Empirical Estimates of 0Day Vulnerabilities in Control Systems

Wayne Boyer, Sean McBride, Miles A. McQueen
Idaho National Laboratory
Idaho Falls, Idaho
{wayne.boyer, sean.mcbride, miles.mcqueen}@inl.gov

Trevor A. McQueen
Harvey Mudd College
Claremont, California
trevor_mcqueen@hmc.edu

Abstract: We define a 0Day vulnerability to be any vulnerability, in deployed software, which has been discovered by at least one person but has not yet been publicly announced or patched. These 0Day vulnerabilities are of particular interest when assessing the risk to well managed control systems which have already effectively mitigated the publicly known vulnerabilities. In these well managed systems the risk contribution from 0Days will have proportionally increased. To aid understanding of how great a risk 0Days may pose to control systems, an estimate of how many are in existence is needed.

Consequently, using the 0Day definition given above, we developed and applied a method for estimating how many 0Day vulnerabilities are in existence on any given day. The estimate is made by: empirically characterizing the distribution of the lifespans, measured in days, of 0Day vulnerabilities; determining the number of vulnerabilities publicly announced each day; and applying a novel method for estimating the number of 0Day vulnerabilities in existence on any given day using the number of vulnerabilities publicly announced each day and the previously derived distribution of 0Day lifespans.

The method was first applied to a general set of software applications by analyzing the 0Day lifespans of 491 software vulnerabilities and using the daily rate of vulnerability announcements in the National Vulnerability Database. This led to a conservative estimate that in the worst year there were, on average, 2,500 0Day software related vulnerabilities in existence on any given day.

Using a smaller but intriguing set of 15 0Day software vulnerability lifespans representing the actual time from discovery to public disclosure, we then made a more aggressive estimate. In this case, we estimated that in the worst year there were, on average, 4,500 0Day software vulnerabilities in existence on any given day.

We then proceeded to identify the subset of software applications likely to be used in some control systems, analyzed the associated subset of vulnerabilities, and characterized their lifespans. Using the previously developed method of analysis, we very conservatively estimated 250 control system related 0Day vulnerabilities in existence on

any given day. While reasonable, this first order estimate for control systems is probably far more conservative than those made for general software systems since the estimate did not include vulnerabilities unique to control system specific components.

These control system specific vulnerabilities were unable to be included in the estimate for a variety of reasons with the most problematic being that the public announcement of unique control system vulnerabilities is very sparse. Consequently, with the intent to improve the above 0Day estimate for control systems, we first identified the additional, unique to control systems, vulnerability estimation constraints and then investigated new mechanisms which may be useful for estimating the number of unique 0Day software vulnerabilities found in control system components.

To make a 0Day estimate for control system specific devices and applications requires a modified approach for a variety of reasons including that vulnerability research in control systems is being performed by a smaller and apparently different set of security researchers, the control system specific applications are not as pervasive, and up to this time the vulnerabilities are generally not being reported publicly. Given these and other constraints, we proceeded to identify a number of new mechanisms and approaches for estimating and incorporating control system specific vulnerabilities into an improved 0Day estimation method. These new mechanisms and approaches appear promising and will be more rigorously evaluated during the course of the next year.

Keywords: 0Day, Vulnerability Disclosure, Security Metrics, CVSS

1 Introduction

This paper extends our previous work [13] and includes 0Day vulnerability models and estimates for control systems. Risk to a system is the probability of a negative event times the consequence of that event, summed over all possible events. Risk is defined in the National Infrastructure Protection Plan [1] as a function of consequence, vulnerability, and threat. One category of vulnerabilities within a system consists of those that reside in software. Within this category there are the questions of how many vulnerabilities exist, how well they are known, how easy they are to exploit, the resulting privileges that could be gained, and the damage that could ensue. This paper focuses specifically on the first two aspects of vulnerabilities and makes a first order estimate of the total number of 0Day software vulnerabilities that exist at any given time. We anticipate that with additional work this estimate will be able to be tailored for supporting an estimation of the exposure of individual systems from these relatively unknown vulnerabilities.

A vulnerability in software is "an instance of a mistake in the specification, development, or configuration of software such that its execution can violate the explicit or implicit security policy" [2]. The rate of vulnerability reporting for individual software applications has been studied for a number of years [3,4].

For software, in total, the number of publicly announced vulnerabilities has changed over the past 8 years from less than 3 per day in 2000 to over 16 per day in 2008 according to the National Vulnerability Database (NVD) [5]. This high rate of

vulnerability announcements has focused attention on the very practical and immediate issues of patch management [6], vulnerability disclosure processes [7,8], and speed of patch generation [9], dissemination and application.

Unfortunately, this focus has left unattended important issues that should be considered when pondering the security of a system consisting of software as well as hardware. One such issue is the question of how many software vulnerabilities are in existence which have been discovered by potential adversaries, but not yet publicly announced or patched (i.e. 0Day vulnerabilities). This paper's primary contribution is that it proposes and applies a novel method for making first order estimates of the number of 0Day vulnerabilities in existence on any given day.

There is no generally accepted formal definition for 0Day (also known as zero-day) vulnerability. The term has been used to refer to flaws in software that no one knows about except the attacker. Sometimes the term is used to mean a vulnerability for which no patch is yet available. For the purposes of this paper, we formally define a 0Day vulnerability as any vulnerability, in deployed software, that has been discovered by at least one person but has not yet been publicly announced or patched. These 0Day vulnerabilities are of particular interest in well managed systems which have effectively mitigated the publicly known vulnerabilities. In these well managed systems the risk contribution from 0Days will have proportionally increased. To aid understanding of how great a risk the 0Days may pose to a system, an estimate of how many are in existence is needed.

Using the 0Day definition given above, we developed and applied a method for estimating how many 0Day vulnerabilities are in existence on any given day. The estimate is made by: empirically characterizing the distribution of the lifespans, measured in days, of 0Day vulnerabilities; determining the number of vulnerabilities publicly announced each day; and applying a novel method for estimating the number of 0Day vulnerabilities in existence on any given day using the number of vulnerabilities publicly announced each day and the previously derived distribution of 0Day lifespans.

In this paper we first make use of 491 vulnerabilities, using the time they were privately reported to a vendor until their public announcement as a conservative stand-in for their lifespan as 0Days. After characterizing these lifespans we proceed to estimate how many 0Days existed on each given day in the past. Separate estimates are made for vulnerabilities in general and for the subset of vulnerabilities that apply to control systems.

We then make a more aggressive estimate of 0Day lifespans by discarding the 491 vulnerabilities mentioned above, and using a small set of 15 0Day vulnerabilities for which we knew the actual date of discovery along with the date of public announcement. As expected, the lifespans of these 15 vulnerabilities were, on average, quite a bit longer than those of the previous 491 vulnerabilities. Consequently, the new aggressive estimate of how many 0Days exist on each day in the past is significantly higher.

We then present lifespan data for vulnerabilities associated with software developed specifically for control systems. We show that for a set of 45 control system specific vulnerabilities that have either been reported to CERT/CC or have been publicly

disclosed, the lifespans are quite similar to the lifespans of general IT vulnerabilities. We identify several new mechanisms and approaches for estimating the number of control system specific 0Day vulnerabilities in existence.

Finally, given the estimations for the number of 0Day vulnerabilities and their lifespans, we looked at whether the risk to a system from 0Days might be less than the raw number estimates would indicate. We tentatively discovered that the more serious the vulnerability the longer its lifespan tended to be in both general enterprise level applications and those applications used in control systems. Thus the risk to systems appears to actually be greater than the 0Day estimates would indicate.

The rest of this paper is organized as follows. Section 2 discusses a vulnerability's lifecycle in the context of enterprise systems, and notes the variations due to differences which eventually should be accounted for in models for control systems. This provides context for the discussion of 0Day vulnerabilities. Section 3 characterizes the 0Day lifespan of 491 vulnerabilities. Section 4 describes and applies a method for estimating the number of 0Day vulnerabilities in existence for each day in the past. Section 5 is a recalculation of the number of 0Day vulnerabilities in existence making use of a small set of 15 0Day vulnerabilities. Section 6 is a discussion of control system specific vulnerabilities. Section 7 is a discussion of a potential relationship between 0Day lifespans and their Common Vulnerability Scoring System (CVSS) base scores [10]. Sections 8 and 9 present our conclusions and discuss future work.

2 0Day Vulnerability Lifecycle

The lifecycle of a vulnerability begins when it is initially created during software development and exists until a patch is released and applied, or the software is retired. A 0Day vulnerability only exists during the part of the lifecycle which starts with discovery of the vulnerability and ends with it either being patched, the software retired, or it being publicly announced. Note that our basic model of the lifecycle of a vulnerability is more fully described in [13].

2.1 0Day Lifespans

By our definition, a 0Day vulnerability is one which has been discovered but has not yet been publicly announced or fixed. This represents the real lifespan of the 0Day. Unfortunately, the actual date of vulnerability discovery is not usually available. The NVD database has "discovery date" records for a small fraction of its entries, and for many of those entries the "discovery date" is the same or nearly the same date as the "publication date". For those cases it is highly unlikely that the recorded "discovery date" is a valid start date for a 0Day vulnerability lifespan. Consequently, in Sections 3 and 4 of this paper, we use the time between reporting a 0Day vulnerability to the vendor and its public announcement as a conservative estimate of the actual 0Day's lifespan. In Section 5 we make a potentially more realistic estimate of the lifespan using 15 vulnerability data points where the actual discovery date was known.

2.2 Patching

For enterprise systems it is a reasonable first order conservative estimate to assert that once a patch has become available it is installed fairly quickly across many systems. For critical infrastructure control systems this assumption may be much less valid since once a patch is available the operators will quite likely follow a cautious approach to installation. One sample process that operators might follow is to install on a shadow system; prove the patch reliably installs on less critical portions of the control system and is validated to not cause reliability issues; and then consider installing on the most critical portions of the control system. Interestingly, this process might leave the most important elements of the control system unpatched for the longest periods of time. We have just begun an effort to determine the actual patch installation process used by owners within a variety of critical infrastructure sectors. For the moment we have chosen to still make use of the assumption that once a patch is available it is installed fairly swiftly on control systems. In practice this just makes our estimate of the number of 0Day vulnerabilities even more conservative.

2.3 Public Announcement

For enterprise systems there is a fairly well defined and reasonably accepted process for responsible vulnerability disclosure. As a consequence, many vulnerabilities are only publicly announced once the vendor has created a viable patch. Unfortunately, the current state of control system vulnerability discovery and disclosure appears less well formed and is much more problematic. We would expect the responsible disclosure process for control systems to eventually shake out in a fashion approximately the same as for enterprise systems except that due to patch installation timelines there may be an attempted intermediate step of disclosure to owner operators only, in order to provide greater time to patch before full public announcement. For this paper we decided to generally use the enterprise responsible disclosure process as our model since a reasonable and accepted alternative model is not yet available for control system vulnerabilities.

3 Estimate of 0Day Vulnerability Lifespans

In the 0Day vulnerability lifecycle discussed in the previous section, the time from vendor notification of the vulnerability to its public announcement is but a portion of the overall lifetime of a 0Day vulnerability and is referred to in Sections 3 and 4 of this paper as its lifespan. This section will describe our work in characterizing these 0Day lifespans.

3.1 Sources of 0Day Lifespan Information

In August, 2005 TippingPoint formed the Zero Day Initiative (ZDI). In TippingPoint's own words "The main goals of the ZDI are to:

- Extend our DVLabs research team by leveraging the methodologies, expertise, and time of others
- Encourage the reporting of zero day vulnerabilities responsibly to the affected vendors by financially rewarding researchers
- Protect our customers through the TippingPoint Intrusion Prevention Systems(IPS) while the affected vendor is working on a patch".

The second bullet is what is of most interest to us since it led ZDI to offering cash incentives to security researchers for the reporting of 0Day vulnerabilities to ZDI.

The process followed by ZDI when offered a 0Day vulnerability is to validate the vulnerability, attempt to negotiate a deal with the researcher, and, if a deal is reached, report the vulnerability to the vendor. When the vendor has developed a patch there is a coordinated public announcement about the vulnerability. The vulnerability details and the disclosure time line are then posted online. The difference between the report to vendor date and the public announcement date found in the disclosure time line may be used as an estimate of the lifespan of the 0Day vulnerability.

In May 1998 iDefense Labs (iDefense) was founded. Their business is to provide clients with leading edge intelligence on vulnerabilities and threats. iDefense efforts include the Vulnerability Contributor Program which is used to acquire 0Day vulnerabilities. They post online a list of all of their acquired 0Day vulnerabilities that have been made public. For each vulnerability the posting includes a description and analysis along with a disclosure timeline. The iDefense disclosure timeline includes the date the vulnerability was reported to the vendor and the date it was announced to the public. These two dates are the same information captured by ZDI so may also be used as an estimate of the lifespan of the 0Day vulnerability. While other dates, such as the date in which the vulnerability was reported to ZDI or iDefense, are occasionally available for the 0Day vulnerabilities, we used the date reported to vendor and the date of announcement to the public in order to be consistent.

Using these two data sources, we collected vulnerability postings from January 5, 2006 through August 16, 2007 and analyzed the lifespans of the 309 0Day vulnerabilities. Due to higher priority commitments we were then forced to set this research aside until May 22 of 2008. At that time we reconstituted our work and collected vulnerability postings from the two data sources for August 17, 2007 through May 22, 2008, and then analyzed the lifespans of the 182 new 0Day vulnerabilities. This provided a total of 491 0Day vulnerabilities for analysis.

We were concerned about the possibility that the statistics of the 0Day vulnerabilities might have dramatically changed between the first collection of data and the second collection period, or that the statistics might be significantly different between the two sources of data due to some unknown differences in the underlying environment.

Consequently, we initially kept the four sets of vulnerabilities separated for individual comparison. The four sets are "ZDI OLD", "ZDI NEW", "iDefense OLD", and "iDefense NEW". ZDI OLD and iDefense OLD consist of the 0Day vulnerability advisories, posted on ZDI and iDefense respectively, during the first data collection period. ZDI NEW and iDefense NEW consist of the 0Day vulnerability advisories, posted on ZDI and iDefense respectively, during the second collection period.

3.2 Characterizing 491 0Day Lifespans

When characterizing the 491 lifespans, four important questions were considered.

1. Was the mean and standard deviation of the lifespans from the four 0Day vulnerability data sets similar?
2. As a first order approximation, could the lifespans be reasonably characterized using a log-normal distribution?
3. Was the underlying population distribution of vulnerability lifespans stable over time?
4. Did the lifespans from ZDI and iDefense come from the same underlying population of vulnerability lifespans?

The rest of this section answers those four questions.

3.2.1 Mean and Standard Deviation of 0Day Lifespans

The calculated means and standard deviations of the four vulnerability data sets may be seen in Table 1. The largest mean for lifespans was 169.81 days (ZDI NEW) and the smallest was 112.74 days (ZDI OLD). The largest standard deviation was 153.21 days (iDefense OLD). The mean and standard deviations of the data sets which were formed by combining the ZDI data into one set, the iDefense data into another set, and combining all of the data together may also be found in Table 1. The means for each of these three combined data sets are very close to each other, approximately 130 days.

Vuln Data Set	Vuln count	Days: Mean Lifespan	Days: STD
ZDI OLD	100	112.74	76.13
ZDI NEW	63	169.81	145.57
iDefense OLD	209	125.11	153.21
iDefense NEW	119	139.47	138.33
ZDI Combined	163	134.80	111.49
iDefense Combined	328	130.32	147.93
All-Data Combined	491	131.81	136.81

Table 1 – Mean and Standard Deviation of 0Day Lifespans

For each data set, including the combined sets, the lifespans were placed in bins 25 days wide. We then plotted the vulnerability counts against lifespans. As an example, see Figure 1 for the plot of the four data sets combined. When visually inspecting the lifespan plot in Figure 1, and each of the other plots as well, it seemed possible that the lifespans might be log-normally distributed.

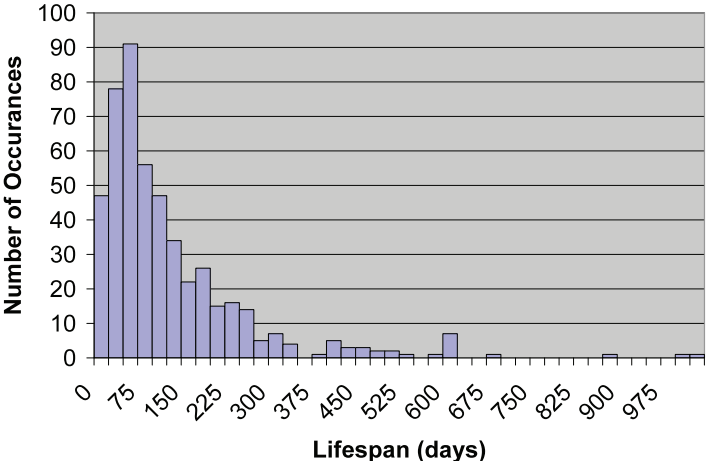

Figure 1 – Lifespan of the Four Combined Data Sets

3.2.2 Modeling Lifespans Using a Log-normal Distribution

If the lifespan of a 0Day vulnerability is thought of as the outcome of the discoverer's unique set of attributes, each vendor's patch development process, the individual vendor's economic factors at any moment, the difficulty of patching, and the particular individuals involved in creating the patch then a log-normal distribution [12] may be an appropriate model. To determine how well a log-normal distribution would model the distribution of lifespans in the various 0Day vulnerability data sets, the natural log was taken of each lifespan and placed in bins 0.5 wide. Then the count of vulnerabilities were plotted against the log values. The plot for all of the data sets combined is shown in Figure 2. Visually, to varying degrees, each of the log plots seemed to have a Gaussian distribution.

To quantitatively determine the goodness of fit of the log-normal distribution to the lifespan data the mean of the log values, and the corresponding standard deviations were calculated for each of the original four data sets, the combined ZDI data, the combined iDefense data, and all of the data combined. For each data set this information was then used to calculate the R^2 value, which roughly represents how well the vulnerability lifespan distributions of the set are represented by a log-normal distribution. The closer R^2 is to 1 the better the fit.

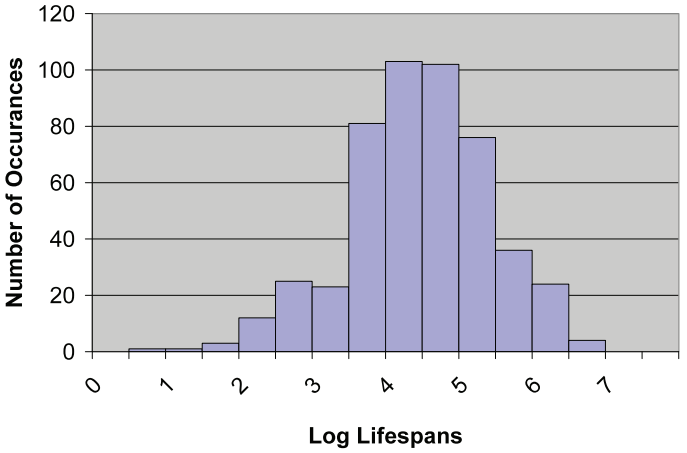

Figure 2 – Log Scale Plot of the Four Combined Data Sets.

The results of these calculations are shown in Table 2. From this we surmise that modeling the lifespans as a log-normal distribution is a reasonable first order estimate of the 0Day vulnerability distributions found in the data sets. We recognize that other distributions, such as the Poisson, might yield improved lifespan models and understanding of underlying causes, but those possibilities will be evaluated in later research.

Modeling the lifespans of each data set as a log-normal distribution leads to the question of whether the underlying population of 0Day lifespans stays constant over time and whether the ZDI data sets and the iDefense data sets represent samples from the same population.

Vuln Data Set	Vuln Count	LN: Mean Lifespan	LN: Standard	R^2
ZDI OLD	100	4.4802	0.7642	0.7547
ZDI NEW	63	4.7878	0.8662	0.6602
iDefense OLD	209	4.2740	1.0967	0.8942
iDefense NEW	119	4.5839	0.8665	0.8657
ZDI Combined	163	4.5991	0.8165	0.9067
iDefense Combined	328	4.3864	1.0287	0.921
All-Data Combined	491	4.4570	0.9678	0.9157

Table 2 – Mean, Standard Deviation and R^2 Values for 0Day Lifespans

3.2.3 Stability of Lifespan Distributions

In order to determine the likelihood that the means of the logs of the four vulnerability data sets came from the same underlying distribution the t-test was performed on each pair of vulnerability sets. The results are shown in Table 3. Corresponding to the t-value is the p-value that represents the chance that the actual measured means of the two data sets are due to them both actually having the same mean value (our null hypothesis).

The table shows that when comparing ZDI OLD with ZDI NEW there is a well over 95% (1-0.0163) confidence that the mean of the underlying ZDI vulnerability distribution changed over time (or there is a changed bias in the sampling process used by ZDI). A similar statement may be made of the underlying iDefense 0Day vulnerability lifespan distribution by inspecting the p-value when comparing iDefense OLD with iDefense NEW. Thus, for ongoing estimates of 0Day lifespans, it may be important to continually collect the 0Day statistics as reported at ZDI and iDefense. Longer term 0Day vulnerability data collection, and analysis of the processes used by ZDI and iDefense are needed to more fully understand the apparent change in 0Day lifespans.

t, p	ZDI OLD		ZDI NEW		iDefense OLD		iDefense NEW	
	t	p	t	p	t	p	t	p
ZDI OLD	0	1. 00	2.425	0.016	1.760	0.079	-0.957	0.340
ZDI NEW	-2.425	0.016	0	1.00	-3.411	0.001	1.511	0.133
iDefense OLD	-1.760	0.079	3.411	0.001	0	1. 00	2.647	0.009
iDefense NEW	0.957	0.340	-1.511	0.133	-2.647	0.009	0	1. 00

Table 3 – t-test on the Four 0Day Vulnerability Lifespan Data Sets

Table 3 also shows with high confidence that the ZDI 0Day vulnerability data sets are not being drawn from the same underlying population as the iDefense vulnerability data sets, or there are different selection biases being imposed by the two firms. Further exploration of the specific processes used by ZDI and iDefense are needed before drawing any firm conclusions.

3.3 Control System Vulnerabilities from NVD

Control Systems are composed of various hardware and software components some of which are also used in the more general domain of Information Technology (IT). For example, the Microsoft Windows operating system is commonly installed within Distributed Control Systems (DCS) or Supervisory Control and Data Acquisition

(SCADA) systems. We needed to identify which IT vulnerabilities disclosed in the NVD applied to control system environments.

Two techniques were used to identify publicly disclosed vulnerabilities applicable to control systems.

1. A review of vendor marketing information
2. Components known to be deployed in control system environments

The first technique involved visiting control system vendor web sites and identifying third party software the advertised product claimed to use or support. This effort included many of the major control system vendors such as Siemens, ABB, Areva, GEFanuc, Honeywell, Emerson, Control Microsystems, Citect, ICONICS, Yokogawa, and others. This technique resulted in a list of common IT software including versions of Windows operating system, Oracle Database, Microsoft Excel, SQL Server, and Sun Java, to name the most common. This list was matched against the "product" field entries in the NVD to identify publicly disclosed vulnerabilities that may exist in control system environments.

The second technique involved manually reviewing public vulnerability disclosures as reported in the National Vulnerability Database (NVD) between April 2006 and September 2008. The entries were manually filtered based on an ability to demonstrate that the vulnerable products were used in a control system environment.

Our NVD control systems vulnerability list was formed by merging the lists of vulnerabilities identified from the two techniques described above. The total number of vulnerabilities in NVD as of 10/9/2008 was 33,217. The number of vulnerabilities applicable to control systems identified by the above method was 4,321 (13% of the total).

This method has the following potential weaknesses.

- Since the collection of vendor marketing information was manual, it is possible that software supported by control systems was missed. The effect of this type of error tends to make our analysis conservative.
- Since the review process was manual, it is possible that the reviewing party failed to dismiss software that is not used in control system environments. We call this error “under-dismissal.” This is the only factor that would make our analysis less conservative. In order to address this concern we performed a final scrub of the 101 vulnerabilities identified by this method as applicable to control systems from the set of 491 vulnerabilities described in section 3.1 above. We found sufficient justification for each of the 101 vulnerabilities to conclude that the associated software was most likely used in a control system environment.

- Each review is limited in time. The review was carried out over a period of 29 months. It is likely that some vulnerabilities applicable to control systems were reported before this period began. Such vulnerabilities were only included in our analysis if they were identified using the first technique. This serves to make our analysis more conservative.
- We were unable to address one final concern: deployment of the identified vulnerable software. In the end we were forced to make a decision about where to draw the line on including or excluding vulnerabilities. Almost any software could be deployed in a control system environment. For example, review of the types of products our method excluded shows numerous content management systems (CMS). At the same time, we have evidence that one commonly deployed control system product uses a version of "PHP-Nuke," a CMS for which there are numerous entries in the NVD. Nevertheless, we excluded PHP-Nuke from our analysis to be conservative.

3.4 Characterizing Control System Vulnerability Lifespans

To characterize the 0Day lifespans of vulnerabilities applicable to control systems we analyzed the statistical characteristics of a subset of the 491 0Day vulnerabilities from ZDI and iDefense described above in section 3.1. There were 101 out of 491 vulnerabilities identified as applicable to control systems using the method described above in section 3.3. Again, the count of vulnerabilities was plotted against the natural log values of the lifespans for this data set and is shown in Figure 3. To determine how well a log-normal distribution would model the lifespans from this dataset the R^2 value was also calculated. Table 4 is a summary of the statistics of 0Day lifespans for the control system data set. Based on these results we again chose the log-normal distribution as a reasonable first order estimate for the control system vulnerability 0Day lifespans.

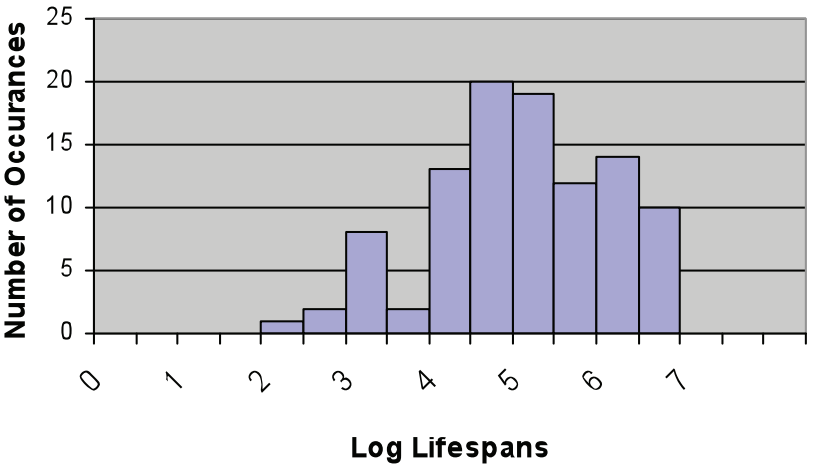

Figure 3 – Log Scale Plot of the Control System Vulnerability Lifespans

Vuln Data Set	Vuln Count	Days:Mean Lifespan	Days: STD	LN:Mean Lifespan	LN: STD	R^2
Control Systems	101	155.63	143.01	4.6624	0.9782	0.7435

Table 4 – Mean, Standard Deviation and R^2 Values for Control System 0Day Lifespans

4 Estimation of 0Day Vulnerabilities in Existence

The number of 0Day vulnerabilities that existed on a specific date in the past was estimated using the publication dates from the NVD and a model for 0Day lifespan as described above in Section 3. The estimated number is simply a count of all the vulnerabilities with a vendor notification date less than the date of interest and a public announcement date greater than the date of interest. The vendor notification date of each announced vulnerability is calculated by projecting backward in time from the publication date based on a sample lifespan selected from one of the 0Day lifespan log-normal distributions. This process is repeated multiple times and the result is averaged to obtain the statistical mean. Figure 4 is the pseudo code for the initial method used to obtain estimates for all dates since the earliest NVD vulnerability was published

(October 1988). Notice that this method tends to underestimate the number of 0Days in existence for dates near the present day because the number of published vulnerabilities for future dates is unknown.

4.1 Estimation for Vulnerabilities of All Types

Figure 5 shows the initial method's results using three different lifespan models from Table 1 (ZDI NEW, ZDI OLD and AllData COMBINED). The results are the average of 1,000 runs. The shapes of the graphs are similar for the three different lifespan models. The graphs show several sharp drops in the estimated number of 0Days. The sharp drops are caused by the irregular nature of the publication dates from NVD. For example, there were a large number of vulnerabilities publically announced on December 31 for each of the years 2002, 2003 and 2004. (798 on Dec. 31 2002, 441 on Dec. 31 2003 and 1113 on Dec. 31 2004). On those dates the estimated number of 0Day vulnerabilities has a large reduction because the estimation method treats a large number of publications as an immediate drop in the number of 0Days in the pipeline. Based on the combined results shown in Figure 5, it is reasonable to estimate the number of 0Day vulnerabilities in existence on any given day during 2006 to be around 2,500.

We compared the estimates obtained for log-normal vulnerability lifespan distribution models to estimates obtained using a simpler model where the 0Day lifespan is assumed to be a constant value. Figure 6 shows a comparison between the ZDI NEW log-normal distribution model versus a constant 169 day lifespan model (169 days is the average lifespan for the ZDI NEW data set). The comparison shows that this simpler model produces estimates that are a reasonably good approximation to the results obtained from the log-normal distribution model.

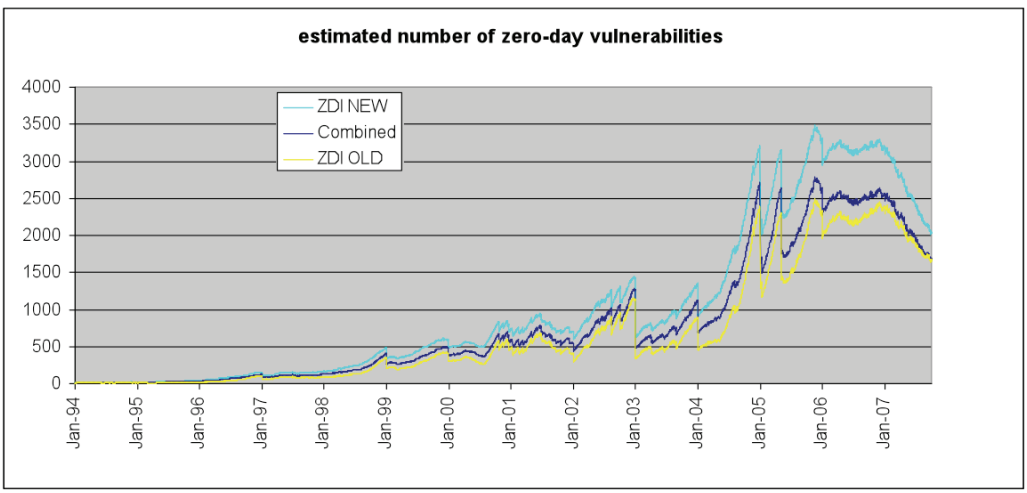

Figure 5 –Estimated Number of 0Day Vulnerabilities Using NVD Public Announcement Dates and Three Cases of Log-normal Distribution for Lifespan

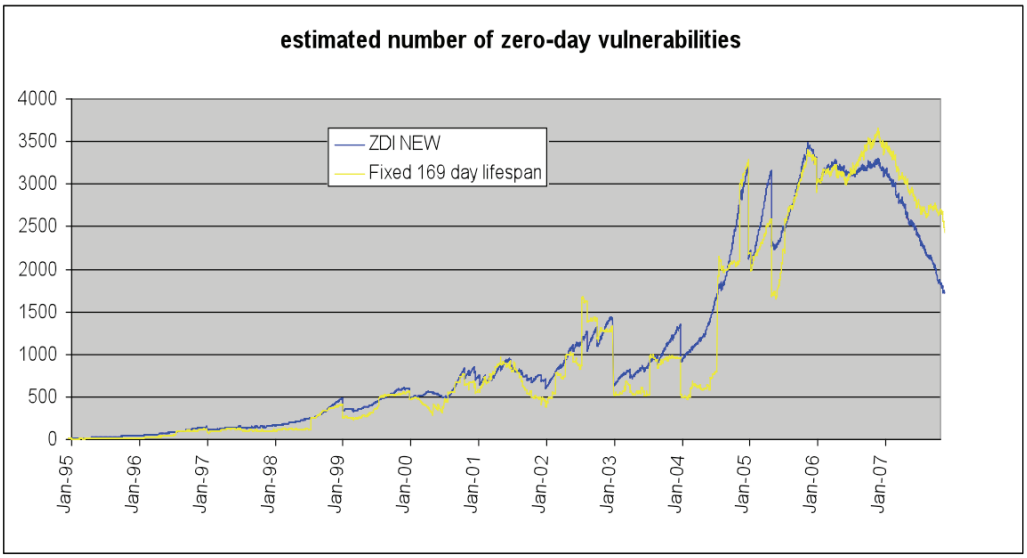

Figure 6 – Estimated Number of 0Day Vulnerabilities Using NVD Public Announcement Dates and Log-normal Distribution for Lifespan (ZDI NEW) Compared to Constant 169-days Lifespan

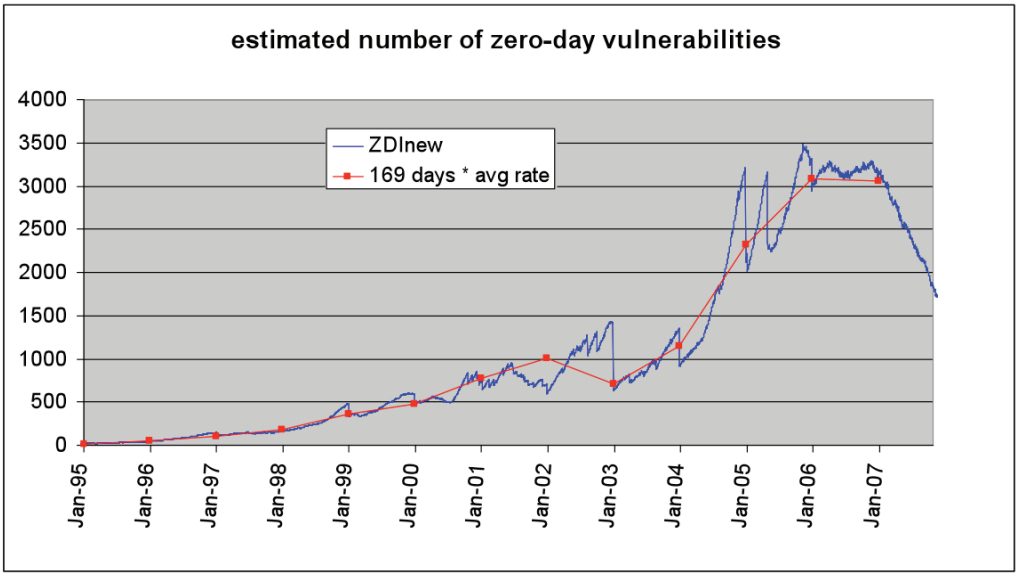

Figure 7 – Estimated Number of 0Day Vulnerabilities Using NVD Public Announcement Dates and Log-normal Distribution of Lifespan (ZDI NEW) Compared to Average Rate Times 169-days. Average Rate is the Number of Announced Vulnerabilities Per Day Averaged over the Succeeding 365 Days

The simplest method we used for estimating the number of existing 0Day vulnerabilities is the average lifespan of the chosen model multiplied by the average daily vulnerability public announcement rate. Figure 7 is a comparison of this method with the initial method described earlier. The ZDI NEW log-normal lifespan distribution model using the initial estimation method is plotted along with this simple method. The daily average public announcement rate is calculated once per year and is averaged over the succeeding 365 days. The estimate is the daily average multiplied by 169 days (the average lifespan for the ZDI NEW data set). This comparison indicates that the simple method provides a first order approximation that is comparable to the other more complex methods and models.

4.2 Estimation for Control System Vulnerabilities

The number of Control System 0Day vulnerabilities that existed on a specific date in the past was estimated using the algorithm in Figure 4 but the "number of vulnerabilities published" on each date in the past included only vulnerabilities applicable to control systems as determined by the method in section 3.3. The lifespan distribution model used for this estimation was a log-normal distribution with mean value = 0.9782; and standard deviation = 4.6624 as obtained from the analysis in section 3.4. Figure 8 shows the estimated values for dates ranging from 1992 through 2007. The most recent estimates (for years 2006 and 2007) range from about 250 to 350.

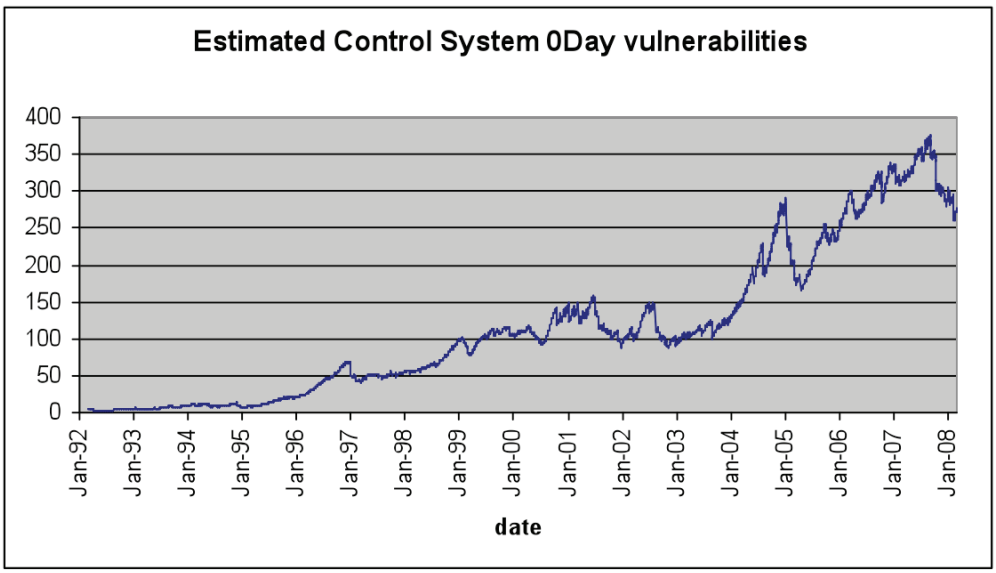

Figure 8 – Estimated Number of Control System 0Day Vulnerabilities Using NVD Public Announcement Dates

5 Modified Estimation of 0Day Vulnerabilities in Existence

The 0Day estimations discussed in the previous section were based on a lifespan that begins with vendor notification and ends with a public announcement. However, this represents only a portion of the true lifespan, which actually begins with vulnerability discovery. Unfortunately, the date of initial discovery is usually not known nor as well documented as the vendor notification date. As explained in section 2.9, the NVD database has "discovery date" records for a small fraction of its entries, and for the cases where there is a recorded discovery date, the validity of the data is suspect, therefore "discovery date" data from the NVD database could not be used to estimate 0Day lifespan.

A small but intriguing set of 15 vulnerability lifespans was supplied to us by a research group that does vulnerability discovery. We cannot disclose the identity of the research group, and there is no evidence that these 15 lifespans are representative of all 0Day vulnerabilities. However, these lifespans were included in our study because they are the elapsed time from actual discovery date to public announcement date. These same 15 vulnerabilities were also discovered independently and publicly disclosed by a different research group, which provides some added assurance that the lifespan data is valid. Figure 9 is a histogram that shows the lifespan data for these 15 vulnerabilities. The average 0Day lifespan for these 15 vulnerabilities is 256 days and the median lifespan is 200 days. As expected, the average lifespan is larger than the lifespans discussed in the previous sections and would be expected to result in larger estimates for the number of 0Day vulnerabilities in existence.

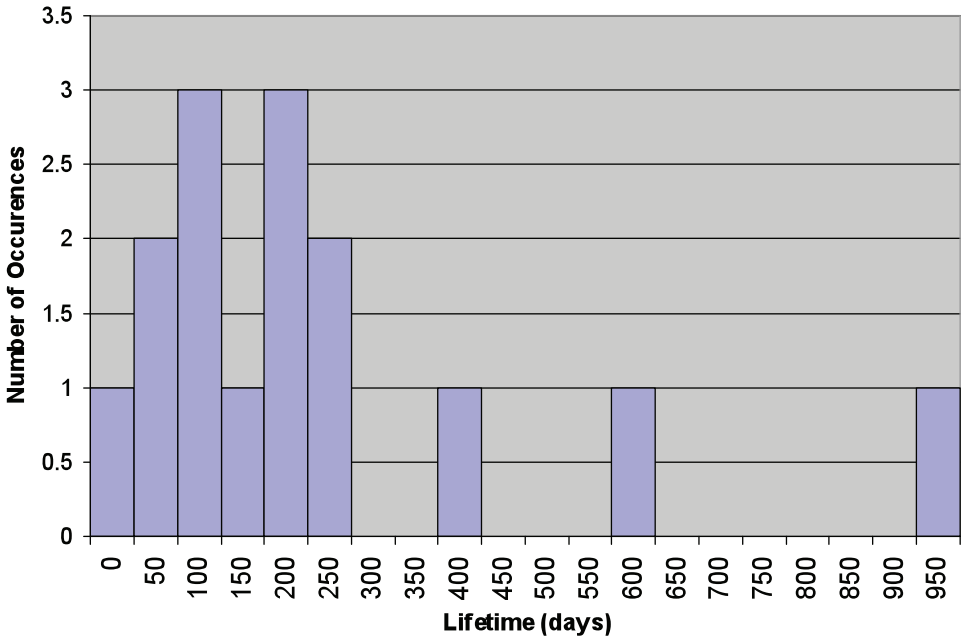

Figure 9 – Fifteen 0Day Life Spans from Discovery to Public Announcement

The 15 lifespans described above were used to estimate the number of 0Day vulnerabilities in existence. The estimates were calculated using the NVD public announcement dates and the method in Figure 4 but with lifespan samples obtained by a uniform random selection from the 15 lifespans. The results are shown in Figure 10. The results show that the new estimate for the worst year (2006) is about 4500 0Day vulnerabilities in existence on any given day. This is a less conservative estimate than the results from the previous section and since the lifespan dataset is small it should not be considered a high confidence estimate. However, since these 15 lifespans begin at the moment of discovery rather than the vendor notification date, this estimate could very well be more realistic.

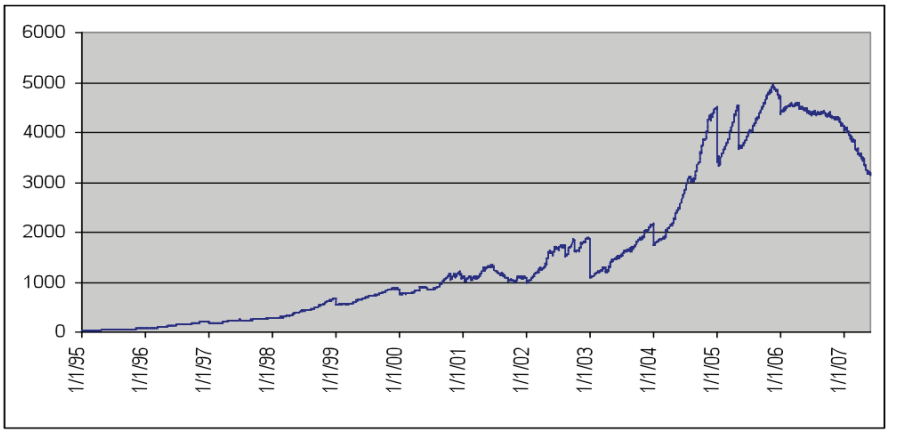

Figure 10 – Estimated Number of 0Day Vulnerabilities Using NVD Public Announcement Dates and Statistics of 15 Discovery to Announcement Lifespans

6 Control System Specific Vulnerabilities

We collected data for a set of 45 vulnerabilities that are specific to control systems. That is, these 45 vulnerabilities apply to software developed specifically for control systems and do not apply to the general IT world. These vulnerabilities were identified partly from open source information and partly from US-CERT. Some of the vulnerabilities originated from CERT/CC. Table 5 is a summary of lifespan results for this set of vulnerabilities. For the cases where discovery date is known, it was used as the start date for the average lifespan calculation. For cases where a vulnerability was reported to CERT/CC, the reported date was used as the start date for the average lifespan calculation. Similarly, the date reported to the vendor was used as the start date for cases where that date was known. Average lifespan was calculated using the NVD published date as the end of lifespan for the cases where that date was available and was also calculated using the patched date as the end of lifespan when that date was available. For vulnerabilities that have no patched date or published date, the lifespan is the time interval from earliest identifiable date to the date of this writing. The average lifespan for

vulnerabilities that have no published or patched date is 504 days, but is only 131.2 days when vulnerabilities for which we have no recent status updates, are excluded.

Category (start of lifespan)	Vulnerability Count			Average Lifespan (days)	
	Total	Published	Patched	Start date to published date	Start date to patched date
Discovery Date Known	6	4	3	272.2	147
Reported to CERT/CC	44	16	13	128.4	121.4
Reported to Vendor	16	13	12	164	153
Not Published or Patched (start is earliest identifiable date)	26			504, 131.2 (to date)	
Total Unique	45	19	17		

Table 5 – 0Day Lifespans for a Set of 45 Control System Specific Vulnerabilities

The average lifespan data for control system specific 0Day vulnerabilities shown in Table 5 is not significantly different than the average lifespan data for general IT vulnerabilities shown in Table 1 and Section 5. That is, when discovery date is unknown the average minimum lifespan is 112 to 170 days while the average lifespan starting with known discovery date is about 8 to 9 months (256 days for 15 IT vulnerabilities and 272 days for 6 control system specific vulnerabilities).

We are aware of many more control system specific vulnerabilities besides those summarized in Table 5 that have not yet been publicly disclosed. For example, INL has discovered vulnerabilities during security assessments, and vulnerabilities are discovered by multiple commercial organizations that specialize in security testing of control system components. The control system vendors are notified when these organizations identify new vulnerabilities but most of the vulnerabilities are currently not publicly disclosed. We are not free to specify the number of identified but undisclosed control system specific vulnerabilities of which we are aware in existence at this time, but we can say it is a significant number, perhaps in the hundreds.

Patches have been created to mitigate many of those vulnerabilities and by our definition, when a vulnerability is patched, it is no longer a 0Day vulnerability. However, the existence of a patch does not necessarily imply it has been deployed in the field. This is especially true for control system specific vulnerabilities. Since most control system specific vulnerabilities have not been publicly disclosed, it may be difficult for an

owner/operator to determine whether their control system has the relevant patches installed.

Control systems have significantly different operational needs than general computing environments and are not patched with the frequency that IT departments are used to patching [14]. There may be valid reasons to have a different vulnerability disclosure process than for general software vulnerabilities. However, at this time there is no generally accepted process for public disclosure of control system specific vulnerabilities. The lack of a public disclosure process means that control system vendors have less incentive to deploy patches to the field. The lack of public disclosure also means that we cannot effectively use disclosure history to estimate the number of control system specific 0Day vulnerabilities in existence.

An estimate of 0Day control system specific vulnerabilities requires a modified approach for a variety of reasons including 1) the lack of public disclosure history, 2) vulnerability research is being performed by a smaller and apparently different set of security researchers, and 3) control system specific applications are not as pervasive. Given these and other constraints we identified a few new mechanisms and approaches which may be useful for estimating and incorporating control system specific vulnerabilities into an improved 0Day estimation method.

- Extrapolate from the results of previous tests of control system applications. A quantification of test coverage could be used to predict the future discovery of vulnerabilities. The number of vulnerabilities not yet discovered may be estimated by modeling the relationship between test coverage and discovery.
- As markets for 0Day vulnerabilities expand, a study of the market may be used as a predictor.
- Software engineering methods may be used to predict the density of security related faults in control system applications.
- Study of vendor practices for deployment of patches for vulnerabilities that have not been publicly disclosed, including sampling of field installations. This study would provide an improved estimate of the true lifespan of control system specific 0Day vulnerabilities.

These new mechanisms and approaches appear promising and will be evaluated during the course of the next year.

7 CVSS Base Score and 0Day Lifespans

Given the estimations of the number of 0Day vulnerabilities and their lifespans, it is important to ask whether the estimates may be misleading about the risk 0Days pose. If the lifespans are very long for low impact, difficult to exploit, vulnerabilities but very short for high impact, easy to exploit, vulnerabilities then the risk would be lower than the raw numbers lead us to believe. To evaluate this possibility we used the 405 0Day vulnerabilities from ZDI and iDefense which had a CVSS Base score assignment, and

broke them into categories based on their score. CVSS is an industry standard for assessing the severity of computer system security vulnerabilities.

The CVSS Base score consists of a weighting of the impact and exploitability subscores. The impact subscore is an evaluation of the potential impact to confidentiality, integrity, and availability from exploitation of the vulnerability. The exploitability subscore is an assessment of the complexity of actual exploitation. This includes measures such as whether the attacker may exploit the vulnerability remotely, how complex the actual attack process is expected to be, and the required number of authentications during attack execution. Details of the CVSS Base score may be found in the CVSS scoring guide [10].

We expected to find that the vulnerabilities which are easily exploitable and have a high impact (very high CVSS Base scores) would have significantly shorter lifespans since it would seem to be in the software owner's and vendor's interest to have the vendor devote their limited resources to fixing them quickly.

	Low CVSS (0.0 – 7.9)	
	Vuln Counts	Mean Lifespan
ZDI OLD	45	90. 0889
ZDI NEW	22	181.9545
iDefense OLD	102	100.4216
iDefense NEW	57	118.7719
Control System	52	137.13
All Data Combined	226	110.9292
	High CVSS (8.0 – 10.0)	
ZDI OLD	53	130.8491
ZDI NEW	34	164.8824
iDefense OLD	31	145.9032
iDefense NEW	60	142.5500
Control System	49	175.26
All Data Combined	178	143.9157

Table 6 – Mean Lifespans Using Two CVSS Base Score Categories

The initial analysis results are displayed in Table 6. We were surprised to see that the mean lifespans of the most severe vulnerabilities (CVSS Base scores 8-10) were actually longer than for the less serious vulnerabilities in three out of the four vulnerability data sets. Upon investigation of the exception we found in ZDI NEW, we discovered that the higher average lifespan value in the low severity category was due to a single

vulnerability with a CVSS Base score of 7.8 and a very long lifespan. It is interesting to note that with all of the 0Day vulnerability data sets combined, the mean lifespan of the high severity vulnerabilities are about 30% longer than the low severity cases. This is just the opposite of what would be hoped for from a security perspective.

	Low CVSS Base Score (0.0 – 3.9)	
	Vuln Counts	Mean Lifespan
ZDI OLD	0	0
ZDI NEW	0	0
iDefense OLD	7	101.8571
iDefense NEW	1	13.00
Control System	3	31.66
All Data Combined[1]	8	90.75
	Medium CVSS Base Score (CVSS 4.0 – 6.9)	
ZDI OLD	16	130.8491
ZDI NEW	16	164.8824
iDefense OLD	57	145.9032
iDefense NEW	26	142.5500
Control System	28	175.26
All Data Combined	115	143.9157
	High CVSS Base Score (7.0 – 10.0)	
ZDI OLD	82	113.2317
ZDI NEW	40	181.55
iDefense OLD	69	124.9420
iDefense NEW	90	134.5333
Control System	70	168.00
All Data Combined	281	132.6548

Table 7 – Mean Lifespans by CVSS Standard Base Score Categories

Rather than use the two, somewhat ad-hoc, CVSS Base score categories found in Table 6, we decided that for the sake of validation and completeness we would do the same analysis as before but make use of the low, medium, and high severity categories as formally defined in the CVSS documentation. This change led to the categories and

[1] The All Data Combined in this Table does not include Control Systems Vulnerabilities

analysis results seen in Table 7. In this case all four data sets show a longer lifespan for the high severity vulnerabilities, and the combined data set shows that the medium severity vulnerabilities have a lifespan approximately 20% longer than the low severity vulnerabilities and the high severity vulnerabilities have a lifespan approximately 20% longer than those of medium severity.

The tentative conclusion from this analysis is that the previous estimates for the number and lifespans of 0Day vulnerabilities may actually underestimate the risk since the estimates don't account for the extended lifespans of the more severe vulnerabilities.

However, more investigation is needed. The question of 0Days with high CVSS base scores having longer lifespans may simply be an artifact of the data we used, or it may be real and be caused by some hidden attribute of the process such as a greater difficulty in developing patches for the more serious vulnerabilities.

8 Conclusions

We demonstrated a method for estimating the number of past 0Day vulnerabilities. In the worst year (2006) we conservatively estimated that there was an average of 2,500 0Days in existence on any given day. Using a much smaller vulnerability data set, but one where the calculated vulnerability lifespans ranged from the moment of discovery to the date of public announcement, we more aggressively estimated that there was an average of 4,500 0Day vulnerabilities in existence on any given day during the worst year (2006). We conservatively estimated the number of 0Day vulnerabilities applicable to control systems that existed on a given day in the recent past (2006-2007) to be about 250. These estimates are first order approximations that are subject to change as more data becomes available.

We also provided preliminary evidence that the most serious of these 0Day vulnerabilities have longer lifespans than lower severity vulnerabilities. Consequently, 0Day vulnerabilities appear to represent a greater risk to our systems than even the estimated number of 0Day vulnerabilities would indicate.

9 Future Work

We are pursuing investigation into a variety of issues related to estimating both the past and current number of 0Day vulnerabilities, the risk they may pose to a variety of systems, and potential mitigations. The work includes the modification and application of the research described in this paper to individual software programs. This will then be followed by an investigation in applying the attack surface metric concept to systems.

Further, critical infrastructure control systems make use of many programs which are not pervasive and thus they have not undergone the vigorous assault of the larger security research community. Consequently the vulnerability data related to control system software is expected to require a modified approach for characterizing the lifespans and for estimating the number of 0Day vulnerabilities.

Of course the risk posed to systems from 0Day attacks rest not just on the number of 0Day vulnerabilities but also on how easy it is to acquire the desired vulnerability. The

ease of acquisition is impacted not just by how many potential attackers are aware of it, but also by the markets available for both buying and selling the 0Day vulnerabilities. This impact will be investigated.

Also, the number of 0Day vulnerabilities and their lifespans may impact the form of an optimal disclosure process which minimizes the risk window, particularly to our most sensitive critical infrastructure systems. Thus, the relation of 0Days to control system vulnerability disclosure will be investigated.

The question of 0Days with high CVSS Base scores having longer lifespans will also be investigated further to determine whether it is simply an artifact of the data we used, whether it relates to the difficulty in developing patches for the more serious vulnerabilities, or whether the more significant vulnerabilities are more closely guarded by the discoverer.

We are in the process of acquiring many more 0Day vulnerability discovery and public announcement dates from firms and individuals who make the discovery themselves. This data will be used to create an improved estimate of the lifespan of a 0Day vulnerability from initial discovery to public announcement.

A number of new mechanisms and approaches for estimating and incorporating control system specific vulnerabilities into an improved 0Day estimation will be investigated.

About the Authors – Miles McQueen has over twenty-five years of increasing technical leadership in the design, development, and protection of hardware-software systems. Before joining the Idaho National Laboratory (INL), Miles worked at Hughes Aircraft-Missile Systems Group, where he led and managed R&D efforts to develop state of the art infrared trackers for a variety of military efforts including anti-tank, anti-ship, and Strategic Defense Initiative systems. More recently at the INL, in support of the DHS-NCSD Control Systems Security Program (CSSP), he led the initial efforts to develop technical security metrics for owner-operators. While continuing to support the technical metrics effort, Miles is currently striving to develop a Threat Environment Characterization capability for the CSSP. His recent technical focus is on aspects of Artificial Intelligence, Intelligent Agents, Multi-agent Systems and how various human concepts and understanding of risk should inform agent behavior.

Dr. Wayne Boyer received a BS degree in Electrical Engineering at Brigham Young University and an MS degree in Electrical Engineering at Stevens Institute of Technology. He earned a Ph.D. in Computer Science from the University of Idaho. Since joining the INL, Wayne has been an Advisory Engineer/Scientist and has acted as a control system engineer, systems analyst, software engineer and researcher on various projects including robotics research, high level waste processing, distributed computing and control system cyber security. Wayne is an affiliate faculty in the Computer Science department at the University of Idaho in Idaho Falls. He has taught courses in Computer Science including Software Engineering, System Software, and Algorithms. He has published several papers on computer security and efficient scheduling algorithms for distributed computing systems.

Sean McBride is a Cyber Security Researcher/Analyst at the Idaho National Laboratory (INL) Critical Infrastructure Protection and Resilience Division. Mr. McBride's responsibilities include

providing federal control systems security stakeholders with situational awareness of the threat environment, analyzing vulnerabilities affecting control system installations, and lending emergency reach-back support. Prior to his position at the INL, Sean received a Masters of Business Administration degree from Idaho State University where he studied under the National Science Foundation cyber corps program. Sean is a Certified Information Systems Security Professional.

Trevor McQueen is a student at Harvey Mudd College.

Acknowledgements

We express appreciation to Debbie McQueen for her helpful contributions. This work was supported by the U.S. Department of Homeland Security, under DOE Idaho Operations Office Contract DE-AC07-05ID14517.

References

[1] National Infrastructure Protection Plan, June 30 2006.

[2] A. Ozment, "Improving Vulnerability Discovery Models", ACM Workshop on Quality of Protection, Alexandria, Virginia, October 2007.

[3] E. Rescorla, "Is Finding Security Holes a Good Idea", Security & Privacy, IEEE, Jan-Feb 2005, 14-19.

[4] A. Ozment,„S. Schechter, "Milk or Wine: Does Software Security Improve with Age?", Proceedings of the Fifteenth Usenix Security Symposium. Vancouver, BC, Canada, July 31 - August 4 2006.

[5] NIST, National Vulnerability database, http://nvd.nist.gov.

[6] H. Cavusoglu, J. Zhang, "Economics of Security Patch Management", The Fifth Workshop in the Economics of Information Security, University of Cambridge, England, June 2006.

[7] A. Arora, R. Telang, "Economics of Software Vulnerability Disclosure", Security & Privacy, IEEE, Jan-Feb 2005, 20-25.

[8] D, McKinney, "New Hurdles for Vulnerability Disclosure", Security & Privacy, IEEE, March-April, 2008, 76-78.

[9] S. Frei, B. Tellenbach, B. Plattner, "0-Day Patch- Exposing Vendors (In)security Performance", BlackHat Europe, Amsterdam, NL, March 2008.

[10] P. Mell, K. Scarfone, S. Romanosky, "CVSS- A Complete Guide to the Common Vulnerability Scoring System Version 2.0", On the Forum for Incident Response and Security Team, June 2007.

[11] D. Brumley, P. Poosankam, D. Song, J. Zheng, "Automatic Patch-Based Exploit Generation is Possible: Techniques and Implications", Proceedings of the IEEE Security and Privacy Symposium, May, 2008.

[12] J. Aitchison, J. Brown, "The Lognormal Distribution", Cambridge University Press, 1957.

[13] M. McQueen, T. McQueen , W. Boyer, M. Chaffin, "Empirical Estimates and Observations of 0Day Vulnerabilities", Proceedings of the 42nd Hawaii International Conference on System Sciences, Jan. 5-8 2009.

[14] J. Brodsky, Critical Infrastructure News, Discussion on patching Citech software vulnerability discovered by Core Security Technologies, http://news.infracritical.com/pipermail/scadasec/2008-June/000818.html, June, 2008.