

Predicting Cyber Threats through Hacker Social Networks in Darkweb and Deepweb Forums

Mohammed Almukaynizi, Alexander Grimm, Eric Nunes,
Jana Shakarian, Paulo Shakarian
Arizona State University
Tempe, AZ 85281, USA
Email: {malmukay, argrimm, enunes1, jshak, shak} @asu.edu

ABSTRACT

We present an approach that combines social network analysis with machine learning techniques to predict future cyber threats through darkweb/deepweb discussions with hacking-related content. Our approach harnesses features derived from hacker social networks and from online sources of cybersecurity advisories. We address the problem of predicting the exploitability of software vulnerabilities to show that features computed from hacker social networks are important indicators of future cybersecurity incidents. We conduct a suite of experiments on real-world hacker and exploit data and demonstrate that social network data improves recall by about 19%, F1 score by about 6% while maintaining precision. We believe this is because social network structures related to certain exploit authors is indicative of their ability to write exploits that are subsequently employed in an attack.

1 INTRODUCTION

The existing methods for assessing the likelihood of software vulnerability exploitation are found to have many limitations, preventing them from being useful tools for prioritizing vulnerability remediation. They are known to assign high severity scores to many vulnerabilities that will never be exploited [1]. Recent studies show that less than 3% of the vulnerabilities reported are ever exploited in the wild [1, 15]. Therefore, cybersecurity researchers have used alternative methods leveraging machine learning techniques on features computed from social media feeds with explicit vulnerability mentions [5, 15]. In both studies, social network data has not been analyzed nor has it been evaluated as a predictor. Additionally, while other studies suggest that darkweb/deepweb (D2web) sites are among the best sources for gathering cyber threat intelligence [1, 9, 17], the value of such intelligence in predicting the likelihood of vulnerability exploitation is not quantified.

In this paper, we introduce an approach that leverages machine learning techniques on features derived from the social network

of users participating in D2web forums, as well as features derived from the National Vulnerability Database (NVD)¹. We demonstrate the viability of user connectivity metadata in predicting cyber threats. Our machine learning model achieves an F1 measure of 0.67 and generalizes well with the biased ground truth. Additionally, we propose an approach for generating a social network graph from the discussions on D2web forums, and we provide analysis on the social network structure. We specifically show that the users who discussed vulnerabilities are about 4 times more active in posting than other users.

2 BACKGROUND

2.1 Vulnerability Lifecycle

In this subsection, we provide a general description of the typical vulnerability lifecycle. We note that vulnerabilities may have variations on their lifecycles depending on different factors.

Vulnerability Reporting. Researchers are encouraged to report discovered security flaws to any CVE Numbering Authorities (CNAs)². Once a flaw is reported, the CNA reserves a new Common Vulnerabilities and Exposures (CVE) number if two constraints are satisfied: first, the flaw can be exploited in a way that violates the security policies; second, no CVE number has been previously assigned to this flaw. Once it is established that the flaw poses cyber threats, the software vendor is notified and allowed a period of time to release patches before any information about the vulnerability is publicly disclosed.

Vulnerability Disclosure. Once a legitimate vulnerability is validated and the time period provided to the vendor is over, the National Institute of Standards and Technology (NIST) releases information about the vulnerability to the cybersecurity community. This disclosure is published through the National Vulnerability Database (NVD) - which is known to be the comprehensive reference vulnerability database with over 92,000 disclosed vulnerabilities as of July 2017.

Vulnerability Exploitation. Ideally, after a vulnerability is disclosed by NIST, organizations immediately begin patching their systems to avoid potential security risks. However, with the ever-increasing number of disclosed vulnerabilities, many organizations

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CSS '17, CSSSA's Annual Conference on Computational Social Science, October 19–22, 2017, Santa Fe, NM, USA

© 2017 ACM. 978-1-4503-5269-7/17/10...\$15.00

DOI: 10.1145/3145574.3145590

¹<https://nvd.nist.gov>.

²For a complete list of CNAs, see <https://cve.mitre.org/cve/cna.html>.

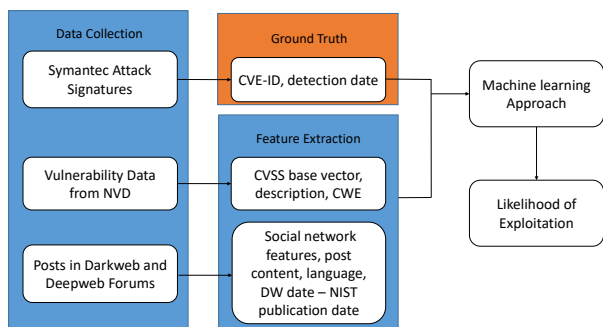


Figure 1: An overview of the predictive model.

fall behind on vulnerability mitigation and become exposed to a wide range of newly deployed exploits that exist in the wild. Recent studies show that only a small fraction of vulnerabilities are found to be exploited in the wild (1-3% [1, 7, 10, 15]). Therefore, organizations look for methods that help to better assess the exploitability likelihood, when compared to the existing severity scoring systems.

3 APPROACH

We view the problem of predicting the vulnerability likelihood of exploitation as a binary classification problem, where the positive class is *exploited*, and the negative class is *not exploited*. Figure 1 gives an overview of our proposed approach.

3.1 Data Collection

Our machine learning models use features derived from two sources: vulnerability data feeds collected from NVD, and a D2web database of posts with cybersecurity-related content collected and filtered from 151 D2web forums [12]. The class labels are determined based on a ground truth set of attack signatures of exploits detected in the wild and reported by Symantec³. For the purpose of this paper, we focus our study on data collected from all the mentioned sources between January 2010- March 2017. Figure 2 shows the number of vulnerabilities published on NVD per year, the number of vulnerabilities mentioned on D2web, and the number of vulnerabilities reported by Symantec. Next, we provide details on the data collection process for the three sources we use.

NVD. Every vulnerability in NVD is assigned a unique CVE number. To collect vulnerability data, we use the JSON data feeds provided by NVD and extract information about the vulnerabilities we study (extracted information is discussed in Section 3.2). Further, a web scraper is developed to retrieve from the NVD’s vulnerability web-pages the data elements that are missing in the JSON files (e.g., the disclosure date).

³<https://www.symantec.com>

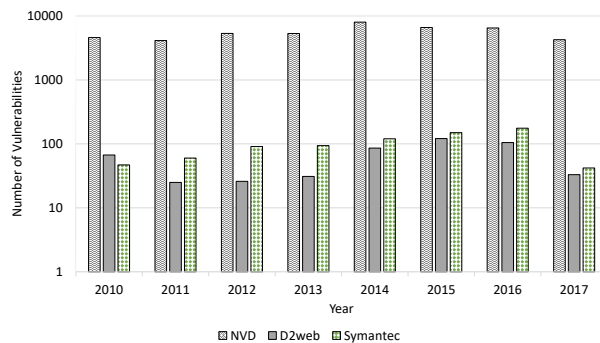


Figure 2: Vulnerabilities reported per year from NVD (per the disclosure year), D2web (per the disclosure year), and Symantec (per the exploitation year).

D2web Forums. We use a database of posts collected from 151 darknet and deepnet forums. The data collection system is described in [12]. In summary, the system extracts the hacking-related content from D2web sites using a learning algorithm with high accuracy. The database we use contains over 2,290,000 posts under 223,074 distinct forum topics and from 151 distinct forums. From these posts, only 3,082 have explicitly mentioned 624 distinct vulnerabilities by referencing their CVE-ID. Of those, 502 are within the time period we focus on, and they are found on 46 different forums. We only use these forums to build the social network of users, as described in Section 4.

Symantec Attack Signatures. We label vulnerabilities as *exploited* in the wild if the CVE-ID is mentioned in the description of the attack signatures reported by Symantec’s anti-virus⁴ or Intrusion Detection Systems’ attack signatures⁵. The fraction of exploited vulnerabilities is found to be very small as shown in Figure 2, and it varies from year to another. The maximum value for the fraction of exploited vulnerabilities is 2.7% (for vulnerabilities published in 2016), and minimum value is less than 1% (for vulnerabilities published in 2017) - previous work has reported fractions comparable to our findings [1, 15].

3.2 Features Description

Table 1 summarizes the features we extract from both data sources discussed in Section 3.1. Here we provide discussions on each of these sets of features.

CVSS Base Score. CVSS is a vulnerability severity scoring framework designed to measure the exploitability and the impact of the vulnerabilities. We use versions v2.0 CVSS base score. There are two components for this set of features: (1) the base score (numeric): a given severity score ranges from 0 to 10, and (2) the CVSS vector: a vector of the metrics that determine the base score (categorical). The measures in the vector are *Access Vector*, *Access Complexity*,

⁴A complete list is found here https://www.symantec.com/security_response/landing/azlisting.jsp. The detection date is labeled with “Discovered”.

⁵https://www.symantec.com/security_response/attacksignatures.

Table 1: Summary of features

Source	Feature Set	Type
NVD	CVSS base score	Numeric and Categorical
	Description	TF-IDF uni-grams
	CWE	Categorical
D2web	Social Network Features	Numeric
	Post content	TF-IDF uni-grams
Combined	D2web date - NVD date	Numeric

Authentication, Confidentiality Impact, Integrity Impact, and Availability Impact. Each one of these measures can take one value from a predetermined set of possible values. For example, *Access Vector* indicates how the vulnerability is exploited. It can take one of these three possible values: *Local (L)*, *Adjacent Network (A)* and *Network (N)*⁶.

NVD Description. NIST provides a textual description of the vulnerability when it is released. The description summarizes the system/software in which the flaw exists and gives information on how it can be exploited. Each vulnerability description undergoes a preprocessing pipeline including stemming (reducing the words to their root forms) and stop words removal (e.g., *and*, *or*, *then*). The preprocessed documents are then turned into numerical feature vectors using Term Frequency-Inverse Document Frequency (TF-IDF) computed for the 250 most frequent words in the collection of documents. Informally, TF-IDF assigns higher values to the words that appeared in a document with higher frequency while rarely did these words appear in other documents in the corpus.

CWE. It is a community-effort project comprising enumerating common software security weaknesses (categorical). These are categories of flaws that can be unintentionally made during the software development and can exist in the software architecture, design, or code.

D2web Social Network Features. This set of features contains measures computed from the social connections of users posting hacking-related content. The basic social network features (e.g., in-degree, out-degree) indicates how active a user is in the social graph. More advanced features measure the centrality of users in the social graph. Highly central users are more important; thus the vulnerability mentions should take more consideration. We compute the features for the set of users who explicitly mentioned one or more CVE-IDs in their posts.

Post Content. We found evidence for many vulnerability mentions with content ranging from exploit offers to content irrelevant of the mentioned vulnerability. This set of features is extracted the same way as NVD description features except that for non-English posts, we automatically translate the content to English using Google

Translate API⁷, then the TF-IDF are computed over the translated words.

3.3 Classifier Training and Prediction

We use a supervised machine learning approach to train classifiers on the presented features. The output of the classifiers is a confidence score. A threshold can be set on the confidence score to determine the best decision boundary. The experimental settings and results are described in Section 6.

4 HACKER SOCIAL NETWORK

In Section 3.2, we describe the features we use, while here, we elaborate on the social network creation approach. We then provide our observations on the created graphs in Section 5. In this work, we adopt the same assumption made in much of the previous work on D2web data, where they consider the same usernames (case insensitive) across different D2web sites to belong to the same person(s) [14]. This allows for generating one network comprising a large number of D2web sites as opposed to a social network for each site [13].

4.1 Users' Social Graph

Formally, the users' social graph $G = (V, E)$ is a weighted, directed graph with no self-loops (i.e., every edge has a weight; every edge points away from one node to another node; there exists at most one edge between any pair of nodes). V is the set of vertices (D2web users) and E is the set of edges.

The graph is creating using posts between January 2010 to June 2016 (747,351 posts under 109,413 forum topics), and for every topic t_x , posts under t_x are grouped together in a list l_x ordered by the date and time of posts. Then, an edge is created (with weight = 1) from user v_i to v_j and labeled with the date of v_i 's posting date only if: (1) $v_i \neq v_j$, (2) both v_i and v_j have posts in l_x , and v_i has posted after v_j , (3) the number of posts between v_i 's post and v_j 's post in l_x is less than thr (it is set to be 10 in all experiments in this paper), and (4) there is no existing edge originating from v_i to v_j and labeled with the same date. Once the edges are created, they are added to a multi-directed graph with parallel edges of weights = 1. The multi-graph is then transformed to a directed graph G by summing the weights of the parallel edges pointing at the same direction.

Degree distributions, for both incoming and outgoing edges, of G are found to resemble the power-law distribution as depicted in Figure 3. This means that there exist very few users with a very large number of connections, and many users with few connections - this observation is known to be common for social media sites [18]⁸.

⁷<https://cloud.google.com/translate>.

⁸In the D2web database, there are few users, each has exactly one post. Some of these posts are found in topics with a number of postings greater than thr ; hence the little up-tick in Figure 3.

⁶See <https://www.first.org/cvss/v2/guide> for complete documentation.

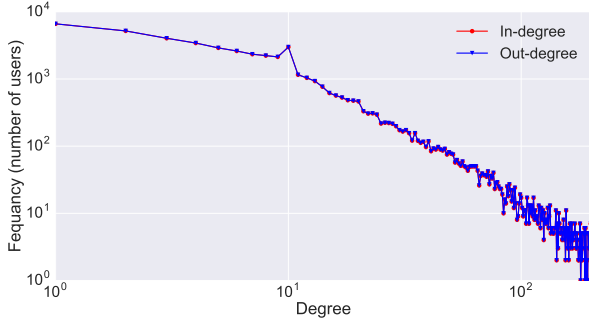


Figure 3: In-degree and out-degree distribution of users in G .

4.2 Social Network Measures

After creating the social network, we compute measures derived from the network structure. In this paper, we consider three categories of social network measures:

Network Structure Measures: the measures under this category are: (1) *In-degree*: the number of edges pointing to the user, (2) *Out-degree*: the number of edges originated from the user, (3) *Sum of In-degree weights*: the sum of the weights for all edges pointing to the user, (4) *Sum of out-degree weights*: the sum of the weights for all edges pointing away from the user. These measures describe the type of activities in which the user engages. For example, higher in-degree than out-degree may indicate the user tendency towards creating new topics or posting under topics short time after they are created.

Centrality Measures: three measures are computed: (1) *In-degree centrality*: it measures the popularity of a user v_i by normalizing v_i 's in-degree by the maximum possible in-degree, (2) *Out-degree centrality*: measures how actively a user v_i replies to others by normalizing v_i 's out-degree measure by the maximum possible out-degree, (3) *Betweenness centrality*: for a user v_i , Betweenness centrality measures the importance of v_i by computing the fraction of shortest paths between all pairs of users that pass through v_i .

Importance Measures: the number of connections user v_i has with other users, by itself, may not be indicative of importance; rather, v_i is important if his/her posts make other important users reply. Hence, influence metrics incorporate the centrality of users with outgoing edges to v_i into v_i 's centrality (i.e., if an important user v_j replies to v_i , then the importance of v_i increases). Two measures are computed under this category: (1) *Eigenvector centrality*: measures the importance of v_i by assigning a centrality proportional to the sum of in-neighbors' centralities. Eigenvector centrality of v_i is the i^{th} value of the eigenvector C_e corresponding to the largest eigenvalue of the network adjacency matrix A^t , and (2) *Pagerank centrality*: measures the centrality of v_i by incorporating fractions of the centralities of in-neighbors, such that each of v_i 's in-neighbors passes the value of his/her centrality divided by the number of outgoing edges⁹.

⁹In all the experiments, the damping factor is set to 0.85.

5 SOCIAL NETWORK ANALYSIS

In this section, we report our observations on the computed measures. Table 2 shows statistics for the D2web social graph G created according to our description, as well as statistics for (1) the subset of users who have discussed vulnerabilities in D2web (s.t., $vulUsers \subset V$), (2) the subgraph of G induced by the $vulUsers$, $G_{vulns}(vulUsers, vulEdges)$, (s.t., $vulEdges \subset E$), and (3), a subgraph induced by $vulUsers$ as well as all their in-neighbors and out-neighbors $G_{vulNei}(vulNeis, vulNeiEdges)$, (s.t., $vulNeis \subset V$ and $vulNeiEdges \subset E$). We create these three graphs to understand the differences in the user connectivity for the subset of users mentioned vulnerabilities (s.t., $vulUsers$, less than 1% of the total population) as opposed to the total user population.

These individuals, or $vulUsers$, are spread across many D2web forums (46 forums out of 151). Further, they are generally more active than other users in the same forums. As depicted in Table 2, the average in-degree (64.90) and out-degree (51.45) for the subset of $vulUsers$ are orders of magnitude higher than the same measures for all users in the graph (13.74)– about 5 times higher in-degree and 4 times higher out-degree. This shows that the average hacker in $vulUsers$ is exposed to a larger population of hackers than a normal hacker. Expectedly, we observe that a hacker from $vulUsers$ is more likely to engage in discussions with another hacker from the same group than he/she does with others. For example, the average in-degree and out-degree (4.08) for the subgraph G_{vulns} indicates that, on average, a user has connections with about 4 other $vulUsers$ (about 1% of the population of $vulUsers$); whereas he/she would reply to about 51 users and make about 64 other users reply to his/her post, less than 0.1% of the total population. For these reasons, $vulUsers$ generally, exhibit significantly higher centrality and importance measures as compared to normal users in G . However, we observe that the distribution of network measures vary largely within this group of users. We also observe that about 30% of the $vulUsers$ have no communication history with other users within the same group. Figure 4 shows a visualization for the subgraph G_{vulns} , which confirms the two observations. Finally, about 25% of $vulUsers$ joined the D2web community less than three days before their first vulnerability mention.

6 EXPERIMENTAL SETUP

We perform experiments on the set of vulnerabilities mentioned on D2web forums in the time period from January 2010 to March 2017. We exclude the vulnerabilities that were mentioned by users with no communication history. We also exclude the vulnerabilities that had been detected in the wild by Symantec before they were mentioned in any of the D2web posts, since those vulnerabilities could be retrieved by querying the database without prediction. Our resultant dataset contains 157 distinct vulnerabilities, 24 of which have the class label *exploited*.

Different machine learning classifiers are compared¹⁰, but we only report the performance achieved by the best performing classifier, which is Random Forest (RF) [4]. RF is an ensemble of decision

¹⁰Classifiers used include: Support Vector Machine (SVM), Logistic Regression, and Naive Bayes.

Table 2: Statistics for the graph G with all users with at least one edge (an in-edge or an out-edge), the subset of users that have discussed vulnerabilities $vulUsers$, subgraph $(G_{vulIns}(vulUsers, vulEdges))$, and subgraph $(G_{vulNei}(vulNeis, vulNeiEdges))$.

Property	D2web users	Users mentioning vulnerabilities		
		as a subset of nodes	as a subgraph	as a subgraph with all 1-hop neighbors
Nodes	53, 178	365	365	10469
Edges	730, 740	undefined	1, 492	202, 070
The average of:				
In-degree	13.74	64.90	4.08	19.30
Out-degree	13.74	51.45	4.08	19.30
Sum of in-degree weights	35.80	250	48.07	75.73
Sum of out-degree weights	35.80	215	48.07	75.73
In-degree centrality	$2.59e^{-4}$	$1.22e^{-3}$	$1.12e^{-2}$	$1.84e^{-3}$
Out-degree centrality	$2.59e^{-4}$	$9.68e^{-4}$	$1.12e^{-2}$	$1.84e^{-3}$
Betweenness centrality	$7.2e^{-5}$	$1.29e^{-3}$	$1.89e^{-4}$	$2.35e^{-4}$
Eigenvector Centrality	$2.18e^{-4}$	$4.68e^{-4}$	$5.95e^{-3}$	$8.21e^{-4}$
Pagerank	$1.90e^{-5}$	$1.27e^{-4}$	$2.74e^{-3}$	$9.6e^{-5}$

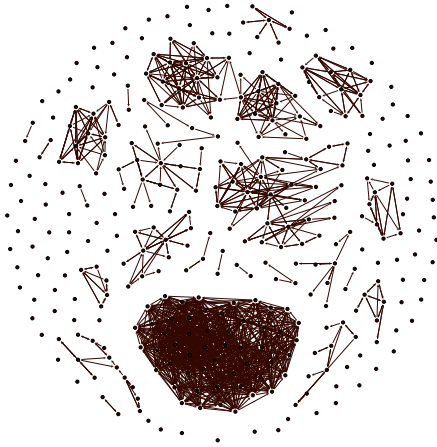


Figure 4: The subgraph of G induced by the set of users who have mentioned vulnerabilities in their postings.

trees generated from the training data. Each tree produces a prediction independent of other trees, with the algorithm taking the majority vote at the end. In each of the experiments, we report an averaged performance of five runs.

6.1 Performance Evaluation

We evaluate the performance of the predictors using precision, recall, and the harmonic mean of precision and recall, which is F1 measure. Precision is the fraction of vulnerabilities that were actually exploited from all vulnerabilities predicted as being exploited, and recall is the fraction of correctly predicted exploited vulnerabilities from the total number of exploited vulnerabilities. Table 3 defines how each metric is computed.

Table 3: Evaluation metrics. TP - true positives, FP - false positives, FN - false negatives, TN - true negatives.

Metric	Formula
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
F1	$2 * \frac{precision * recall}{precision + recall}$

We also compute Receiver Operating Characteristics (ROC) for each run and report Area Under Curve (AUC) of the classifier. ROC graphically illustrates the classification performance by plotting the true positive rate (TPR) against the false positive rate (FPR) at different points of the decision boundary.

6.2 Results

Experiments under Real-World Conditions. In this set of experiments, we sort the vulnerabilities by their D2web date, then we train our classifiers on the vulnerabilities mentioned before June 2016 (125 vulnerabilities), and test on the vulnerabilities from June 2016 to March 2017 (32 vulnerabilities, only 3 are *exploited*). The classification performance achieved by our RF model has an average precision of 0.57, recall of 0.93, and F1 of 0.67. The same classifier is able to achieve on average AUC of 0.95¹¹. The lower score of precision is attributed to the fact that Symantec’s data is biased towards reporting exploits targeting vulnerabilities that exist in software products from certain software vendors such as Microsoft, Adobe [15]. Since our model is found to predict vulnerabilities as being exploited from other vendors as well, we believe that some

¹¹The results of 5 runs show relatively high variance due to the small number of samples on which the models are tested.

false positives were actually exploited in the wild but never detected by Symantec.

Ablation Test and Cross-Validation. Since the number of vulnerabilities in our testing dataset in the previous experiment is relatively small, we further apply stratified 5-fold cross-validation on the whole dataset. In this experiment, the samples are intermixed; hence these conditions do not reflect the conditions of real-world streaming prediction (i.e., predicting the likelihood of exploitation at the time of the vulnerability mention). The average F1 achieved is 0.72, with a precision of 0.61, a recall of 0.89, and an AUC of 0.88¹².

To measure the impact of individual feature sets on the overall classification performance, we apply two tests: (1) an ablation test (depicted in Figure 5) where the change in precision, recall, F1, and AUC is recorded when each set of features is removed from the prediction model, and (2) a test on individual feature sets (depicted in Figure 6) where the classification performance is reported for models trained on only one set of features at a time. In the ablation test, when the set of social network features is removed from the model, some decrease in performance was recorded as depicted in Figure 5. In the individual feature tests, the social network measures resulted in an improvement in performance that is significantly higher than the improvement recorded from the inclusion of other feature sets. We note that the simple classifier - which labels all vulnerabilities as being *exploited*, results in a precision of 0.16, a recall of 1, at an F1 of 0.27 and an AUC of 0.5.

7 DISCUSSION

In the ablation test, the largest drop in F1 occurred when CVSS vector set of features were removed, followed by the removal of social network measures with comparable drop in F1. It is important to note that CVSS vector is designed to assess the vulnerability exploitability and impact, and the assignment of values to the different components of the vector is manually done by domain experts¹³. Hence, being able to achieve performance comparable to the expensive CVSS vector when social network features are used is very promising. However, we do note that even the experiments here where we only use the CVSS vector differ substantially from previous work. As we have selected a-priori on vulnerabilities that have appeared on D2web, we observe a superior performance of CVSS score in this paper when compared to previous work.

Additionally, when the individual feature sets were examined, the best performing features were the TF-IDFs computed from both the content of D2web postings and the vulnerability description retrieved from NVD. The second best performing features were the social network measures, scoring F1 that is significantly higher than F1 scores achieved by the other individual feature sets - excluding the textual features. The D2web textual content provides rich information about the context in which the vulnerability is discussed. Furthermore, the software vendor (e.g., Microsoft, Adobe) can be easily derived from the NVD description; leading the model

¹²Standard deviation for the evaluation metrics across the five runs is between 0.02 (F1 and AUC) and 0.06 (recall).

¹³See <https://www.first.org/cvss/v2/guide>.

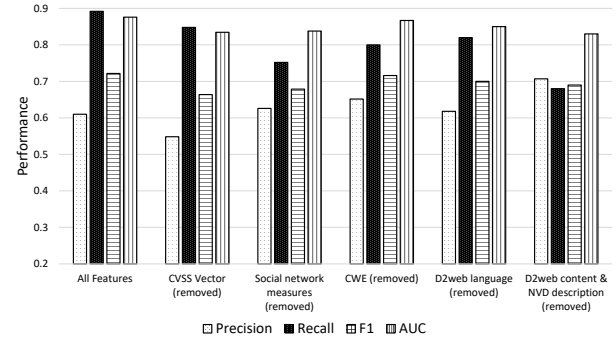


Figure 5: Classification performance achieved by applying ablation test with 5-fold cross-validation.

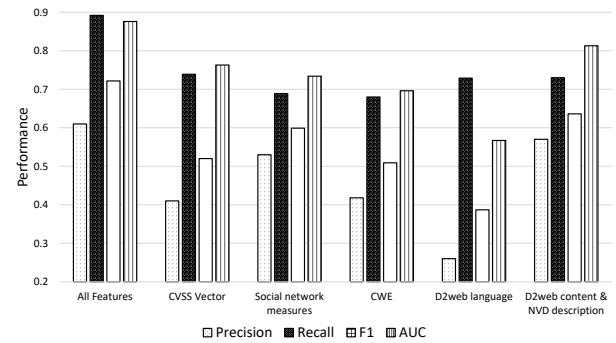


Figure 6: Classification performance achieved by individual feature sets.

to potentially over-fit the biased ground truth. In all experiments, social network measures demonstrated their viability as predictors of potential cyber threats.

8 RELATED WORK

8.1 Vulnerability Exploitation Prediction

A recent study found that CVSS base score metrics are poor indicators of exploitation [1]. Many vulnerabilities are assigned high scores, resulting in very high false positive rates. Sabottle et al. [15] proposed an SVM classifier that leverages data feeds from Twitter with explicit mentions to legitimate CVE identifiers to predict whether a vulnerability will have proof-of-concepts available in one experimental setup, and predict whether a vulnerability will be detected in the wild in other experimental setup. Results in their work are reported on time-intermixed samples (i.e., samples in testing set may have appeared before samples in training set), and including samples where the exploitation date is before any of the tweets are posted. Both practices have been discussed in [5]

to measure their impact on real-world proactive exploitation prediction settings. The work in [5] replicates the experiments done in [15] (with data feeds spanning different time period) to predict the existence of proof-of-concepts from EDB. They found that the experimental methodology highly influence the results. In both papers, the dynamics of user connectivity are not studied.

8.2 Social Network Analysis

An extensive amount of work has focused on usage of measures computed from a social network of actors to identify malicious actions [3, 6, 11]. For example, Cao et al. [6] proposed a method called *SybilRank* that relies on social network measures to identify fake accounts (Sybils). Across multi-disciplines, hacker communities in underground hacking forums have been widely studied to understand the dissemination of information among hackers, the motives for hacking, and the reputation and skill level of hackers to detect threats [2, 8, 16]. However, the dynamics of connections within hacker communities have not been quantified as predictors for vulnerability exploitation. We find the measures computed from social connections to be promising predictors of future cyber attacks.

9 CONCLUSION

Our work contributes toward the understanding of user connectivities in D2web forums, and extracting measures that serve as predictors of cyber attacks. We address the problem of predicting the likelihood of vulnerability exploitation to demonstrate the value of these measures. Our experimental results also suggest that these measures produce models that generalize well when the ground truth is biased.

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] Luca Allodi and Fabio Massacci. 2014. Comparing vulnerability severity and exploits using case-control studies. *ACM Transactions on Information and System Security (TISSEC)* 17, 1 (2014), 1.
- [2] Victor Benjamin, Weifeng Li, Thomas Holt, and Hsinchun Chen. 2015. Exploring threats and vulnerabilities in hacker web: Forums, IRC and carding shops. In *Intelligence and Security Informatics (ISI), 2015 IEEE International Conference on*. IEEE, 85–90.
- [3] Alex Beutel, Leman Akoglu, and Christos Faloutsos. 2015. Fraud detection through graph-based user behavior modeling. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1696–1697.
- [4] Leo Breiman. 2001. Random Forests. *Machine Learning* 45, 1 (2001), 5–32.
- [5] Benjamin L. Bullough, Anna K. Yanchenko, Christopher L. Smith, and Joseph R. Zipkin. 2017. Predicting exploitation of disclosed software vulnerabilities using open-source data.. In *Proceedings of the 2017 ACM International Workshop on Security and Privacy Analytics*. ACM.
- [6] Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro. 2012. Aiding the detection of fake accounts in large scale social online services. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*. USENIX Association, 15–15.
- [7] Michel Edkrantz and Alan Said. 2015. Predicting Cyber Vulnerability Exploits with Machine Learning.. In *SCAI*. 48–57.
- [8] Thomas J Holt, Deborah Strumsky, Olga Smirnova, and Max Kilger. 2012. Examining the social networks of malware writers and hackers. *International Journal of Cyber Criminology* 6, 1 (2012), 891.
- [9] Marti Motoyama, Damon McCoy, Kirill Levchenko, Stefan Savage, and Geoffrey M. Voelker. 2011. An Analysis of Underground Forums. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference (IMC '11)*. ACM, New York, NY, USA, 71–80. DOI: <https://doi.org/10.1145/2068816.2068824>
- [10] Kartik Nayak, Daniel Marino, Petros Efstathopoulos, and Tudor Dumitras. 2014. Some vulnerabilities are different than others. In *International Workshop on Recent Advances in Intrusion Detection*. Springer, 426–446.
- [11] Mariam Noun and Jason RC Nurse. 2015. Identifying Key-Players in Online Activist Groups on the Facebook Social Network. In *Data Mining Workshop (ICDMW), 2015 IEEE International Conference on*. IEEE, 969–978.
- [12] Eric Nunes, Ahmad Diab, Andrew Gunn, Ericsson Marin, Vineet Mishra, Vivin Paliath, John Robertson, Jana Shakarian, Amanda Thart, and Paulo Shakarian. 2016. Darknet and deepnet mining for proactive cybersecurity threat intelligence. In *Intelligence and Security Informatics (ISI), 2016 IEEE Conference on*. IEEE, 7–12.
- [13] Elizabeth Phillips, Jason RC Nurse, Michael Goldsmith, and Sadie Creese. 2015. Extracting social structure from darkweb forums. (2015).
- [14] John Robertson, Ahmad Diab, Ericsson Marin, Eric Nunes, Vivin Paliath, Jana Shakarian, and Paulo Shakarian. 2017. *Darkweb Cyber Threat Intelligence Mining*. Cambridge University Press.
- [15] Carl Sabottke, Octavian Suciu, and Tudor Dumitras. 2015. Vulnerability Disclosure in the Age of Social Media: Exploiting Twitter for Predicting Real-World Exploits.. In *USENIX Security*, Vol. 15.
- [16] Sagar Samtani, Ryan Chinn, and Hsinchun Chen. 2015. Exploring hacker assets in underground forums. In *Intelligence and Security Informatics (ISI), 2015 IEEE International Conference on*. IEEE, 31–36.
- [17] Jana Shakarian, Andrew T Gunn, and Paulo Shakarian. 2016. Exploring malicious hacker forums. In *Cyber Deception*. Springer, 261–284.
- [18] Reza Zafarani, Mohammad Ali Abbasi, and Huan Liu. 2014. *Social media mining: an introduction*. Cambridge University Press.