Forecasting Zero-Day Vulnerabilities

David Last
United States Air Force
425 Brooks Road
Rome, NY 13441
david.last.1@us.af.mil

ABSTRACT

It seems that computer network defenders are always two steps behind attackers. This is due in part to the need for defenders to protect against the exploitation of zero-day vulnerabilities which they may not yet know exist. If network defenders were able to forecast the location and severity of zero-day vulnerabilities that would be discovered in the near future, this would be a valuable tool. This paper describes ongoing research that seeks to develop Vulnerability Discovery Models that will provide forecasts for zero-day vulnerability discovery rates. The initial work addresses forecasts at the global and category (web browser, operating system, and video player) levels, and this will be extended to individual software applications in the future. This research has developed three distinct zero-day vulnerability forecast suites, one based on regression and two based on machine learning. The accuracy of several of the forecast models from each forecast suite is evaluated, and the results are promising for the future development of these forecast models. Future work in this area will involve combining individual forecast models into a consensus forecast model, as well as extending the forecast models to the software application level.

CCS Concepts

• Security and privacy → Systems security → Vulnerability management

Keywords

Zero-day vulnerability; vulnerability forecasting; vulnerability forecasting model

1. INTRODUCTION

Computer network defenders are locked in an ongoing struggle against attackers. The defenders are at an extreme disadvantage, since they must defend against exploits of zero-day vulnerabilities which have either not yet been discovered or have been discovered by attackers but are as yet unknown to the defenders. Facing this situation, defenders would benefit immensely from having some idea about the probable location and severity of zero-day vulnerabilities. A number of researchers have addressed the problem of modeling software vulnerability discovery through the use of Vulnerability Discovery Models (VDM) [1] [2] [3]. However, past work has only applied these VDMs to a few specific software applications and did not explore their capacity for forecasting zero-day vulnerabilities. The work detailed in this paper will expand on the work of previous researchers and apply these VDMs to a broad range of software applications to test their ability to forecast zero-day vulnerabilities. This research will begin by generating vulnerability discovery forecasts at the global and category (web browser, operating system, and video player) levels. Vulnerability discovery trends may be easier to identify at these levels, and thus provide insight into forecasts at the software application level. Additionally, discovery rates across a category may be indicative of attacker interest in these areas; this attacker interest can serve as an input into the application-level vulnerability discovery models. Later, these models will be extended to the software application level, where they will incorporate results from the work of other researchers.

What benefits would such a forecasting ability provide? Operating on the belief that software presenting more zero-day vulnerabilities is more likely to be attacked and successfully exploited, network defenders can deploy cyber sensors or cyber defenses that provide protection against a broad range of vulnerabilities (such as firewalls) in areas with high-vulnerability-forecast software. Alternatively, network defenders might choose to deploy different software applications on their enterprise networks based on the forecast zero-day vulnerabilities. These forecasts can inform security/functionality tradeoffs when institutions are designing their networks. Network defenders can also make long-term plans for allocating software patching and maintenance resources based on where new mission-critical vulnerabilities are expected to appear.

2. VULNERABITIY FORECAST MODELS

The vulnerability forecast models developed under this research seek to identify historical trends in the number and severity of discovered software vulnerabilities, and to extend these trends to forecast zero-day vulnerability discovery 12-24 months into the future. A number of different vulnerability forecast models are used, and the forecasts generated by these models are combined into a consensus forecast that should perform more accurately over time than any individual model. The data used to train these models is taken from the National Vulnerability Database (NVD), which is maintained by the National Institute of Standards and Technology (NIST) [4]. The NVD maintains records of almost all software vulnerabilities that are publically reported; these records include information on the severity and ease of exploit for these vulnerabilities. For the purposes of forecasting vulnerabilities at the global and category level, vulnerabilities are arranged into four datasets. The GlobalVulns dataset contains all software vulnerabilities recorded in the NVD; the BrowserVulns, OSVulns, and VideoVulns datasets contain all vulnerabilities affecting the software indicated in Table 1. These datasets can be formatted in terms of cumulative vulnerabilities (all vulnerabilities affecting the relevant software applications over the entire period covered by the NVD) or monthly vulnerabilities.

This research has developed three suites of forecast models. The first suite is called the Composite Regression suite. This suite of models utilizes linear and quadratic regression over cumulative vulnerabilities to identify short- and long-term trends in software

CISRC '16, April 05-07, 2016, Oak Ridge, TN, USA
ACM 978-1-4503-3752-6/16/04
DOI: http://dx.doi.org/10.1145/2897795.2897813

Table 1. Software Applications included in NVD analysis.

BrowserVulns	OSVulns	VideoVulns
Microsoft Internet Explorer Mozilla Firefox Google Chrome Apple Safari Opera Mozilla Seamonkey	Microsoft Windows Apple OS X Ubuntu Linux Debian Linux FreeBSD HP HP-UX IBM AIX Linux Kernel NetBSD Linux Redhat Linux kernel Sun Solaris / OpenSolaris Novell Suse / OpenSuse Linux	RealNetworks RealPlayer Adobe FlashPlayer FFmpeg Adobe ShockWave Apple QuickTime Windows MediaPlayer VideoLAN VLC Media Player

vulnerability discovery and incorporate those trends into zero-day vulnerability discovery forecasts. The Composite Regression suite uses three basic models. The *linear regression* model assumes that vulnerability discovery rates will remain unchanged over the training/forecast period, and the *quadratic regression* model assumes those rates will increase or decrease over the training/forecast period. The *combined regression* model combines these two models. These regression models are fit to cumulative vulnerability data. These three basic models can be used in four different modes to identify short- and long-term trends and incorporate them into forecasts. The *simple time horizon* mode uses a single regression model to generate the forecast. Recognizing that different short- and long-term trends may affect future performance, the *disjoint time horizon* mode fits regression models to short-, mid-, and long-term training data (e.g. 25%, 50%, and 100% of the available training data, respectively) and then uses these models to generate short-, mid-, and long-term forecasts and piece them together. This mode, however, ignores the fact that long-term trends may affect short-term future performance, and vice versa. The *homogenous time horizon* mode addresses this shortfall by averaging the short-, mid-, and long-term forecasts together to generate a single forecast. Still, this mode neglects the likely situation where short-term trends will affect short-term future performance more heavily than long-term trends will, etc. Therefore, the *proportional time horizon* mode generates short-, mid-, and long-term forecasts by averaging short-, mid-, and long-term models using appropriate weights. Given three basic models and four modes, as well as using 12, 24, 36, and 48 months of historical data for model training, there are 48 different forecast models in the Composite Regression suite.

The second forecast suite is Machine Learning over Cumulative Vulnerabilities. This suite utilizes the Weka machine learning toolkit [5] to analyze historical data between January 2000 and the start of the forecast period. Forecasts are generated based on the historical patterns identified by Weka. Different models in this suite are created using raw data or data that is smoothed using 3- and 5-month sliding window averages. Additionally, independent data that may affect zero-day vulnerability discovery rates are used as overlay data for the machine learning training; software application version release dates are used for this purpose, and future work will include other data such as Google search trends.

The third forecast suite is Machine Learning over Monthly Vulnerabilities. This suite is similar to the previously mentioned suite, except that it trains and forecasts over monthly software vulnerability data rather than cumulative data. It is more useful for certain datasets where patterns are more identifiable in the monthly data. This suite also uses data smoothing and overlay data for different models.

3. RESULTS

As part of this research, the individual forecast models in the previously-discussed forecast suites were evaluated for their accuracy in forecasting over the GlobalVulns, BrowserVulns, OSVulns, and VideoVulns datasets. Historical vulnerability discovery data spanning from January 2000 to December 2011 was used as training data, and the forecast models were evaluated over multiple forecasts covering the period from January 2012 to December 2015. Each model was evaluated over 12-, 18-, and 24- month forecast periods using all available previous historical data as training data (Figure 1). The performance of each model was evaluated in terms of the Root Mean Square Percent Error (RMSPE) between the cumulative vulnerability forecast and true data over each forecast period. The median of the RMSPE over all forecast periods in the January 2012-December 2015 timeframe (e.g. January 2012-December 2012, February 2012-January 2013, etc. for 12-month forecasts) is taken as the metric for comparing forecast model performance. It should be noted here that there is no cross-validation in these preliminary results; these results merely show the forecast performance of different forecast models. In future work, models will be trained over a certain time period, their performance will be evaluated over a subsequent validation period, and then the best-performing models will be combined to generate a consensus forecast, which will be evaluated over a subsequent test period.

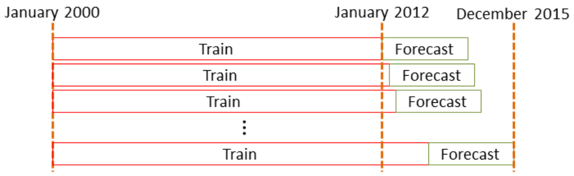

Figure 1. Training/forecast periods for forecast model evaluation.

Due to space limitations in this paper, the performance of only a few forecast models from each of the three forecast suites are reported here. For the Composite Regression suite, the best-, 12th best-, and 24th best (median)-performing forecast models for that particular dataset are reported (Models 1a-1c). For the Machine Learning over Cumulative (Models 4-6) and Monthly (Models 7-9) Vulnerabilities suites, results from models demonstrating the effects of no data smoothing/ no overlay data, as well as the effects of data smoothing and software application version release date overlay data, are reported. Software application version release dates are represented as overlay data by generating data corresponding to expected vulnerability discovery rates for multi-version software based on the Alhamzi-Malaiya model [2]. For the BrowserVulns, OSVulns, and VideoVulns datasets, version release date data for the covered software applications (Table 1) are used as overlay data; for the GlobalVulns dataset, version release data from the three other datasets are used as overlay data. However, since the GlobalVulns dataset covers many more software applications than those covered by the other three datasets, forecasts made using this overlay data for GloablVulns are expected to demonstrate poor performance. Table 2 details the specifics of the models whose results are reported.

Table 2. Forecast models with reported results.

Model #	Description
1a	Best performing Composite Regression model for this dataset
1b	Worst of 1st quartile (12th best) Composite Regression model for this dataset
1c	Worst of 2nd quartile (24th best, i.e. median) Composite Regression model for this dataset
4	Machine Learning over Cumulative Vulnerabilities, no smoothing, no overlay data
5	Machine Learning over Cumulative Vulnerabilities, 3-month sliding window average smoothing, no overlay data
6	Machine Learning over Cumulative Vulnerabilities, 3-month sliding window average smoothing, version release date overlay data
7	Machine Learning over Monthly Vulnerabilities, no smoothing, no overlay data
8	Machine Learning over Monthly Vulnerabilities, 3-month sliding window average smoothing, no overlay data
9	Machine Learning over Monthly Vulnerabilities, no smoothing, version release date overlay data

The forecast results are displayed in Figure 2 - Figure 5. In order to understand these results, it should be noted that manual inspection of the cumulative vulnerability datasets indicates that the GlobalVulns dataset is relatively smooth, with each successive dataset becoming less smooth. The VideoVulns dataset is the least smooth, displaying a number of sudden jumps in vulnerability discovery, interspersed with periods of relatively low discovery rates. As a result, we expect that the accuracy of forecasts will decrease over subsequent datasets. Also, it should be noted that while these results report the best-performing (Model 1a) and 25th percentile (Model 1b) models from the Composite Regression suite, the identities of the best performers are not known a priori; therefore the Machine Learning models should be evaluated in comparison to the 2nd quartile of Composite Regression models (range between Models 1b and 1c). A brief analysis of the results indicates that the 2nd quartile of the Composite Regression model suite generally performs similarly or better than the Machine Learning models. For the GlobalVulns, OSVulns, and VideoVulns datasets, eliminating obvious outlier forecasts (e.g. not using smoothing or overlay data for the GlobalVulns cumulative or OSvulns cumulative or monthly datasets) makes the Machine Learning forecast models comparable to the 1st and 2nd quartile Composite Regression models. However, for the BrowserVulns dataset, the 2nd quartile of the Composite Regression suite performs equally or better than the Machine Learning models, sometimes much better.

The use of data smoothing returns mixed results; it generally returns worse forecast results when used on Machine Learning Monthly forecast models (Model 8). For the Machine Learning Cumulative forecast models (Models 5 and 6 vs. Model 4), data smoothing returns better forecasts for the BrowserVulns and VideoVulns datasets, but it returns worse forecasts for the GlobalVulns and OSVulns datasets. The use of version release date overlay data returns worse forecasts for the GlobalVulns cumulative dataset (Figure 2, Model 6 vs. Model 5), while it returns small mixed results for all other datasets. This result is expected; since the GlobalVulns dataset encompasses many more software applications than those contained in the BrowserVulns, OSVulns, and VideoVulns datasets (Table 1), it is expected that the zero-day vulnerability discovery rate for that dataset would not correlate strongly with the version release dates of a few software applications. It should also be noted that preliminary experiments indicate that the use of version release date overlay data for forecasts over individual software applications returns better forecast improvements than for forecasts over software

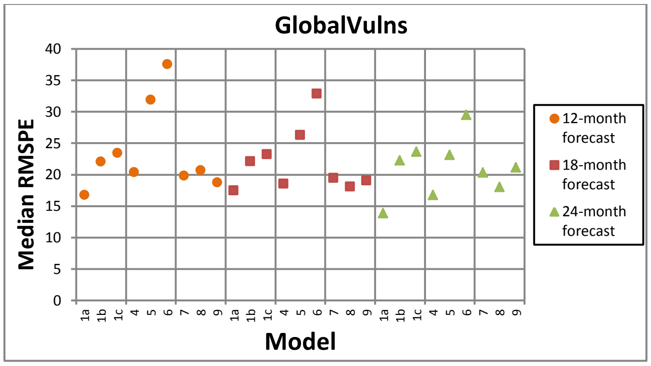

Figure 2. Median RMSPE for forecasts on GlobalVulns dataset.

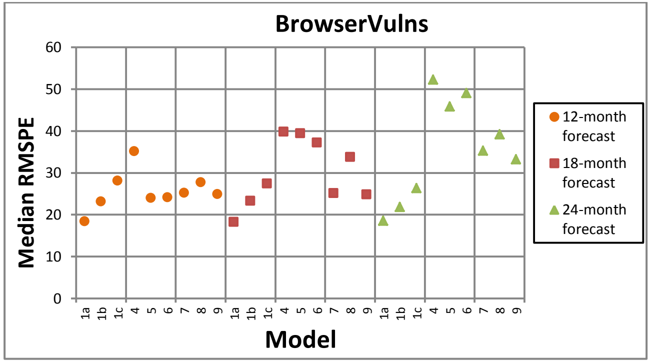

Figure 3. Median RMSPE for forecasts on BrowserVulns dataset.

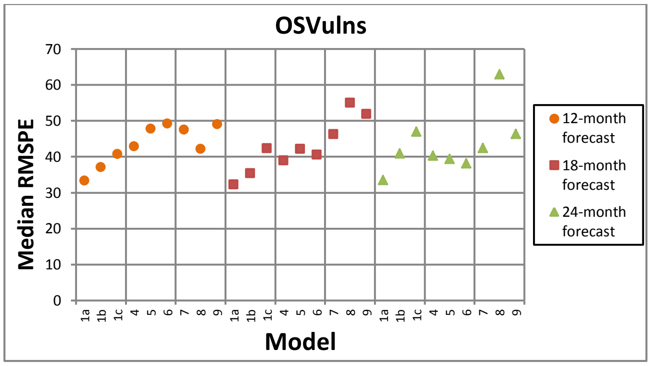

Figure 4. Median RMSPE for forecasts on OSVulns dataset.

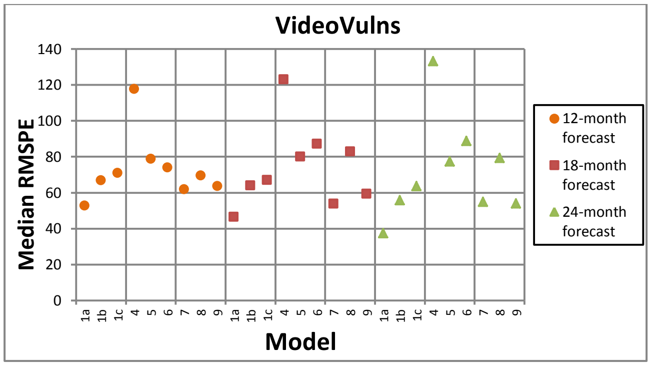

Figure 5. Median RMSPE for forecasts on VideoVulns dataset.

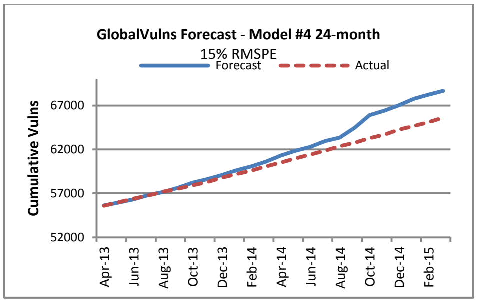

Figure 6. 15% RMSPE forecast on GlobalVulns.

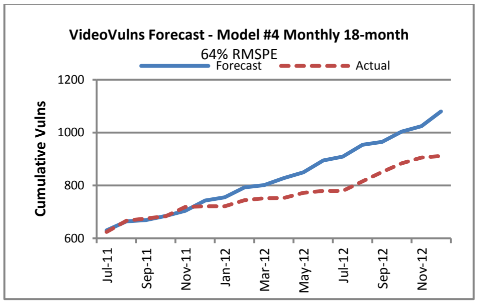

Figure 7. 64% RMSPE forecast on VideoVulns.

application categories. Future research will explore the usefulness of version release date overlay data in that context. Finally, there is no strong advantage between the Machine Learning Cumulative and Monthly models as long as no smoothing or overlay data is used on the cumulative dataset. For the OSVulns dataset, the Machine Learning Cumulative models provide better forecasts (Figure 4, Model 4 vs. Models 7-9), whereas for the BrowserVulns and VideoVulns datasets, the Machine Learning Monthly models generate better forecasts (Models 7-9 vs. Model 4).

The RMSPE statistics in Figure 2 through Figure 5 may be difficult to grasp intuitively; it may be difficult to assess what a “good” RMSPE is. In order to provide some context, Figure 6 and Figure 7 illustrate forecasts with 15% and 64% RMSPE, respectively.

4. FUTURE WORK

This research will be continued to pursue the goal of forecasting software vulnerability discovery for individual software applications. The first step will be to enhance the forecasts at the global and category level. While some of the forecast models in the different forecast suites will perform better than others, even well-performing models will perform badly under certain circumstances. A well-established technique for addressing this issue is to average all of the model forecasts into a single consensus forecast model. This consensus model should smooth out the poor individual forecasts for each model and produce forecasts that, over time, perform better than forecasts from any individual forecast model. One way to generate a forecast model is to average all of the individual model forecasts together with equal weights; however, this ignores the fact that some forecast models may be more accurate over time than others. This research will leverage an accepted averaging method called Quantile Regression Averaging (QRA) [6] to generate consensus forecasts from individual model forecasts. QRA will generate consensus forecasts that are weighted according to the recent forecast performance of the component models. Additionally, QRA can be used to generate prediction intervals around the forecast, which give upper and lower bounds around the forecast and a percent probability that the true data will fall between those bounds.

This research will also develop forecast models for vulnerability discovery in individual software applications. Previous work in this area has developed a number of different Software Vulnerability Models, which describe the vulnerability discovery rate over the lifetime of a software application. This research will use the single software application vulnerability discovery model [1] and the multi-version software application model [2] developed by Alhazmi, Malaiya, and others to adapt the forecast models discussed in this paper for individual software applications. Additionally, this research will utilize machine learning to identify and predict trends in the *A*, *B*, and *C* parameters of the Alhazmi-Malaiya model over the lifetime of a software application in order to generate more accurate forecasts.

5. CONCLUSION

Network defenders face a disadvantage against attackers, since the defenders must protect against zero-day vulnerabilities of which they are not yet aware. They would benefit greatly from an ability to forecast the number and severity of zero-day vulnerabilities that will be discovered in the near future. The research presented in this paper is developing Vulnerability Discovery Models that can be used to generate these forecasts. These models are organized into three distinct forecast model suites, and they are currently being applied at the global and category (web browser, operating system, and video player) levels. The performance of some of these forecast models is evaluated in this paper, and the results show promise for future use of these models. Future development of these models will include a consensus forecast model that combines the individual models into a larger model with better long-term accuracy. These models will also be extended an adapted to forecast vulnerabilities at the software application level. This research will provide an additional tool to aid cyber defenders against attacks.

6. REFERENCES

[1] O. H. Alhazmi and Y. K. Mawlaiya, "Modeling the Vulnerability Discovery Process," in *The 16th IEEE International Symposium on Software Reliability Engineering (ISSRE'05)*, Washington, DC, 2005.

[2] J. Kim, Y. Malaiya and I. Ray, "Vulnerability discovery in multi-version software systems," in *High Assurance Systems Engineering Symposium, 2007. HASE'07. 10th IEEE*, IEEE, 2007, pp. 141-148.

[3] S. Zhang, D. Caragea and X. Ou, "An empirical study on using the national vulnerability database to predict software vulnerabilities," *Database and Expert Systems Applications*, pp. 217-231, 2011.

[4] "National Vulnerability Database," National Institute of Standards and Technology (NIST), [Online]. Available: https://nvd.nist.gov/. [Accessed 8 December 2015].

[5] "Weka 3: Data Mining Software in Java," [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/. [Accessed 1 December 2015].

[6] K. Maciejowska, J. Nowotarski and R. Weron, "Probabilistic forecasting of electricity spot prices using Factor Quantile Regression Averaging," *International Journal of Forecasting* , 2015.