Proactive Identification of Exploits in the Wild Through Vulnerability Mentions Online

Mohammed Almukaynizi, Eric Nunes, Krishna Dharaiya,
Manoj Senguttuvan, Jana Shakarian, Paulo Shakarian
Arizona State University
Tempe, AZ 85281, USA
Email: {malmukay, enunes1, krishna.dharaiya, mbalasu9, jshak, shak} @asu.edu

***Abstract*—The number of software vulnerabilities discovered and publicly disclosed is increasing every year; however, only a small fraction of them is exploited in real-world attacks. With limitations on time and skilled resources, organizations often look at ways to identify threatened vulnerabilities for patch prioritization. In this paper, we present an exploit prediction model that predicts whether a vulnerability will be exploited. Our proposed model leverages data from a variety of online data sources (white-hat community, vulnerability researchers community, and darkweb/deepweb sites) with vulnerability mentions. Compared to the standard scoring system (CVSS base score), our model outperforms the baseline models with an F1 measure of 0.40 on the minority class (266% improvement over CVSS base score) and also achieves high True Positive Rate at low False Positive Rate (90%, 13%, respectively). The results demonstrate that the model is highly effective as an early predictor of exploits that could appear in the wild. We also present a qualitative and quantitative study regarding the increase in the likelihood of exploitation incurred when a vulnerability is mentioned in each of the data sources we examine.**

***Keywords*—** vulnerability exploit prediction; online vulnerability mentions; darkweb analysis; adversarial machine learning

I. Introduction

An increasing number of software vulnerabilities are discovered and publicly disclosed every year. In 2016 alone, more than 10,000 vulnerability identifiers were assigned and at least 6,000 were publicly disclosed by the National Institute of Standards and Technology (NIST)[1]. However, only a small fraction of those vulnerabilities (less than 3%) are found to be exploited in the wild [1], [2], [3], [4] - a result confirmed in this paper. The current methods for prioritizing patching vulnerabilities appear to fall short. Verizon reported that over 99% of breaches are caused by exploits to known vulnerabilities [5].

In this paper, we examine the ability to predict whether a vulnerability will be exploited in the wild using supervised machine learning techniques. Our models are trained/tested on features derived from cyber threat intelligence feeds comprised of a variety of sources, a task that has been previously studied by [4], which used data feeds from Twitter - though recent work suggests this method is not viable in practice [6]. This problem is of direct relevance to patch prioritization. Other previous work on vulnerability online mentions either studies the correlation between these feeds and the existence of exploits in the wild [2], [7], or develops machine learning models to predict the existence of Proof-of-Concept (PoC) exploits [1], [8], [6]. However, only a small fraction of the vulnerabilities having PoCs is found to be exploited in the wild.

After reviewing the literature, especially studies on data gathered from darkweb and deepweb (DW) [9], [10], [2], [11], [12], and after over one hundred interviews with professionals working for managed security service providers (MSSP's), firms specializing in cyber risk assessment, and security specialists working for managed (IT) service providers (MSP's), we identified three sources of data representative of current threat intelligence used for vulnerability prioritization: (1) ExploitDB (EDB)[2] contains information on PoCs for vulnerabilities provided by security researchers from various blogs and security reports, (2) Zero Day Initiative (ZDI)[3] is curated by a commercial firm called TippingPoint and uses a variety of reported sources focused on disclosures by various software vendors and their security researchers, and (3) a collection of information scraped from over 120 sites on the darkweb and deepweb (DW) sites from a system originally introduced in [13], [14]. The intuition behind each of these feeds was to not only utilize information that was aggregated over numerous related sources, but also to represent feeds commonly used by cybersecurity professionals.

Specifically, the contributions of this paper include:

- We demonstrate the utility of the developed machine learning models in predicting exploits in the wild with True Positive Rate (TPR) of 90% while maintaining the a False Positive Rate (FPR) of less than 15%. We also examined the performance of variants of our model when we control for temporal mixing and in the case where only a single source is used.
- We report the increase in the vulnerability exploitation likelihood for vulnerability mentions on EDB (9%), ZDI (12%), and DW (14%) over vulnerabilities only disclosed

[1]https://www.nist.gov/

[2]https://www.exploit-db.com/

[3]http://www.zerodayinitiative.com/

2017 International Conference on Cyber Conflict U.S.
978-1-5386-2379-4/17/$31.00

on NVD (2.4%). We also analyze exploited vulnerabilities based on language used in DW. We specifically noted that Russian language sites on DW discuss vulnerabilities that are 19 times more likely to be exploited than random, which was greater than other languages examined.

The rest of the paper is organized as follows: In Section II we provide an overview of our supervised machine learning model and describe our data sources. Analysis on the exploited vulnerabilities is discussed in Section III. In Section IV we provide experimental results for predicting exploits. Finally, related work is discussed in Section VI.

II. Exploit Prediction Model

An overview of the proposed exploit prediction model is provided in Figure 1. The model consists of the following three phases,

- **Data Collection:** We utilize three data sources in addition to NVD. These data sources are EDB (ExploitDB), ZDI (Zero Day Initiative) and data mined from DW (darkweb and deepnet) markets and forums focusing on malicious hacking. We assign ground truth for our binary classification problem using Symantec attack signatures that detect exploits in the wild targeting disclosed vulnerabilities. We discuss the data sources in Section II-A.
- **Feature Extraction:** We extract features from each of the data sources. The features include bag of words features for vulnerability description and discussions on the DW, binary features which checks for the presence of PoC exploits in EDB, vulnerability disclosures in ZDI and DW. We also include additional features from NVD namely, CVSS score, CVSS vector.
- **Prediction:** We perform binary classification on the selected features indicating whether the vulnerability will be exploited or not. To address this classification problem, we evaluate several standard supervised machine learning approaches.

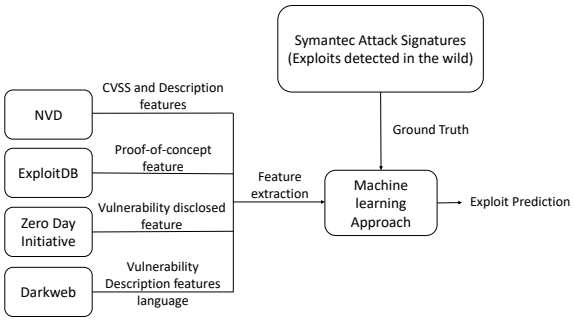

Fig. 1. Exploit Prediction Model.

A. Data Sources

NVD. The National Vulnerability Database maintains a database of publicly disclosed vulnerabilities, each one is identified using an unique CVE-ID. We collect vulnerabilities disclosed between 2015 and 2016. Our dataset is comprised of 12,598 vulnerabilities. Figure 2 shows the month-wise disclosure of vulnerabilities. At the time of data collection there were only 30 vulnerabilities disclosed in December 2016, hence the small bar at the end of 2016. For each vulnerability, we gather its description, and CVSS base score and vector.

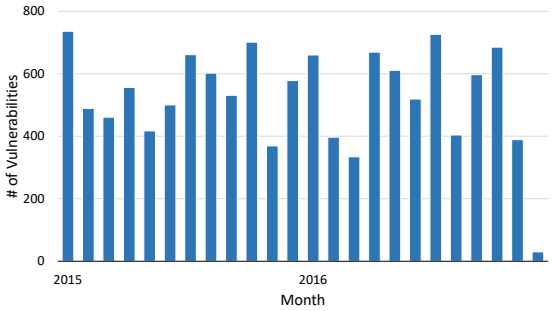

Fig. 2. Vulnerabilities disclosed per month.

EDB (white-hat community). Exploit Database is a collection of PoC exploits maintained by Offensive Security Training that has CVE's associated with available exploits. Using the unique CVE-ID's from the NVD database for the time period between 2015 and 2016, we query the EDB to find whether a PoC exploit is available. We also record the date the PoC exploit was posted, for our experiments. For the set of vulnerabilities we consider, 799 of the vulnerabilities were found to have verified PoCs.

ZDI (vulnerability detection community). Zero Day Initiative, launched by TippingPoint, maintains a database of vulnerabilities submitted by security researchers. Monetary incentive is provided if the vulnerability is verified to the researcher. ZDI then notifies the vendor to develop patches for the reported vulnerability before public disclosure. We query the ZDI database to collect information regarding vulnerabilities that are disclosed by ZDI. Between 2015 and 2016, the query returned 824 vulnerabilities common between NVD and ZDI.

DW (black-hat community). We summarize the data collection infrastructure described in [13]. In this paper, the authors built a system that crawls sites on DW, both marketplaces and forums, to collect data relating to malicious hacking. They first identify sites before developing scripts for automatic data collection. A site is put forward to script development after it has been established that the content is of interest (hacking-related) and relatively stable. Using a machine learning approach with high accuracy, data related to malicious hacking is filtered from the irrelevant postings (e.g., carding, pornography) and added to a database. Not all exploits or vulnerability items in the database have a CVE number associated with them. For our study, we only keep the items/posts with explicit vulnerability mentions. Some vulnerabilities are mentioned in DW using Microsoft Security Bulletin Number[4] (e.g., MS16-006) we map every bulletin number to its corresponding CVE-ID, making ground truth assignment easy. These items/posts can be both products sold on markets as well as posts extracted from forums discussing topics relating to malicious hacking. We find 378 unique CVE mentions between 2015 and 2016

[4]https://www.microsoft.com/en-us/download/details.aspx?id=36982

from more than 120 DW websites. We also query the posting date and descriptions associated with all the CVE mentions including product title and description, entire discussion with the CVE mention, and the topic of the discussion.

Attack Signatures (Ground truth). For our ground truth, we identified vulnerabilities that were exploited in the wild using Symantec's anti-virus attack signatures[5] and Intrusion Detection Systems' (IDS) attack signatures[6] for attacks detected within the timeframe of our study. Some attack signatures are associated with the CVE-ID of the vulnerability that was exploited. We map these CVE's to the CVE's mined from NVD, EDB, ZDI and DW. This ground truth indicates actual exploits that were used in the wild and not just PoC exploits. However, this ground truth is found to be biased towards reporting exploits targeting vulnerabilities that exist in, or that can run on, software products from certain vendors (e.g., Microsoft, Adobe) [4]. Table I shows the number of vulnerabilities exploited as compared to the ones disclosed in 2015 and 2016 for all the data sources considered. For NVD, around 2.4% of the disclosed vulnerabilities were exploited, which is consistent with previous literature. Additionally, we define the *exploitation date* of a vulnerability as the date it was first detected in the wild. Symantec IDS attack signatures are reported without recording the dates when they were first detected ($n = 112$), but anti-virus attack signatures are reported with their *exploitation date* ($n = 194$).

TABLE I
NUMBER OF VULNERABILITIES (2015-2016)

Database	Vulnerabilities	Exploited	% Exploited
NVD	12,598	306	2.4%
EDB	799	74	9.2%
ZDI	824	95	11.5%
DW	378	52	13.8%

B. Feature Description

We combine features from all the data sources discussed in Section II-A. Table II gives a summary of the features with their type. We now discuss each of the features.

NVD and DW description. NVD description provides information on the vulnerability and the capabilities attackers will gain if they exploit it. DW description often provides rich context about the discussion (mostly in forums rather than marketplaces since items are described in fewer words). This description was appended to the NVD description with the corresponding CVE. We observed that some of the descriptions on DW are in foreign languages as discussed in Section III. We first translate the foreign text to English using Google Translate API[7]. We then vectorize the text features using Term Frequency-Inverse Document Frequency (TF-IDF) model, which creates a vocabulary of all the words in the description. The importance of a word feature increases the more times it occurs, but it is normalized by the total number of words in the description. This eliminates common words from being important features. We limit our TF-IDF model to the 1000 most frequent words in the entire dataset (we did not observe performance improvement using additional words with this method).

TABLE II
SUMMARY OF FEATURES.

Feature	Type
NVD and DW description	TF-IDF on bag of words
CVSS	Numeric and Categorical
DW Language	Categorical
Presence of Proof-of-Concept	Binary
Vulnerability mention on ZDI	Binary
Vulnerability mention on DW	Binary

CVSS. NVD provides us with the CVSS base scores, and the CVSS vectors. These metrics indicate the severity of each disclosed vulnerability[8]. We use the CVSS base score version 2.0 as a feature for our classifier (numeric type). The CVSS base vector lists the components from which the score is computed. The components of the vector include Access Complexity, Authentication, Confidentiality, Integrity and Availability. Access complexity indicates how difficult it is to exploit the vulnerability once the attacker has gained access. It is defined in terms of three levels: High, Medium and Low. Authentication indicates whether authentication is required by the attacker to exploit the vulnerability. It is a binary identifier taking the values Required and Not Required. Confidentiality, Integrity and Availability indicate what loss the system would incur if the vulnerability is exploited. It takes the values None, Partial and Complete. All the CVSS vector features are categorical. We vectorize these features by building a vector with all possible categories. Then if that category is present we insert "1" otherwise "0".

Language. DW feeds are posted in different languages. We found 4 languages that are used in DW posts referencing vulnerabilities. These languages are English, Chinese, Russian, and Swedish. Since we have a limited number of samples from every language, we use the text translation as we described. However, translation can result in a loss of important information, but we can retain the impact of knowing the language by using it as feature. We show analysis on the languages of DW feeds and their variation in the exploitation rate in section III.

Presence of Proof-of-Concept (PoC). The presence of PoC exploits in EDB increases the likelihood of a vulnerability

[5]https://www.symantec.com/security_response/landing/azlisting.jsp
[6]https://www.symantec.com/security_response/attacksignatures/
[7]https://cloud.google.com/translate/docs/

[8]We do not consider the temporal metrics because they are undefined for the majority of the vulnerabilities we study.

being exploited. We treat it as a binary feature indicating whether PoC is present for a vulnerability or not.

Vulnerability mention on ZDI. Vulnerability mention on ZDI also increases the likelihood of it being exploited. Similar to PoC exploit we use a binary feature to denote whether a vulnerability was mentioned (disclosed) in ZDI before it is exploited.

Vulnerability mention on DW. Vulnerability mention on DW also increases the likelihood of it being exploited. Binary feature indicating vulnerability mention on DW is considered as a feature.

III. Vulnerability and Exploit Analysis

To assess the importance of aggregating different data sources for the early prediction of threatened vulnerabilities, we first analyze the likelihood that a given vulnerability mentioned in different data sources will be exploited, as well as the coverage of each of these sources. Then we provide a language-based analysis on the DW data to shed light on some socio-cultural factors present in DW sites and which appear to have implications on the exploitability likelihood.

Likelihood of Exploitation. The likelihood of a vulnerability mentioned in each of the data sources we consider being exploited is presented in Table III. As already mentioned, around 2.4% of the vulnerabilities disclosed in NVD are exploited in the wild. Intuitively, by including other sources, we increase the likelihood of correctly predicting vulnerabilities that will be exploited. Furthermore, we observe that each data source covers different set of exploited vulnerabilities, with few vulnerabilities appeared in more than one source, which confirms our hypothesis that the underlying communities are highly distinct (i.e., EDB is a white-hat community, ZDI is a vulnerability researcher community, and DW sites are dominated mostly by hackers).

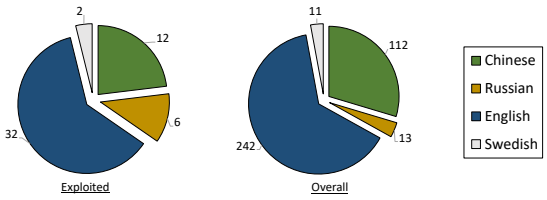

Fig. 3. Number of the *exploited* vulnerabilities mentioned by each language (left), and number of vulnerabilities mentions in each language (right).

Language-Based Analysis. Interestingly, we found notable variations on the exploitation likelihood conditioned on the language used on DW data feeds referencing CVE's. In DW feeds, four languages are detected with different vulnerability posts and items distributions. Expectedly, English and Chinese have far more vulnerability mentions ($n = 242$, and $n = 112$, respectively) than Russian and Swedish ($n = 13$, and $n = 11$, respectively). However, vulnerabilities mentioned in Chinese postings exhibit the lowest exploitation rate. For example, of those vulnerabilities, only 12 are exploited (about 10%) while 32 of the vulnerability mentioned on English postings are exploited (about 13%). Although vulnerability mentions in Russian or Swedish postings are few, these vulnerabilities exhibit very high exploitation rates. For example, about 46% of the vulnerabilities mentioned in Russian were exploited ($n = 6$), and about 19% for vulnerabilities mentioned in Swedish ($n = 2$). Figure 3 shows the number of vulnerability mentions by each language as well as the number of *exploited* vulnerabilities mentioned by each language.

IV. Experimental Setup

To determine the effectiveness of our early prediction approach, we examine different supervised machine learning (ML) algorithms (e.g., Support Vector Machine (SVM), Random Forest (RF), Naive Bayes Classifier (NB), Logistic Regression (LOG-REG)) on the features summarized in Section II-B. For our model, we found that Random Forest (RF) gives us the best F1 measure. Random forest is an ensemble method proposed by Breiman [15]. It is based on the idea of generating multiple decision trees each having their own independent opinion on class labels (*exploited* or *not exploited*), which are then used in combination to classify a new disclosed vulnerability. To take the final decision, a majority vote is taken and the class label with most votes is assigned to the vulnerability.

We conduct our experiments after restricting the training samples to the vulnerabilities published before those published in the testing samples. Also, we only use data feeds that are present before the *exploitation date*. Thus, we guarantee that our experimental settings resemble the real-world conditions with no temporal-intermixing as discussed in [6]. Furthermore, since we cannot make any assumptions regarding the sequence of events for the exploited vulnerabilities reported by Symantec without the *exploitation date* ($n = 112$), we remove these vulnerabilities from our experiments. Hence, the fraction of exploited vulnerabilities becomes 1.2%. We compare the performance of our model with the CVSS base score, a standard used in industry to prioritize vulnerability remediation.

A. Performance Evaluation

We evaluate our classifiers based on precision and recall as well as Receiver Operating Characteristics (ROC). They are computed as reported in Table IV. Precision is defined as the fraction of vulnerabilities that were exploited from all vulnerabilities predicted to be exploited by our model. It highlights the effect of mistakenly flagging non-exploited vulnerabilities. Recall is defined as the fraction of correctly predicted exploited vulnerabilities from the total number of exploited vulnerabilities. It highlights the effect of unflagging important vulnerabilities that were later used in attacks. The F1 measure is the harmonic mean of precision and recall. It summarizes the precision and recall in a common metric. The F1 measure can be varied based on the trade-off between precision and recall. This trade-off is dependent on the priority

TABLE III
NUMBER OF VULNERABILITIES EXPLOITED, THE NUMBER OF EXPLOITED VULNERABILITIES, THE FRACTION OF EXPLOITED VULNERABILITIES THAT APPEARED IN EACH SOURCE, AND THE FRACTION OF TOTAL VULNERABILITIES THAT APPEARED IN EACH SOURCE. RESULTS ARE REPORTED FOR VULNERABILITIES AND EXPLOITED VULNERABILITIES APPEARED IN EDB, ZDI, DW (DISTINCT CVES), CVES IN ZDI OR DW, AND RESULTS FOR INTERSECTION OF THE THREE SOURCES.

	EDB	**ZDI**	**DW**	**ZDI ∨ DW**	**EDB ∨ ZDI ∨ DW**
Number of vulnerabilities	799	824	378	1180	1791
Number of exploited vulnerabilities	74	95	52	140	164
Percentage of exploited vulnerabilities	21%	31%	17%	46%	54%
Percentage of total vulnerabilities	6.3%	6.5%	3.0%	9.3%	14.2%

of the applications. If keeping the number of incorrectly flagged vulnerabilities to a minimum is a priority, then high precision is desired. To keep the number of undetected vulnerabilities that are later exploited to a minimum, high recall is desired. We further report Receiver Operating Characteristics (ROC) curve as well as Area Under Curve (AUC) of the classifier. ROC graphically illustrates the performance of our classifier by plotting the true positive rate (TPR) against the false positive rate (FPR) at various thresholds of the decision boundary. In binary classification problems, the overall TPR is equivalent to recall for the positive class while FPR is the number of *not exploited* vulnerabilities that are incorrectly classified as *exploited* from all *not exploited* samples. ROC is a curve; thus, AUC is the area under ROC. The higher, the better (i.e., a classifier with AUC =1 is a perfect classifier).

TABLE IV
EVALUATION METRICS. TP - TRUE POSITIVES, FP - FALSE POSITIVES, FN - FALSE NEGATIVES, TN - TRUE NEGATIVE.

Metric	**Formula**
Precision	$\frac{TP}{TP+FP}$
TPR (recall in case of binary classification)	$\frac{TP}{TP+FN}$
F1	$2 * \frac{precision*recall}{precision+recall}$
FPR	$\frac{FP}{FP+TN}$

B. Results

Avoiding Temporal Intermixing. In [6], the authors point out that the temporal intermixing of exploit warnings could lead to future events being used to predict past ones. This could lead to performance results that are far higher than the performance achieved by a real-world predictor processes streaming vulnerability data. In this experiment, we sort the vulnerabilities according to their disclosed dates on NVD. We reserve the first 70% for training and the rest for testing[9].

For a baseline comparison we use the version 2.0 CVSS base score to classify whether a vulnerability will be exploited or not based on the severity score assigned to it. The CVSS score has been used as a baseline in previous studies [2], [4]. Figure 4 shows the precision-recall curve for the CVSS score. It is computed by varying the decision threshold that decides whether to predict a vulnerability as exploited or not. The best F1 measure that could be obtained was 0.15. We now perform the experiment using our proposed model.

Figure 5 shows the performance comparison between the proposed model using the random forest classifier and the CVSS score. The model outperforms the CVSS score with an F1 measure of 0.4 with precision 0.45 and recall 0.35.

The performance on the minority class (i.e., *exploited*) is promising when realizing that the class imbalance is very severe and the ground truth is not perfect. Additionally, our classifier shows very high TPR (90%) at low FPR (13%) and AUC of 94% as depicted in Figure 6.

Evaluation with Individual Data Sources. We study the gain in the performance achieved on the vulnerabilities mentioned on each data source when features from that source are used. To do so, we first quantify the performance achieved by a classifier trained/tested only on the features extracted from NVD. Then we compare that classifier with a classifier trained/tested on features from NVD as well as features from the data source we study. We find that time based split used in the previous experiment leaves very few vulnerabilities mentioned in these data sources in the test set (ZDI: 18, DW: 4, EDB: 2). Therefore, we increase the numbers by (1) performing a 10-fold cross validation (2) increasing the ground truth by considering the exploited vulnerabilities that did not have an exploit date (these were removed from earlier experiments since it was not clear whether these were exploited before or after the vulnerability was exploited). Using these two techniques, we have 84 vulnerabilities mentioned in ZDI that have been exploited, 57 in EDB, and 32 in DW. For the vulnerabilities mentioned in DW, we only consider DW features along with NVD. The classifier predicts 12 vulnerabilities as exploited with a precision of 0.67 and recall of 0.38; whereas when only considering the NVD features, the classifier predicts 12 vulnerabilities as exploited with a precision of 0.23 and recall of 0.38. Hence using the DW features, the precision improved significantly from 0.23 to 0.67. Table V shows the precision-recall with corresponding F1 measure. DW information was thus able to correctly identify the positive sample mentioned in DW with higher precision.

For ZDI we have 84 vulnerabilities mentioned in it. By just

[9]We examine other training/testing splits including 60%/40%, and 80%/20%. The former leaves fewer vulnerabilities from which the predictor can learn good representation, while the latter leaves very few *exploited* vulnerabilities to use for validation in the testing set.

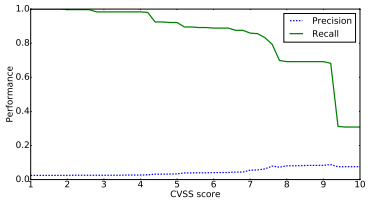

Fig. 4. Precision and recall for classification based on CVSS base score version 2.0 threshold .

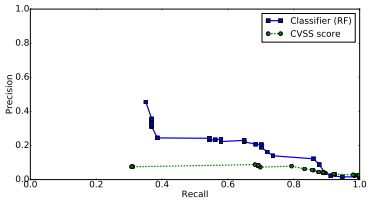

Fig. 5. Precision-Recall curve for classification based on CVSS score threshold (RF).

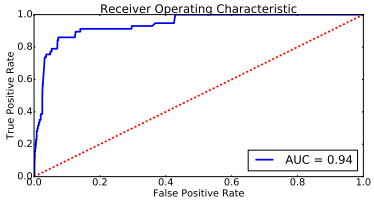

Fig. 6. ROC curve for classification based on Random Forest classifier.

TABLE V
PRECISION, RECALL, F1 MEASURE FOR VULNERABILITIES MENTIONED ON DW, ZDI, AND EDB.

Source	Case	Precision	Recall	F1 measure
DW	NVD	0.23	0.38	0.27
	NVD + DW	0.67	0.375	0.48
ZDI	NVD	0.16	0.54	0.25
	NVD + ZDI	0.49	0.24	0.32
EDB	NVD	0.15	0.56	0.24
	NVD + EDB	0.31	0.40	0.35

utilizing NVD features, the classifier achieves an F1 measure of 0.25 on these vulnerabilities (precision: 0.16, recall: 0.54). With the addition of the ZDI feature, the classifier achieves an F1 measure of 0.32 (precision: 0.49, recall: 0.24), a significant improvement in precision. Table V also shows the precision-recall with corresponding F1 measure for the vulnerabilities mentioned on ZDI.

Additionally, there are 57 vulnerabilities with PoCs archived on EDB. The classifier achieves an F1 measure of 0.24 (precision: 0.15, recall: 0.56) when only features from NVD are used, while the classifier achieves an F1 measure of 0.35 (precision: 0.31, recall: 0.40) when the EDB feature is added, a significant improvement in precision as shown in Table V.

V. DISCUSSION

Viability of the Model and Cost of Misclassification. The performance achieved by our model is very promising. Recall that the random forest classifier outputs a confidence score for every testing sample. A threshold can be set to identify the decision boundary. We shall note that all the results we report in this paper are achieved based on hard-cut thresholds such that all samples that are assigned a confidence score greater than a threshold thr are predicted as *exploited*. Blindly relying on a hard-cut threshold may not be practical in real-world exploits prediction; rather, thr should be varied in accordance to other variables within the organizations - different thresholds can be set for different vendors (i.e., thr_{ven1}, thr_{ven2}), or information systems (i.e., thr_{sys1}, thr_{sys2}). For instance, if an organization hosts an important website on an Apache server, and the availability of that site is a top priority for that organization, then any vulnerability in Apache server should receive higher attention. Other vulnerabilities, tens of which are disclosed every day, may exist in many other systems within the organization. Since it is very expensive to be responsive to all security advisories (e.g., some patches may be unavailable or some systems may need to be taken offline to apply patches), exploitation assessment measures help in quantifying the risk and prioritizing remediations.

Model Success and Failure Cases. By analyzing the false negatives and false positives, we gain an understanding as to why and where our model performs well along with why and where it suffers. We first look into the false negatives. The 10 exploited vulnerabilities (about 18% of the *exploited* samples in the testing dataset) that received the lowest confidence scores seem to have common features. For example 9 of these 10 exist in products from Adobe, namely Flash Player (5 vulnerabilities) and Acrobat Reader (4 vulnerabilities). The Flash Player's vulnerabilities appear to have very similar description from NVD, and the same observation holds for the Acrobat Reader's vulnerabilities. We also observe that they were assigned CVE-IDs at the same date (April 27th, 2016), and 7 out of these 9 were published at the same date as well (July 12th, 2016), and assigned a CVSS base score = 10.0 (except for one, assigned 7.0). The other vulnerability exist in Windows Azure Active Directory (CVSS score = 4.3). Out of these 10 vulnerabilities, only one had a verified PoC archived on EDB before it was detected in the wild, and another one had a ZDI mention, while none was mentioned in DW. We attribute misclassifying these vulnerabilities to the limited representation of these samples in the training dataset. Further, this observation signifies the importance of avoiding experiments on time-intermixed data because if some of the said vulnerabilities were used for training, it is highly likely the others will be predicted as *exploited* as they possess very similar features. We also look into the false positive samples that receive high confidence scores - samples where our model predicted as *exploited* while they are not. For our random forest classifier, with an F1 of 0.4, all the false positives (about 40 out of about 3600 vulnerabilities) exist in products from Microsoft although we do not use the vendor as feature. Our model seems to be able to infer the vendor from other textual features. We posit that this level of over-fitting is unavoidable and marginal, and we attribute this largely to the limitations on our ground truth - which was shown to have bias

towards reporting vulnerabilities that exist in products from limited set of software vendors dominated by Microsoft [4]. However, the model is highly generalizable. We find examples of vulnerabilities from other vendors with confidence scores close to thr we set; but we cannot assume that they were *exploited* since they were not reported by Symantec.

VI. Related Work

Our approach closely resembles previous work on using publicly disclosed vulnerability information as features to train a machine learning model and predict if a vulnerability will be exploited or not. Bozorgi et al. [8] proposed a model that engineered features from two online sources, namely, the Open Source Vulnerability database (OSVDB)[10] and vulnerability data from NVD to predict whether PoCs will be developed for a particular vulnerability. In their data, 73% of the vulnerabilities are reported as exploited (have PoCs). Similar technique is employed by [1] using NVD as a data source. They use Exploit Database (EDB) as their ground truth, with 27% vulnerabilities marked as exploited (have PoCs). They achieve high accuracy on a balanced dataset. Both studies aim at addressing a problem different from ours. Our analysis aims to predict vulnerabilities that will be used in real-world attacks and not just have PoCs available.

Building on the work regarding the use of publicly disclosed vulnerabilities, Sabottle et al. [4] look to predict the exploitability based on vulnerabilities disclosed from Twitter data. They collect tweets, that have CVE-ID's mentioned in them. They use these tweets to compute features and use linear SVM classifier for prediction. They use Symantec threat signatures to label positive samples. As compared to previous predictive studies, even though [4] maintains the class ratio of 1.3% vulnerabilities exploited, they use a resampled and balanced dataset to report their results. Also, the temporal aspect (training data should precede testing) of the tweets is not maintained while performing the experiment. This temporal intermixing causes future events to influence the prediction of past events. These practices have been reported to influence the prediction accuracy [6]. Therefore, in our approach we maintain the class imbalance and the temporal aspect while reporting our results. We also do not use the existence of PoCs as positive labels but use them as a feature indicating higher chance that a vulnerability will be exploited.

VII. Conclusion

We conduct a study of aggregating early signs from online vulnerability mentions for predicting whether a particular disclosed vulnerability will be exploited. We demonstrate the utility of our prediction model through a series of experiments on real-world vulnerability and exploit data. Our results show that, while maintaining high True Positive Rate, we achieve significantly low False Positive Rate in predicting exploits. In the future, we look to combine our model with other vulnerability data sources including social media sites and online blogs.

[10]https://blog.osvdb.org/

Acknowledgment

Some of the authors were supported by the Office of Naval Research (ONR) contract N00014-15-1-2742, the Office of Naval Research (ONR) Neptune program and the ASU Global Security Initiative (GSI). Paulo Shakarian and Jana Shakarian are supported by the Office of the Director of National Intelligence (ODNI) and the Intelligence Advanced Research Projects Activity (IARPA) via the Air Force Research Laboratory (AFRL) contract number FA8750-16-C-0112. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ODNI, IARPA, AFRL, or the U.S. Government.

References

[1] M. Edkrantz and A. Said, "Predicting cyber vulnerability exploits with machine learning." in *SCAI*, 2015, pp. 48–57.

[2] L. Allodi and F. Massacci, "Comparing vulnerability severity and exploits using case-control studies," *ACM Transactions on Information and System Security (TISSEC)*, vol. 17, no. 1, p. 1, 2014.

[3] K. Nayak, D. Marino, P. Efstathopoulos, and T. Dumitraş, "Some vulnerabilities are different than others," in *International Workshop on Recent Advances in Intrusion Detection*. Springer, 2014, pp. 426–446.

[4] C. Sabottke, O. Suciu, and T. Dumitras, "Vulnerability disclosure in the age of social media: Exploiting twitter for predicting real-world exploits." in *USENIX Security*, vol. 15, 2015.

[5] V. R. Team, "2015 data breach investigations report," 2015.

[6] B. L. Bullough, A. K. Yanchenko, C. L. Smith, and J. R. Zipkin, "Predicting exploitation of disclosed software vulnerabilities using open-source data." in *Proceedings of the 2017 ACM International Workshop on Security and Privacy Analytics*. ACM, 2017.

[7] L. Allodi, W. Shim, and F. Massacci, "Quantitative assessment of risk reduction with cybercrime black market monitoring," in *Security and Privacy Workshops (SPW), 2013 IEEE*. IEEE, 2013, pp. 165–172.

[8] M. Bozorgi, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond heuristics: learning to classify vulnerabilities and predict exploits," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2010, pp. 105–114.

[9] M. Motoyama, D. McCoy, K. Levchenko, S. Savage, and G. M. Voelker, "An analysis of underground forums," in *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, ser. IMC '11. New York, NY, USA: ACM, 2011, pp. 71–80. [Online]. Available: http://doi.acm.org/10.1145/2068816.2068824

[10] T. J. Holt and E. Lampke, "Exploring stolen data markets online: products and market forces," *Criminal Justice Studies*, vol. 23, no. 1, pp. 33–50, 2010. [Online]. Available: http://dx.doi.org/10.1080/14786011003634415

[11] E. Marin, A. Diab, and P. Shakarian, "Product offerings in malicious hacker markets," in *Intelligence and Security Informatics (ISI), 2016 IEEE Conference on*. IEEE, 2016, pp. 187–189.

[12] J. Shakarian, A. T. Gunn, and P. Shakarian, "Exploring malicious hacker forums," in *Cyber Deception*. Springer, 2016, pp. 261–284.

[13] E. Nunes, A. Diab, A. Gunn, E. Marin, V. Mishra, V. Paliath, J. Robertson, J. Shakarian, A. Thart, and P. Shakarian, "Darknet and deepnet mining for proactive cybersecurity threat intelligence," in *Intelligence and Security Informatics (ISI), 2016 IEEE Conference on*. IEEE, 2016, pp. 7–12.

[14] J. Robertson, A. Diab, E. Marin, E. Nunes, V. Paliath, J. Shakarian, and P. Shakarian, *Darkweb Cyber Threat Intelligence Mining*. Cambridge University Press, 2017.

[15] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.