Vulnerability survival analysis: A novel approach to vulnerability management.

Katheryn A. Farris[a], John Sullivan[a], and George Cybenko[a]

[a]Dartmouth College, Thayer School of Engineering, 14 Engineering Dr., Hanover, NH, 03755, USA

ABSTRACT

Computer security vulnerabilities span across large, enterprise networks and have to be mitigated by security engineers on a routine basis. Presently, security engineers will assess their "risk posture" through quantifying the number of vulnerabilities with a high Common Vulnerability Severity Score (CVSS). Yet, little to no attention is given to the length of time by which vulnerabilities persist and survive on the network. In this paper, we review a novel approach to quantifying the length of time a vulnerability persists on the network, its time-to-death, and predictors of lower vulnerability survival rates. Our contribution is unique in that we apply the cox proportional hazards regression model to real data from an operational IT environment. This paper provides a mathematical overview of the theory behind survival analysis methods, a description of our vulnerability data, and an interpretation of the results.

Keywords: Vulnerability, Common Vulnerability Severity Score (CVSS), survival analysis, Kaplan-Meier probability curves, Cox proportional hazards regression model

1. INTRODUCTION

One of the most fundamental steps in software and network security is vulnerability remediation. Vulnerabilities provide an open door for malicious penetration into an organization's network. The attackers motives may be to exfiltrate sensitive and personally identifying data, spy or steal proprietary assets and intellectual property, or install malware that can't be removed without paying a large ransom. Digitally securing an organization's enterprise-level network from these sort of attacks involves reducing total vulnerability exposure through methods such as vulnerability scans, and network patching.

Engineers and analysts are directed to which areas of the network need patching via vulnerability scan reports. But reports are difficult to understand due to both their spatial and temporal nature. The vulnerability counts are dense and their locations across the network, vast. Vulnerability scanning software outputs large reports delineating unique occurrences on host services. The data are difficult to parse and act on, making the resulting total vulnerability exposure not immediately evident. If we can better quantify total vulnerability exposure, we can act in a principled manner to minimize organizational risk to malicious threats.

Different efforts have been made to understand vulnerabilities, their lifecycles and associated risks. Two metrics have been proposed for assessing vulnerability exposure, namely, the median active vulnerabilities (MAV) and vulnerability free days (VFD).[1] These metrics were deduced from a vulnerability lifeline model, which was based on measuring when vulnerabilities are reported to a vendor to when that same vendor

Further author information: (Send correspondence to Katheryn A. Farris)
K.A.F.: E-mail: katheryn.a.farris.th@dartmouth.edu

Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security, Defense, and Law Enforcement Applications XVI, edited by Edward M. Carapezza, Proc. of SPIE Vol. 10184, 101840H · © 2017 SPIE · CCC code: 0277-786X/17/$18 · doi: 10.1117/12.2266378

publicly releases a patch. To further build upon this area, more studies need to be conducted with regard to the true effectiveness of vulnerability patches.

Some additional work has been conducted to investigate "real world" vulnerability patch deployments. Nappa, et. al. investigates the patch rate of client-side vulnerabilities for shared code software.[2] Our investigation differs in that it focuses on survival rates and patch effectiveness of server-side vulnerabilities. We report the observational results from a survival analysis study to assess which vulnerability features predict lower survival rates relative to the IT infrastructure. By understanding which features best predict reduced vulnerability exposure we obtain two insights: 1.) We get a glimpse into what currently drives decision-making for vulnerability management.; and 2.) We have a baseline metric by which we can compare future work in vulnerability management algorithms.

In this study, the following research questions are posed:

1. What are the overall trends of vulnerability survival rates as organized by the following features:
   - severity level;
   - whether or not the host is considered a mission critical service;
   - operating system (OS) type?
2. Of the three features outlined in question #1, which best predict lower vulnerability remediation rates (i.e. Higher time to death (hazard) rates.)?
3. How does accounting for vulnerability age change the prediction for each feature on vulnerability remediation rates?

The primary contributions of this paper is to investigate a vulnerability's time-to-death (i.e. hazard rate) by applying survival probability estimates and the cox proportional hazards regression model on monthly vulnerability scan output from a Security Operations Center (SOC). Such an analysis has never previously been applied to of real vulnerability data. This novel concept was inspired by survival studies in clinical medicine. Our survival model accounts for the data's spatial aspect through observations of one unique vulnerability occurring on one unique host service, and the temporal aspect through assessing overall vulnerability persistence and time-to-death. The final goal of this work to provide empirical performance metrics as a baseline for testing future vulnerability algorithms for improved management and triage.

In the following sections, we review related work in Section 2 and in Section 3, we offer an overview of the formal math behind survival analysis models. We review our dataset in Section 4, and the results from our analysis in Section 5. Finally, in Sections 6, and 7, we discuss threats to validity, future work and conclusions.

2. RELATED WORK

This section describes related literature as it is categorized into different domains. First, we review recent research in the area of vulnerability scoring, management and treatment, as well as literature on data-driven solutions to vulnerability management. Finally, we cover the concepts behind survival models and a class of statistical methods that can be utilized in a survival analysis such as Kaplan-Meier survival curves, and the cox-proportional hazards regression model.

Vulnerability Scoring, Management and Treatment The Common Vulnerability Scoring System (CVSS) provides a base score for assigning vulnerability severity.[3] It is defined by organizations such as Carnegie-Melon University's CERT and the National Institutes of Standards (NIST). Cyber-Security Operations Centers (CSOCs) typically manage vulnerabilities with triage by the CVSS base value alone. Allodi, et al. find that incorporating external factors such as black market exploit data into the CVSS base score provide a more statistically significant indication of *true* vulnerability severity.[4] Other research indicates that security modeling with CVSS data alone does not accurately portray the time-to-compromise of a system. They also found that security models which base decisions on only the most severe CVSS data are less reliable than those that consider all vulnerabilities, regardless of their CVSS severity.[5] To date, most CSOC vulnerability response programs that use CVSS typically amount to working down the list of vulnerabilities from the "most severe" to "least severe" CVSS values.

Two metrics related to this work have recently been proposed for assessing vulnerability exposure, namely, the Median Active Vulnerabilities (MAV) and Vulnerability Free Days (VFD).[1] These metrics were deduced from a vulnerability lifeline model, which was based on measuring when a vulnerability is reported to a vendor to when that same vendor issues the patch. Our work is different in that we are investigating vulnerability patch effectiveness and overall vulnerability exposure based on live data coming out of a CSOC. Basing an analysis on when the vendor discovers a new vulnerability to when they release the patch is a good first step in modeling vulnerabilities; however, it does not offer a complete picture. Just because the vendor creates a vulnerability patch does not necessarily mean it is effectively being deployed. As a movement toward research in vendor-to-CSOC relationships in vulnerability patch deployment, Cavusoglu, et al. developed a game-theoretic model which incorporates a cost-benefit analysis of patch management to understand better methods of interaction between a vendor and a given CSOC.[6]

Data-Driven Solutions in Vulnerability Management Data-driven analysis has been fairly rare until recent years. NIST's most recent standards on data management have been developed and written in their 2011 report on continuous monitoring in CSOCs.[7] In the pursuit of science, some CSOCs are becoming more willing to share their data, along with proper data handling agreements, and large research organizations who own cyber-security data are becoming more interested in consolidating, anonymizing and sharing their data for the mutual interest of moving the discipline of cyber-security forward. Some recent work in this area include proposing a data-driven vulnerability maintenance policy, which uses Markov-decision processes (MDP) for the generation and graphical evaluation of relevant maintenance policies with limited data visibility.[8] Additionally, some work has been achieved to develop research methods for autocorrelated vulnerability data. Afful-Dadzie, et al. use a hybrid moving centerline residual-based and adjusted demerit (MCRAD) chart.[9] The authors provide an analysis which directs an administrator to unusual cases when automated patching is insufficient.

Survival Analysis Models Many real-world problems require the investigation into the time to the occurrence of an event, and, survival and death rates. As Harrell, et al. point out, if only the mere occurrence of an event is of interest (and not the time-line leading up to that event), the data can be analyzed via logistic regression.[10] An example of an appropriate use of this could be if rats in a tuberculosis study died within 6 months. The time-line of deaths before and after the 6-month threshold may not matter to the researcher, but, rather, the binary output of whether or not the rat survived beyond 6 months. Conversely, a survival analysis can be used in circumstances where the researcher is not only interested in whether the event occurred, but also the time-line leading up to the event.[10,11] Additionally, methods such as Kaplan-Meier probability curves, life tables, the Cox proportional hazards regression analysis, and hazard ratios are all components to

a comprehensive survival analysis, bringing about a statistically deep understanding into the complex nature of survival time-lines and death rates.

3. VULNERABILITY SURVIVAL ANALYSIS MODELS

In this section we describe the two key aspects of vulnerability survival analysis, namely: 1.) the survival probability estimates, and 2.) the cox proportional hazards regression model.

3.1 Survival Probability Estimates

Survival probability estimates can either be represented as a Kaplan-Meier curve or life tables. The Kaplan-Meier curve plots the empirically observed probability of survival rates against time. They typically display as step-functions because $S(t)$ is a constant between times of events. One can deduce metrics such as the survival rate median and mode as well as gain an understanding for overall trends between covariates. The results can also be summarized in a life table, which provides both survival and hazard estimates. These statistics are most useful for observing, summarizing, and comparing groups from a study.

The survival function is:

$$S(t_a) \quad = \quad S(t_{a-1}) * (1 - \frac{d_a}{n_a}),$$

where $S(t_{a-1})$ is the probability of the event (vulnerability remediation) not having occurred at time t_{a-1}. The number of observations that experienced the event at t_a are represented as d_a, and n_a is the number of observations still at risk of the event occurring just before t_a. The value n_a is calculated as $n_a = n_{a-1} - d_{a-1} - c_{a-1}$, where c_{a-1} represents the total number of censored observations at time t_{a-1}.

A censored* measurement is a phenomenon where the individual has not experienced the event of interest before the end of the study. In the context of our analysis, one observation is defined as the unique instance of a vulnerability on a unique host, and the event is the vulnerability remediation (i.e. "death"). If a vulnerability appears on a host, but does not experience remediation before the end of the study, it is effectively called *right censored.*

3.2 Cox Proportional Hazards Regression Model

The Cox Proportional hazards regression model measures the variable effect on the hazard of death. In the context of our study, the hazard is the expected number of events (vulnerability remediation) per unit of time (month).

Assumptions that must be met for the cox proportional hazards regression model are:

1. Independent among covariates (predictors).
2. A multiplicative/proportional relationship between covariates and the hazard of death.

*Analyzing data across time as well as censoring are features that make survival analysis and time-to-event studies more rigorous than other methods such as scatter plots, and logistic regression.

3. Constant hazard ratio over time (i.e. the natural log of the hazard coefficient and the covariates are linearly related).

The hazard function h(t) denotes the probability that if the observation of interest survives to time t, it will experience the event in the next instant (i.e. the instantaneous event of death), and is given by:

$$\begin{aligned} h(t) &= \lim_{dt \to 0} \frac{Pr[(t \leq T \leq t + dt) | T \geq t]}{dt} \\ &= \frac{f(t)}{S(t)}. \end{aligned}$$

Our research goal is to assess the relationship between the survival function to covariates (predictors). A parametric model of n fixed covariates that is based on the exponential distribution can be written in the form:

$$\begin{aligned} h_a(t) &= exp(\alpha + \beta_1 * x_{a_1} + \beta_2 * x_{a_2} + ... + \beta_n x_{a_n}) \\ &= \lambda_0(t) exp(\beta_1 * x_{a_1} + \beta_2 * x_{a_2} + ... + \beta_n x_{a_n}). \end{aligned}$$

The proportional hazards tells us the hazard ratio of two covariates of interest, such that:

$$\begin{aligned} HR_{ab} &= \frac{h_a(t)}{h_b(t)} \\ &= \frac{\lambda_0(t) * exp(\beta_1 * x_{a_1} + \beta_2 * x_{a_2} + ... + \beta_n x_{a_n})}{\lambda_0(t) * exp(\beta_1 * x_{b_1} + \beta_2 * x_{b_2} + ... + \beta_n x_{b_n})} \\ &= exp(\beta_1 * (x_{a_1} - x_{b_1}) + \beta_2 * (x_{a_2} - x_{b_2}) + ... + \beta_n * (x_{a_n} - x_{b_n})). \end{aligned}$$

A hazard ratio indicates the proportion of "deaths" (i.e. vulnerability patches) to total "illnesses" (i.e. the unique vulnerability occurrence on a unique host service).

4. DATA

Our data consist of monthly vulnerability scan reports from the Nessus software. We have scans from a twelve month period, spanning over 2,000 machines. We have one month of missing data, which we left out and therefore we effectively had eleven scans total.

The event of interest in this study is vulnerability remediation (i.e. vulnerability "death"). We define one observation as the unique combination of a vulnerability appearance on a given host service (IP address/domain name), and we define a remediation as two or more months where the vulnerability does not appear. This definition is based on the result of subject matter expert interviews where it was determined that if a vulnerability is offline for one month, it may be due to a host service being taken offline for reasons such as routine maintenance or upgrades. However, a host service would never normally be offline for more than one month. Therefore, if a vulnerability disappears for two months or mote, it is either due to remediation,

Table 1: Data distribution breakdown.

		Event Occurred		Total Observations
		No (Not remediated. (i.e. censored))	Yes (Remediated.)	
OS Type	Linux	907 (64.97%)	489 (35.03%)	1,396 (100%)
	Windows	940 (75.26%)	309 (24.74%)	1,249 (100%)
Severity	Critical	24 (22.01%)	85 (77.98%)	109 (100%)
	High	217 (59.45%)	148 (40.55%)	365 (100%)
	Medium	1,523 (73.75%)	542 (26.25%)	2,065 (100%)
	Low	83 (78.30%)	23 (21.70%)	106 (100%)
Mission-Critical Service	No	1642 (70.44%)	689 (29.56%)	2,331(100%)
	Yes	205 (65.29%)	109 (34.71%)	314 (100%)

or because the machine was decommissioned. Under either circumstance, it would count as a vulnerability remediation.

The power and precision of our vulnerability survival analysis model are related to the number of events that occur, and not the number of observations. Looking at Table 1, we see that the lowest number of vulnerability remediation events that occurred is 23 (for the low risk covariate). Survival analysis simulation studies have found that if at least 10 events occur per covariate, the study still holds. We also find some anecdotes from experts which encourage as much as 25 to 50 events occur per covariate. Given these guidelines, our data meet or exceed the standard and our study still holds power and precision. Table 1 breaks down each feature used in our study and by the percentage censored and remediated.

4.0.1 Master Dataset Example and Terminology.

Table 2 offers an example of what a snippet of our master data set looks like, with pertinent definitions listed below.

Table 2: Sample of what the master data set looks like.

Obs.	PluginID	Host	OStype	Time	Censored	CritService	Severity	CVECode	CVEAge
1	57608	A	Linux	5	1	1	High	CVE-2004-8219	12
2	45411	A	Linux	5	1	1	Low	N/A	N/A
3	10262	B	Windows	4	1	1	Critical	CVE-2012-4356	4
4	65821	B	Windows	2	0	0	Low	N/A	N/A
5	20007	C	Linux	8	0	0	Medium	N/A	N/A
6	57608	C	Linux	8	0	1	High	CVE-2004-8219	12
7	89058	C	Linux	3	1	1	Medium	N/A	N/A
8	78479	D	Windows	5	1	0	Medium	CVE-2009-6114	7
9	10262	D	Windows	2	1	0	Critical	CVE-2012-4356	4

Definitions of terminology in Table 2:

- Plugin ID: Unique vulnerability identifier code from Nessus scanning software.
- Host: Unique combination of IP address, port and protocol.
- OStype: Operating system type. We focused specifically on understanding the differences in vulnerability remediations between the Windows operating systems versus Linux.

- Time: Length of time vulnerability was on the host, calculated as the difference from its first point of observation to its final point of observation.
- Censored: Indicates a binary value "1" where the vulnerability did not receive remediation before the study ended, and "0" where the vulnerability did receive remediation.
- CritService: Indicates a binary value "1" if the vulnerability appears on a mission critical service, and "0" if not. Host services are considered mission critical if the organization is dependent on it for day-to-day core network operations. Examples include: Email servers, grading systems and web payment services.
- Severity: The Common Vulnerability Scoring System (CVSS) base score is an industry standard used to assess vulnerabilities based on a scale of 1 to 10. It represents the characteristics of a vulnerability which are constant over time and across user environments. It is composed of metrics capturing how the vulnerability is accessed and the conditions necessary to exploit it. The Nessus vulnerability scanning software we used categorizes the scores as follows:
  1. Critical severity vulnerabilities correspond to a CVSS metric of 10.
  2. High severity vulnerabilities correspond to CVSS metrics within the range of 7 to 9.
  3. Medium risk vulnerabilities correspond to CVSS metrics within the range of 4 to 6.
  4. Low risk vulnerabilities correspond to CVSS metrics within the range of 1 to 3.
- CVE Code: The Common Vulnerability Enumeration (CVE) code corresponds to the National Vulnerability Database (NVD), which composes a subset of VulnID. The Nessus vulnerability scanning software outputs a set of vulnerability detection data that goes beyond the CVE data. Hence, Table 2 lists "N/A" in some rows where the CVE data was not available.
- CVE Age: We obtain the CVE age directly from the CVE code. For instance, a CVE code "CVE-2013-1119" means that the vulnerability has been publicly known since the year 2013. We can therefore derive the age of that vulnerability as being 3 years old as of the year 2016.

5. RESULTS

In this section, we will review the results from our study. First, we offer an overview of the vulnerability trends based on reports from the National Institutes of Standards (NIST) National Vulnerability Database (NVD). Then we will review Kaplan-Meier probability curves in Section 5.1, and the univariate and multivariate cox proportional models in Section 5.2. Finally, we will offer a summary of points to be understood about the study in Section 5.4.

5.1 Survival Probability Estimates

In this section, we review general trends in the survival probability estimates in Figure 1, also known as Kaplan-Meier survival curves. The y-axis represents the likelihood of survival and the x-axis represents the number of months of survival. Notice that the Kaplan-Meier curve forms a stair-step function, indicating a percentage decrease between each "step" of vulnerability survival from month-to-month. We investigated the Kaplan-Meier curves for all data, by Operating System type, mission critical services, and, finally, by severity.

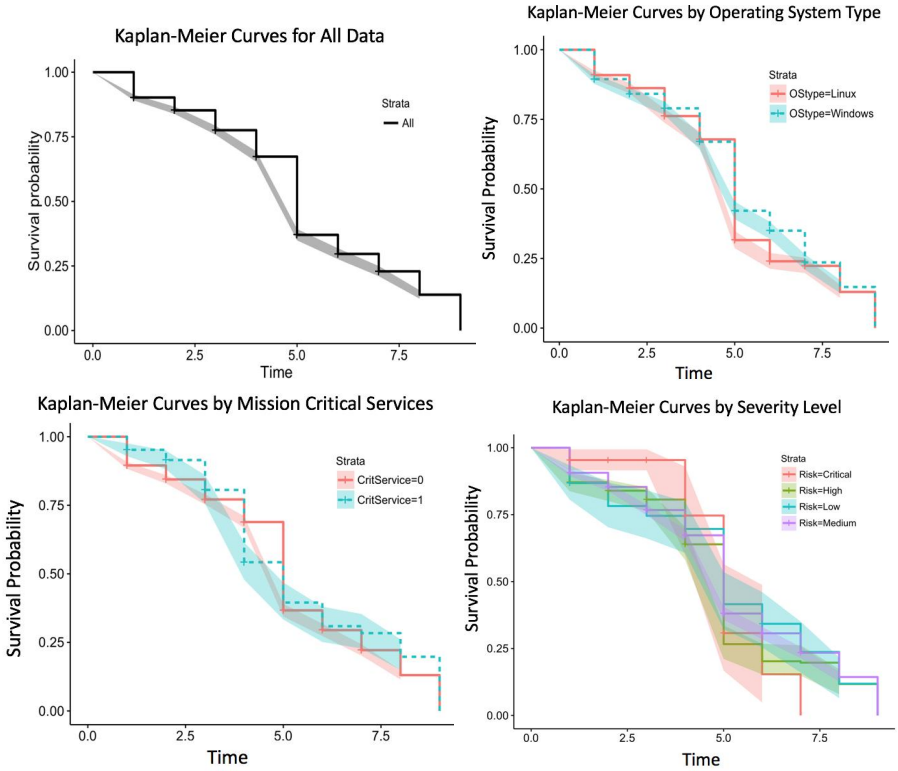

Figure 1: Kaplan-Meier Survival Estimates

5.2 Cox Proportional Hazards Regression Analysis

In this section, we review the results from the proportional hazards regression analysis of two data sets. Referring back to Section 4, we are reminded that the primary data set encompasses a total of 2,645 observations, and the secondary data set is a subset of the primary data set, amounting to 672 observations. We arrived at dividing our data into the primary and secondary data sets because we wanted to first investigate the following covariates of interest from the primary data set, namely: 1.) whether a vulnerability appears on a mission critical host service, 2.) the operating system type by which the vulnerability appeared, and 3.) the vulnerability's severity level. Then, we wanted to investigate how incorporating vulnerability age into the analysis influences the results, which we achieve by sub-setting the results to the data set with the CVE Age values.

5.2.1 Investigating how Operating System, Severity Levels and Mission Critical Hosts Influence Vulnerability Remediations

In this section, we use the full data set with 2,645 observations to perform both univariate and multi-variate analysis with the Cox proportional hazards regression model. In Tables 3, 4, and 5, we focus on one predictor at a time, namely the operating system type, severity level, and mission critical host services, while holding all other predictors constant at a time. In Table 6, we adjust for all covariates for a multivariate analysis.

Table 3 displays results from the cox proportional hazards univariate regression analysis for operating system type.

We test the following hypothesis:

Ho: Operating system type is not associated with vulnerability remediation rates.
Ha: Operating system type is associated with vulnerability remediation rates.

Notice a negative association between Windows OS vulnerabilities to Linux OS vulnerabilities. Recall that the parameter estimates represent the increase in the expected log of the relative hazard for each one unit increase in the predictor, holding all other predictors constant. To interpret the hazard ratio, we exponentiate the parameter such that exp(-0.387) = 0.6791, indicating that for every 67 Windows vulnerabilities which are remediated, 100 Linux vulnerabilities are remediated.

Additionally, the p-value of 0.015 indicates that we reject the null hypothesis, and find that operating system type is a statistically significant predictor of vulnerability remediation rates.

Table 3: Cox proportional hazards univariate analysis results for operating system type.

Vulnerability Risk Factor	Reference Variable	Parameter	Hazard Ratio (95% CI)	p-value
Windows OS	Linux OS	-0.387	0.6791	0.015

Moving on to table 4, we observe the cox proportional hazards univariate results for vulnerability severity. We find that severity levels alone are not statistically significant predictors of vulnerability remediations and we fail to reject the null hypothesis.

Hypothesis Tests:

Ho: Severity is not associated with vulnerability remediation rates.
Ha: Severity is associated with vulnerability remediation rates.

Table 4: Cox proportional hazards univariate analysis results for vulnerability severity.

Vulnerability Risk Factor	Reference Variable	Parameter	Hazard Ratio (95% CI)	p-value
High Severity	Critical Severity	0.00855	1.0893	0.70
Medium Severity	Critical Severity	-0.0244	0.7619	0.26
Low Severity	Critical Severity	-0.0272	0.7835	0.25

In Table 5, we observe the cox proportional hazards univariate results for whether or not a vulnerability appears on a mission critical service. We find that mission critical services are remediated at half the speed as non-mission critical services. One possible explanation for this is that mission-critical services need to remain operational at all times. Therefore, if a vulnerability remediation requires temporarily taking a service down, given the choice, system administrators and security engineers will opt for keeping the server operational over patched.

Hypothesis Tests:

Ho: Mission critical services are not associated with vulnerability remediation rates.
Ha: Mission critical services are associated with vulnerability remediation rates.

Table 5: Cox proportional hazards univariate analysis results for whether or not a vulnerability is on a mission critical service.

Vulnerability Risk Factor	Reference Variable	Parameter	Hazard Ratio (95% CI)	p-value
Mission Critical Service	Non-Mission Critical Services	-0.2419	0.5729	0.052

Finally, in Table 6, we observe the cox proportional hazards multivariate results for vulnerability survival rates. These parameter estimates are calculated after taking all other parameters into account. We find that the parameter values and hazard ratios are fairly close to the same output as the univariate regressions, and that most of the p-values increase in magnitude (hence, we lose some precision).

Table 6: Cox proportional hazards multivariate analysis results for covariates operating system type, vulnerability severity and whether or not a vulnerability appears on a mission critical service.

Vulnerability Risk Factor	Reference Variable	Parameter	Hazard Ratio (95% CI)	p-value
Windows OS	Linux OS	-0.0306	0.7364	0.058
High Severity	Critical Severity	0.00871	1.0087	0.691
Medium Severity	Critical Severity	-0.0214	0.8073	0.317
Low Severity	Critical Severity	-0.0250	0.7788	0.302
Mission Critical Service	Non-Mission Critical Services	-0.0531	0.5880	0.061

5.2.2 Investigating how Age Influences Vulnerability Remediations

The second data set we have accounts for vulnerability age. As the reader may recall from Table 2, we had a limited amount of information from the columns *CVE Code*, and *CVE Age*. We opted to subset the primary (master) data set by leaving out all rows where vulnerability age is "N/A". This secondary data set amounted to a total of 672 observations out of the original 2,645 possible observations.

In Tables 7, 8, 9, and 10 we assess how accounting for vulnerability age changes the prediction results.

Table 7 tells us that the older the vulnerability is, the shorter its overall survival rate. We find that for each unit increase in year, the hazard of death increases by 5%.

Table 7: Cox proportional hazards univariate analysis results for vulnerability age covariate.

Vulnerability Risk Factor	Reference Variable	Parameter	Hazard Ratio (95% CI)	p-value
Vulnerability Age	N/A	0.0049	1.0491	0.038

Table 8 tells us that after adjusting for the operating system type, vulnerability age has an even steeper rate of vulnerability remediation as age increases. Our precision also increased after we accounted for operating system type (p-value = 0.016). Finally, notice that although we lost precision due to the smaller sample size, the hazard of death for Windows OS servers are still close roughly 0.70 the hazard of death for Linux OS servers.

Table 8: Cox proportional hazards multivariate analysis results for covariates operating system type, and vulnerability age.

Vulnerability Risk Factor	Reference Variable	Parameter	Hazard Ratio (95% CI)	p-value
Windows OS	Linux OS	-0.03164	0.7288	0.201
Vulnerability Age	N/A	0.0628	1.065	0.016

Table 9 represents the results for vulnerability age after adjusting for mission-critical services. We found that for each unit increase in year of age, the hazard of death remains at about 5% per year.

Table 9: Cox proportional hazards multivariate analysis results for covariates mission critical services, and vulnerability age.

Vulnerability Risk Factor	Reference Variable	Parameter	Hazard Ratio (95% CI)	p-value
Mission-Critical Services	Non-Mission Critical Services	-0.3581	0.699	0.636
Vulnerability Age	N/A	0.0483	1.0495	0.043

In Table 10, We find that after adjusting for all three covariates, the hazard of death goes up to about 6.5% for every one year increase in vulnerability age.

Table 10: Cox proportional hazards multivariate analysis results for covariates operating system type, mission critical services, and vulnerability age.

Vulnerability Risk Factor	Reference Variable	Parameter	Hazard Ratio (95% CI)	p-value
Windows OS	Linux OS	-0.3162	0.7289	0.204
Mission Critical Service	Non-Mission Critical Services	-0.3482	0.7060	0.647
CVE Age	N/A	0.0628	1.0648	0.016

5.3 Trend Analysis

In this section, we provide some trend analysis on vulnerability severity levels, and operating system vulnerability reports based on the output from the National Institutes of Standards (NIST) National Vulnerability Database (NVD). Because this survival analysis is an observational study on one data set, looking at the "global" snapshot of vulnerability trends will further strengthen this study.

From Figure 2, it is clear that the operating system is a statistically strong predictor of vulnerability remediation rates and severity levels were not. We sought to understand if trends in the NVD may have affected these results.

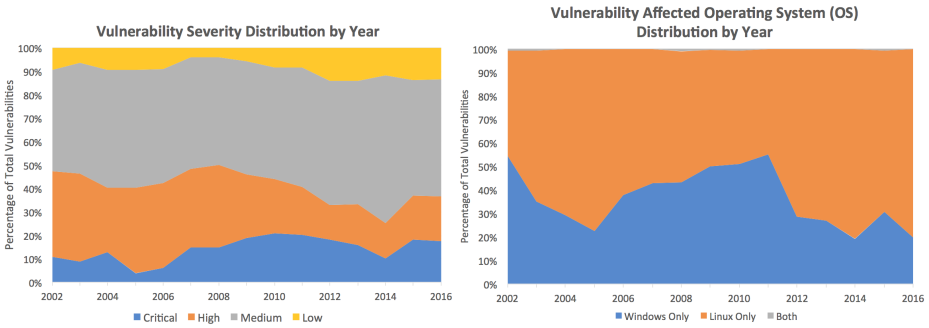

Figure 2: Trend analysis and distributions.

The charts above represent the division of vulnerabilities by severity and affected operating system per year. From Figure 2, one can see that vulnerability severity levels have been distributed consistently since 2002. Standard deviations are below 0.1 for each of category. Since it is consistent, it is unlikely that the vulnerability severity distribution is a confounding variable explaining its weak predictive power of remediation rates.

Table 11: The summary statistics representing the distribution of vulnerability severity since 2002 as reported in the National Vulnerability Database (NVD). These data are also represented on the right-hand side of Figure 2

Variable	Obs.	Mean	Std. Dev	Min	Max
Critical	15	0.1410	0.0523	0.0386	0.2098
High	15	0.2659	0.0867	0.1479	0.3749
Medium	15	0.4982	0.0443	0.4329	0.6302
Low	15	0.0949	0.0344	0.0395	0.1413

Table 12: The summary statistics representing the distribution of vulnerabilities as reported per operating system since 2002 as reported in the National Vulnerability Database (NVD). These data are also represented on the left-hand side of Figure 2

Variable	Obs.	Mean	Std. Dev	Min	Max
Windows Only	15	0.3643	0.1234	0.1925	0.5494
Linus Only	15	0.6323	0.1257	0.4472	0.8048
Both	15	0.0034	0.0038	0.0000	0.0103

From Figure 2, one can see that the distribution of operating systems that vulnerabilities affect has varied since 2002. The distribution of Windows and Linux vulnerabilities have standard deviations above 0.1. It is noteworthy that despite the changes in the vulnerability affected operating systems since 2002 that the operating system is still a strong predictor of remediation rates.

5.4 Takeaways

The data present a fascinating story. With respect to this observational analysis, we report the following findings:

1. *Operating system type provides the greatest predictive power for whether or not a vulnerability will be remediated. In short, we found Linux vulnerabilities are remediated at a faster rate than windows vulnerabilities. In fact, for every 67 Windows vulnerabilities which were remediated, 100 Linux vulnerabilities are remediated.*

2. *Severity levels do not presently drive decisions as to which vulnerabilities should be remediated first. Vulnerability survival times based on severity level are either a result of randomness, or, other confounding factors which are not a part of this study. Since vulnerability severity levels have had little variation since 2002, we also know it is unlikely that the distribution by which vulnerability severity scores are defined is a confounding factor.*

3. *Vulnerabilities on mission critical services have two-times the survival rates than that of non-mission critical services. In other words, our data show that mission-critical services are being remediated twice as slow. This may be due to the fact that mission critical services are slower to be patched due to their daily operational importance.*

4. *When we take into account vulnerability age, via a smaller data set, we find that the number of years since the date of vulnerability discovery (i.e. "vulnerability age") is the strongest predictor of vulnerability remediations. The "older" a vulnerability is, the more time there has been to develop the right patch for it and hence, the shorter its survival rate. We also find a 5% increase in likelihood the vulnerability will be remediated for every one year's increase in age.*

These takeaways offer a different perspective than what is known from other research results as reviewed in Sections 2.

6. CHALLENGES TO INTERNAL AND EXTERNAL VALIDITY

In this section, we review a number of identified challenges to internal and external validity. Recognizing and mitigating these challenges are critical to both generating and processing the data that produces reliable results and thorough scientific analysis. With that being said, we recognize that there are some challenges to the internal and external validity in our study.

6.0.1 Internal Validity

Internal validity refers to how well an experiment was conducted. Factors that support good internal validity are having as few confounding variables as possible. As the number of independent variables acting at the same time increases, so do the challenges to internal validity. Some of the challenges to internal validity for our study include servers being taken offline during data collection, scan policy changes, a one month gap in our data set, and potential software flaws in the vulnerability scanner.

Servers can temporarily be taken offline or decommissioned for various reasons. It may be the case that a system has to be upgraded, or access policies change. Temporary safeguards may need to be imposed if a "bug" is found on a database. In which case, the server is taken offline to prevent further damage until it can be investigated later. Additionally, scan policies can change over time. When an enterprise-level network undergoes periodic changes, such as permanently adding and deleting servers, host names either need to be added or deleted from the organization's current scan policy.

Finally, software flaws are problems in the software of the Nessus Scanner itself and can contribute to problems such as false-positives. There is a body of research aiming to address vulnerability software false-positives and false-negatives. We direct the curious reader to the work of Holm, et al.[5,12] and Doupe et al.[13]

6.0.2 External Validity

External validity refers to how well scientific results generalize to similar circumstances. We know that the less a scientific study generalizes to other studies, then the greater the challenges to external validity. The primary challenge to external validity for our study is that this is effectively an observational study from one organization. It is ideal to have data from multiple, and distinct organizations to improve the significance of our observations.

7. FUTURE WORK AND CONCLUSION

In this paper, we introduced the application of survival models to vulnerability remediation and patch deployment analysis on one year's worth of data from a twelve month period. We investigated the time-to-remediation of 2,645 observations of the occurrence of a unique vulnerability on a unique host. We sourced our data from an operational information security community that scans over 2,000 host services within two data centers per-month. We then performed a secondary analysis, investigating the total impact of vulnerability age on vulnerability remediation rates. There are three areas of opportunity for future work, namely:

1. Source more data from operational information security environments;
2. Use the Cox proportional hazards model to develop predictions under a different mix of distributions; and,

3. Devise a vulnerability triage schema, test it in an operational environment, and use this vulnerability survival analysis as a baseline metric.

Finally, to further investigate the methods for improved vulnerability triage, we could investigate an adjustment of behavior to address higher severity vulnerabilities sooner, or increase the rate at which vulnerabilities are remediated on mission critical services.

ACKNOWLEDGMENTS

This work was supported by the ARO MURI grant W911NF-13-1-0421. The authors are grateful to the DOD SMART Scholarship for student financial support, as well as Ben Priest for offering feedback, and Dartmouth Computing Services for providing the vulnerability data.

REFERENCES

[1] McQueen, M. A., McQueen, T. A., Boyer, W. F., and Chaffin, M. R., "Empirical estimates and observations of 0day vulnerabilities," in [*System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on*], 1–12, IEEE (2009).

[2] Nappa, A., Johnson, R., Bilge, L., Caballero, J., and Dumitras, T., "The attack of the clones: A study of the impact of shared code on vulnerability patching," (2015).

[3] Mell, P., Scarfone, K., and Romanosky, S., "A complete guide to the common vulnerability scoring system version 2.0," in [*Published by FIRST-Forum of Incident Response and Security Teams*], 1–23 (2007).

[4] Allodi, L. and Massacci, F., "Comparing vulnerability severity and exploits using case-control studies," *ACM Transactions on Information and System Security (TISSEC)* **17**(1), 1 (2014).

[5] Holm, H., Ekstedt, M., and Andersson, D., "Empirical analysis of system-level vulnerability metrics through actual attacks," *Dependable and Secure Computing, IEEE Transactions on* **9**(6), 825–837 (2012).

[6] Cavusoglu, H., Cavusoglu, H., and Zhang, J., "Security patch management: Share the burden or share the damage?," *Management Science* **54**(4), 657–670 (2008).

[7] Dempsey, K., Chawla, N. S., Johnson, L., Johnston, R., Jones, A. C., Orebaugh, A., Scholl, M., and Stine, K., [*Information security continuous monitoring (ISCM) for federal information systems and organizations*] (2011).

[8] Afful-Dadzie, A. and Allen, T. T., "Data-driven cyber-vulnerability maintenance policies," *Journal of Quality Technology* **46**(3), 234 (2014).

[9] Afful-Dadzie, A. and Allen, T. T., "Control charting methods for autocorrelated cyber vulnerability data," *Quality Engineering* **28**(3), 313–328 (2016).

[10] Bull, K. and Spiegelhalter, D. J., "Tutorial in biostatistics survival analysis in observational studies," *Statistics in medicine* **16**(9), 1041–1074 (1997).

[11] Harrell, F., [*Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis*], Springer (2015).

[12] Holm, H., Sommestad, T., Almroth, J., and Persson, M., "A quantitative evaluation of vulnerability scanning," *Information Management & Computer Security* **19**(4), 231–247 (2011).

[13] Doupé, A., Cavedon, L., Kruegel, C., and Vigna, G., "Enemy of the state: A state-aware black-box web vulnerability scanner.," in [*USENIX Security Symposium*], 523–538 (2012).