135.4	.99
	DE	.00062	110.9	.895	.99	47.2	135.4	.99
Firefox	**IVE**	.0089	35	.849	.962	6.5	32.6	.99
	DE	.0042	65.1	.278	.96	9.1	32.6	.98

5. Vulnerability Severity Levels

The severity of a vulnerability indicates how serious the impact of an exploitation can be. Vulnerabilities are usually divided into three severity categories: high, medium and low. The National Vulnerability Database (NVD) of the National Institute of Standards and Technology utilizes the CVSS metric, which uses many factors to assign a value ranging from 1 to 10 to each vulnerability. The NVD classification defines the range 1-3.99 as low severity, 4-6.99 as medium severity, and 7-10 as high severity. NVD describes these severity levels as follows[14]:

1. High Severity: "This makes it possible for a remote attacker to violate the security protection of a system (i.e., gain some sort of user, root or application account), or permits a local attack that gains complete control of a system, or if it is important enough to have an associated CERT/CC advisory or US-CERT alert."
2. Medium Severity: "This does not meet the definition of either "high" or "low" severity."
3. Low Severity: "The vulnerability typically does not yield valuable information or control over a system but rather gives the attacker information that may help him find and exploit other vulnerabilities or otherwise determine the vulnerability is inconsequential for most organizations."

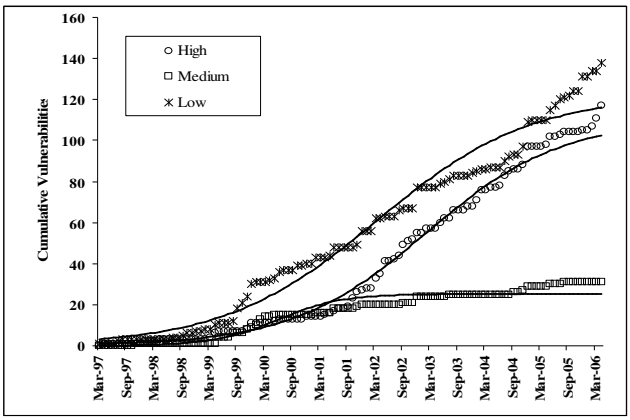

Figure 4. IE Vulnerabilities by Severity

The AML model is applied to the three web browsers. Figures 4 and 5 show the results of fitting the AML model to the three severity levels. With the exception of the low severity IE vulnerabilities, the fit is significant for all the cases. Even for the low severity IE vulnerabilities, the R^2 value is quite high.

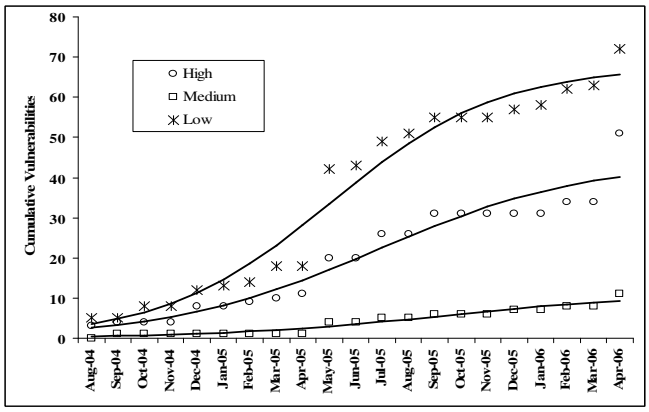

Figure 5. Firefox Vulnerabilities by Severity

Table 5. χ^2 Goodness of Fit for Total Vulnerabilities

Browser	Severity	*A*	*B*	*C*	R^2	χ^2	$\chi^2_{critical}$	*P-value*
IE	**High**	.0006	110	1.25	.99	40.2	135.4	1
	Med.	.0057	25.1	12.4	.92	46.3	135.4	.99
	Low	.0004	122.2	.325	.96	107.4	135.4	.12
Firefox	**High**	.0057	44.1	.447	.92	8.83	32.6	.98
	Med.	.0206	11.3	2.56	.93	4.02	32.6	.99
	Low	.0046	67.6	.36	.96	12.1	32.6	.91

6. Discussion

We have examined the vulnerability data sets to determine whether the discovery process tends to follow specific patterns and whether these patterns can be modeled. The results show that when the aggregate number of vulnerabilities is examined, the AML model fit the datasets well, as shown in Table 2. The model was found to fit even when vulnerabilities are partitioned by cause or severity levels (Table 4 and Table 5). This suggests that the models can potentially be used to estimate the number of vulnerabilities expected to be discovered in a given future period, and what category and severity level distributions are likely.

We note that there is sufficient data for high and low severity vulnerabilities, and the fit is quite good. This suggests that the model can be used to project the expected number of high severity vulnerabilities, which may be of much greater interest than others.

Examining the current vulnerability discovery trends for the three web browsers, all three appear to be in the linear phase. Hence, it can be expected that more vulnerabilities will be discovered in all three.

When comparing browser security, we need to keep in mind that the vulnerability discovery rate in the near future may be more important than vulnerabilities already discovered in the past. Other factors to consider include severity levels and quick availability of patch releases. Currently, certain experts [22] regard IE to be less secure; some of them point to the integration of IE into Windows and Active X. Secunia [23] reports that IE has more unpatched vulnerabilities than Firefox. However, if Firefox's popularity continues to increase, it will attract more attempts to discover its vulnerabilities. The new version of IE7, currently in beta version, is intended to be more secure because of incorporation of additional security features.

7. Conclusion

The results show that the AML vulnerability discovery model generally tracks the available data well. We found that the fit is significant when aggregate vulnerabilities are divided into classes (for example, vulnerabilities arising due to design errors and input validation errors), provided there are sufficient vulnerabilities in a class. The model can thus be used to project the classes of vulnerabilities that are more likely to be encountered, and consequently can be used to make testing more effective. The model also fits the data for high and low severity vulnerabilities. Hence, it is possible to project the high severity vulnerabilities that may be expected in the near future.

The results indicate that the models originally proposed and validated for operating systems are also applicable to web browsers. These models can be used to estimate vulnerabilities discovery rates, which can be integrated with risk assessment models in the future. Furthermore, these models can be integrated into the development process to create more secure software systems [24].

References

[1] A. Kumar, *Phishing - a new age weapon.* Tech. rep., Open Web Application Security Project (OWASP), 2005.

[2] E. E. Schultz, Internet Explorer Security: is there any hope? *Network Security, Vol. 2005, issue 1,1* (Jan. 2005), pp 6-10.

[3] R. Naraine, *Zero-day firefox exploit sends mozilla scrambling.* http://www.eweek.com/, May 2005.

[4] A. Moshchuk, T. Bragin,, , S. D. Gribble , and H. M. Levy, *A crawler-based study of spyware on the web. The 13th Annual Network & Distributed System Security Symposium* (2006).

[5] W. Dormann, and J. Rafail, *Securing your web browser.* Tech. rep., CERT Coordination Center, 2006.

[6] B. Krebs, *Real world impact of IE flaw, Brian Krebs on Computer Security,* http://blog.washingtonpost.com /securityfix/2006/04/real_world_impact_of_internet_1.html, Posted April 3, 2006.

[7] CERT Advisory CA-2001-26 Nimda Worm. http://www.cert.org/advisories/CA-2001-26.html, April 2006.

[8] S.W. Woo, O.H. Alhazmi and Y.K. Malaiya, Assessing Vulnerabilities in Apache and IIS HTTP Servers, *to appear in the Proc. the 2nd IEEE International Symposium on Dependable Autonomic and Secure Computing,* (Sep 2006).

[9] 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), pp. 1–15.

[10] E. Rescorla, *Is finding security holes a good idea? IEEE Security and Privacy 03*, 1 (2005), 14–19.

[11] O. H. Alhazmi and Y. K. Malaiya, "Quantitative Vulnerability Assessment of Systems Software," Proceedings of 51st Reliability and Maintainability Symposium, Alexandria, Virginia, U.S.A, January 23-27 2005.pp. 615-621.

[12] O. H. Alhazmi, and Y. K. Malaiya, *Modeling the vulnerability discovery process. Proc. 16th International Symposium on Software Reliability Engineering* (Nov. 2005), pp. 129–138.

[13] O. H. Alhazmi and Y. K. Malaiya, "Prediction Capabilities of Vulnerability Discovery Models," Proceedings of 52nd Reliability and Maintainability Symposium, Newport Beach, California, January 23-26 2006. pp.86-91.

[14] National Vulnerability Database, http://nvd.nist.gov/, April 2006.

[15] Net Applications, http://www.netapplications.com/, April 2006.

[16] SLOCCount,http://www.dwheeler.com/sloccount/,October2006

[17] T. Aslam, I. Krsul, and E. Spafford, "Use of a taxonomy of security faults," in *Proc. 19th NIST-NCSC Nat. Information Systems Security Conf.,* 1996, pp. 551–560.

[18] M. Bishop, *Vulnerability analysis. Proc. IInd Int. Symp. on Recent Advances in Intrusion Detection* (Sep 1999), pp. 125-136.

[19] R. Gopalakrishna, E. Spafford, and J. Vitek, *Vulnerability likelihood: a probabilistic approach to software assurance.* Tech rep., CERIAS, 2005

[20] C. E. Landwehr, A. R. Bull, J. P. McDermott, and W. S. Choi, *A taxonomy of computer program security flaws. ACM Comput. Surv. 26*, 3 (1994), pp. 211–254.

[21] R. C. Seacord, and A. D. Householder, Structured approach to classifying vulnerabilities. Tech rep., CMU/SEI-2005-TN-003, Carnegie Mellon, 2005.

[22] E. E. Schultz, D. S. Brown, and T. Longstaff *A. Responding to computer security incidents.* Tech rep., Lawrence Livemore National Laboratory, July 1990.

[23] Secunia, http://secunia.com/, April 2006.

[24] Seacord, R. C. *Secure Coding in C and C++*. Addison Wisely, 2005.