Predicting Cyber Threats through the Dynamics of User Connectivity 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

The existing methods for assessing the likelihood of exploitation for software vulnerabilities are found to have many limitation - preventing them from being useful tools for prioritization of vulnerability remediation. We present a method that combines social network analysis with machine learning techniques to predict the vulnerability exploitability. Our method harnesses features based on user connectivity in darkweb/deepweb sites as well as features derived from the vulnerability data. Our results suggest that the features computed from the user social connections are highly indicative of future cyber attacks. 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 structure related to certain exploit authors is indicative of their ability to write exploits that are subsequently employed in an attack.

1 INTRODUCTION

The majority of cyber-attacks across the world leveraged exploits for vulnerabilities known to the cybersecurity community. As a recent example, consider the ransomware WannaCry, which leverages vulnerabilities described in MS17-010. It was launched with high attack volume on May 12, 2017. Microsoft has released patches for that vulnerability weeks before the attacks were carried out. However, at least 230,000 computers in over 150 countries have been infected by that attacks [24]. In the same week when WannaCry's vulnerability was released by the National Institute of Standards and Technology (NIST) (with CVE-2017-0144), over 370 other vulnerabilities were also released. The WannaCry's vulnerability was assigned a Common Vulnerability Scoring System (CVSS) [11] base score v3.0 of 8.1 while 60 other vulnerabilities, released in the same week, were assigned CVSS scores more than 8.1, which explains why this vulnerability has not received much of attention at the time of disclosure, weeks before the attacks. The existing vulnerability severity scoring systems are known not to be useful for prioritization of vulnerability remediation. They are known to assign high severity scores to many vulnerabilities that will never be exploited [2]. Recent studies show that less than 3% of the vulnerabilities reported are ever exploited in wild [1, 2, 20]. Therefore, the cybersecurity researchers have used alternative methods leveraging machine learning techniques on features computed from social media feeds with explicit vulnerability mentions [6, 20]. In both studies, the dynamics of user connectivity have not been analyzed nor have they quantified as predictors. Additionally, while other studies suggest that darkweb/deepweb (D2web) sites are among the best sources for gathering cyber threat intelligence [2, 12, 14, 22], the value of such intelligence in predicting the vulnerability likelihood of exploitation was not quantified.

In this paper, we introduce our method that leverage machine learning models trained on features derived from the social network of users participating in D2web forums as well as features derived from the vulnerability information. We evaluate our method using ground truth obtained from attack signatures reported from a large cybersecurity firm, namely Symantec[1]. In this paper, we provide three case studies illustrating that the vulnerability mentions recorded on D2web indicate that exploits are being developed and recirculated. These exploits are employed in real-world attacks detected in the wild short time after the recorded events. We also propose an approach for generating a social network graph from the discussions on D2web forums, and provide analysis on the dynamics of connectivity. Furthermore, we demonstrate the viability of the features derived from the social network of D2web users in predicting cyber threats, and we use the task of predicting the likelihood of exploitation to show that these features, combined with other features, result in machine learning model that achieves F1 measure of 0.67 and generalize well with the biased ground truth.

2 BACKGROUND

2.1 Vulnerability Lifecycle

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

Vulnerability Reporting. Researchers are encouraged to report security flows they discover to any CVE Numbering Authorities (CNAs)[2]. Once a new flow is reported to a CNA, the CNA reserves a new Common Vulnerabilities and Exposures (CVE) number after validating that the flow can be exploited in a way that violates the security policies, and it has not been previously assigned a CVE number. Once it is established that the flow is a software vulnerability, the software vendor is notified and allowed a period of time to release patches before any information about the vulnerability is publicly disclosed. To allow for validating the vulnerability, a Proof-of-Concept (PoC) exploit might be developed as part of the reporting process. PoCs are scripts written by white-hat hackers or software vendors with limited functionality aiming to show that a

[1] https://www.symantec.com
[2] For a complete list of CNAs, see https://cve.mitre.org/cve/cna.html.

flow is exploitable. Some researchers and software vendors choose to make PoCs exploits available for the public community; hence, submitting them to PoC public archives such as Exploit-DB[3]. Possibly, some PoCs are also developed such that they can be integrated with penetration testing tools (e.g., MataSploit[4]) to allow pen testers to identify vulnerable systems within their organizations to plan remediations.

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)[5] releases information about the vulnerability to the cybersecurity community through the National Vulnerability Database (NVD)[6]- which is known to be the comprehensive reference vulnerability database with over 92,000 disclosed vulnerabilities as of July 2017. Along with the CVE number of the vulnerability, other details are provided (e.g., description, effected system/software, CVSS severity score).

Vulnerability Exploitation. After a vulnerability is disclosed by NIST, ideally, organizations start patching their systems to avoid any risks. However, with the ever-increasing number of disclosed vulnerabilities, many organizations fall behind on vulnerability mitigation and become exposed to a wide range of newly deployed exploits exist in the wild. Recent studies show that only a small fraction of vulnerabilities are found to be exploited in the wild (1-3% [2, 10, 15, 20]); nonetheless, exploits are being detected in the wild short-time after disclosure. For example, exploits for the well-known vulnerability ("Heartbleed", with CVE-2014-0160) were detected in the wild less than a day after its public disclosure [9]. Therefore, organizations look for methods help in assessing the exploitability likelihood better than the existing severity scoring systems, which have been criticized for their tendency towards assigning high scores for most vulnerabilities, resulting in high false positive rates (i.e., long lists of vulnerabilities to patch - the vast majority will never be exploited) [2].

2.2 Case Study: D2web Vulnerability Events

Different reactions from the cybersecurity communities are spotted in the online social networking platforms (e.g., blogs, Twitter, deepweb/darkweb (D2web) forums) after a vulnerability is publicly disclosed. Here, we only focus on the events appeared in D2web with explicit reference to known vulnerabilities. We include three timelines that trace the events surrounding three different vulnerabilities to show that D2web vulnerability events might be recorded across multiple sites and by different users.

The first of these timelines, depicted in Figure 1, has many events that can be traced across multiple D2web sites over a long period. In late January, 2016, the vulnerability in question was posted by users 1, 2, and 3 on forums *A*, *B*, and *C* respectively. By the next day, it had been posted by another user 4 on another D2web forum *D*. On February 7th, 2016, NIST formally disclosed the vulnerability (with CVE-2016-0728). On the same day, another post regarding the vulnerability was made by user 5 on forum *D*. Exactly one week later, on February 14th, Symantec first detected an exploit that targets CVE-2016-0728. Then, over four months later, an exploit

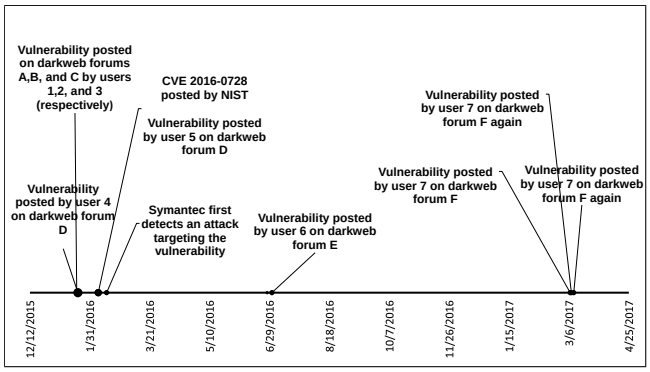

Figure 1: A vulnerability (with CVE-2016-0728) that has been posted by many users on various different darkweb sites.

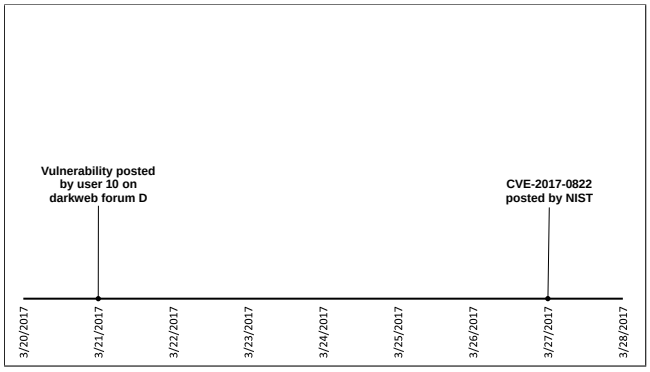

Figure 2: A vulnerability (with CVE-2017-0822) that has only been posted on one darkweb site by just one user.

targeting the vulnerability first appeared on a D2web marketplace. Few days later, the vulnerability was posted by user 6 on D2web forum *E*. Much later, the vulnerability was posted by user 7 on D2web forum *F* three times in the second week of March 2017.

Figure 2 shows an example of a vulnerability that has very little activity surrounding it. It was first posted by user 10 on D2web forum *D* late March 2017. Later, within the same week, NIST released the vulnerability information (with CVE-2017-0822). These two events are the only known activities regarding this particular vulnerability. It has not been detected any exploits in the wild targeting CVE-2017-0822.

The third and final timeline, illustrated in Figure 3 is an example of a vulnerability that has only been posted and discussed by a few users. Hence, it can be traced over only a select few sites. This timeline begins on September 18th, 2016 when the vulnerability was first disclosed by NIST (with CVE-2016-6415). A few days later, on September 24th, the vulnerability was posted by user 8 on a darkweb forum D. Just two days following this (September 26th), it was posted again on darkweb forum D but this time by a different user 9. This vulnerability exists in the server *IKEv1* implementation in Cisco IOS. Although this vulnerability does not have an attack signature reported by Symantec, some reports indicate

[3]https://www.exploit-db.com.
[4]https://www.metasploit.com.
[5]https://www.nist.gov.
[6]https://nvd.nist.gov.

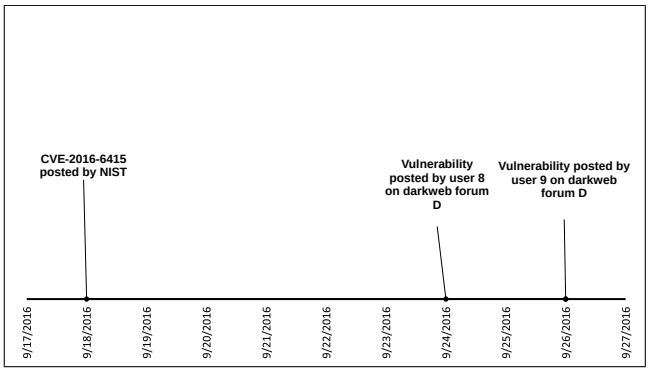

Figure 3: A vulnerability that has been posted multiple times on the same darkweb site by different users.

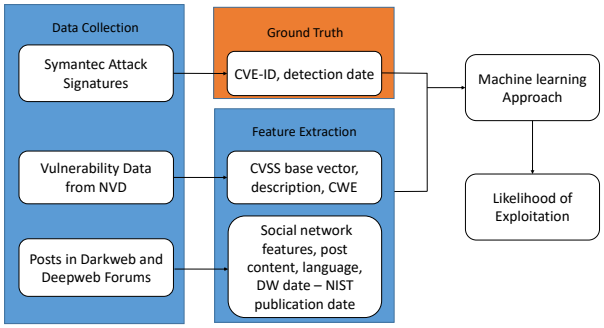

Figure 4: An overview of the prediction model.

that this vulnerability was exploited in the wild [7] – though the exact exploitation date is not known. This supports the observation discussed in previous work [20] regarding the bias of Symantec's attack signatures.

While higher number of vulnerability mentions across D2web sites is an indicative of possible risks (as illustrated in the first timeline), vulnerabilities with lower number of mentions may also be targeted (e.g, third timeline). Meta-data about the vulnerability mentions on D2web such as the importance of the user who referenced the vulnerability and the textual content of the posts are also other measures we use for assessing the likelihood of exploitation.

3 METHOD

We view the task 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*. Our approach is based on supervised learning with features computed from the social network of users posting in D2web forums as well as other features derived from data feeds from NVD. Figure 4 gives an overview of our proposed approach for predicting the vulnerability exploits in the wild.

[7] See http://www.securityfocus.com/bid/93003/exploit.

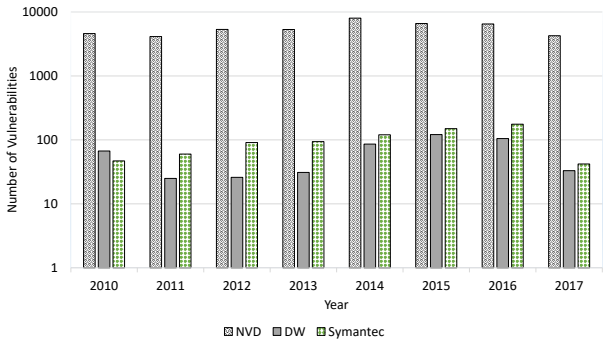

Figure 5: Number of vulnerabilities disclosed on NVD, vulnerabilities mentioned on D2web, and vulnerabilities reported as *exploited* on Symantec by NVD publication year vs. the number of vulnerability each data source reported.

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 [17]. The class labels are determined based on a ground truth set of attack signatures of exploits detected in the wild, and reported by the well-known cyber defense corporation, Symantec[8]. For the purpose of this paper, we focus our study on data collected from all the mentioned sources between January 2010- March 2017. Here we provide details on the data collection process for the three sources we use. Figure 5 shows the number of vulnerabilities published on NVD per year as well as the number of these vulnerabilities mentioned on D2web and reported by Symantec.

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 scrapper is developed to retrieve from the vulnerabilities' webpages the data elements that are missing from the JSON files (e.g., the disclosure date).

D2web Forums. We use a database of posts collected from 151 darknet and deepnet forums. The data collection system is originally described in [17]. D2web sites are periodically scraped by the system to collect postings with hacking-related content. A team of experts first identify D2web sites that have content of interest. After it is determined that a website is of interest, a customized crawler and parser is developed, and it is put forward for periodic data collection. Although the focus is only on sites with rich hacking-related content, some discussions irrelevant to hacking (e.g., discussions related to illicit drugs, pornography) are also collected. To filter data relating to hacking from other data, a machine learning model with high classification accuracy is used on features derived from the content of posts. The database we use contains over 2,290,000 posts under 223,074 distinct topic in 151 distinct forums. From these posts, only 3,082 have explicitly mentioned 624 distinct vulnerabilities by referencing their CVE-ID. Of those,

[8] https://www.symantec.com

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-virus[9] or Intrusion Detection Systems' attack signatures[10]. The fraction of exploited vulnerabilities is found to be very small as shown in Figure 5, 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 [2, 20].

3.2 Features Description

Features are extracted from both data sources we discussed in Section 3.1. We summarize the features in Table 1. Here we provide discussions on each of the sets of features.

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

CVSS Base Score. CVSS is a vulnerability severity scoring framework designed to measure the exploitability and the impact of the vulnerabilities. We use CVSS base score versions v2.0. There are two main 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, Authentication, Confidentiality Impact, Integrity Impact*, and *Availability Impact*. Each of these measures can take one of different categories. For example, *Access Vector* indicates how the vulnerability is exploited. It can take one of three possible levels: *Local (L), Adjacent Network (A) and Network (N)*[11].

NVD Description. NIST provides a textual description of the vulnerability when it is released. The description summarizes the system/software in which the flow exists and gives information on how it can be exploited. The textual features derived from NVD description undergo preprocessing pipeline including stemming (reducing the words to their root forms) and stop words removal (e.g., *and, or, then*). Then the text is vectorized using Term Frequency-Inverse Document Frequency (TF-IDF), which computes the importance of words in a document by comparing the frequency of the word in a document with the length of that document and comparing it with the frequency of the word in all other documents. Thus, the more times a word occurs is a document and less times it occurs in other documents the higher TF-IDF score assigned. Only the 250 most frequent words are used as features to limit computational cost.

CWE. Is a community-effort project comprising enumerating common software security weaknesses (categorical). These are categories of flows that can be unintentionally made during the software development and can exist in the software architecture, design, or code[12].

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 (can be viewed as importance) 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 NVD description features are extracted except that for non-English posts, we automatically translate the content to English using Google Translate API[13], then the TF-IDF are computed as described earlier.

3.3 Classifier Training and Prediction

We use a supervised machine learning approaches to train classifiers on the said 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

Previous work proposed for predicting the vulnerability likelihood of exploitation has mainly examined features derived from vulnerability data from NIST (e.g., number of references [6, 20], CVSS score [2, 6, 20]), or from online mentions (e.g., tweets [6, 20], Exploit-DB [20]). Features derived from the social network of users in these platforms have not been examined on neither their correlation nor causality of vulnerability exploitation. In this paper, we leverage social network theory to analyze the social network of users who are actively posting malicious hacking-related content in D2web. Such features have been widely investigated in the literature in applications of cyber threat intelligence, but for tasks other than predicting threats (for details, see Section 7.2). In this paper, we adopt the same assumption made in much of the previous work on D2web data, where they consider the same usernames (case

[9] A complete list is found here https://www.symantec.com/security_response/landing/azlisting.jsp. The detection date is labeled with "Discovered". For example, see https://www.symantec.com/security_response/writeup.jsp?docid=2017-031318-1819-99.

[10] https://www.symantec.com/security_response/attacksignatures.

[11] see https://www.first.org/cvss/v2/guide for complete documentation.

[12] For example, CWE-119 refer to a very common vulnerability type, improper restriction on the bounds of memory buffers. A hacker can exploit vulnerabilities of this type by overwriting data in memory then the system will crash.

[13] https://cloud.google.com/translate. Amongst different translation services examined, this appears to be the best in our study.

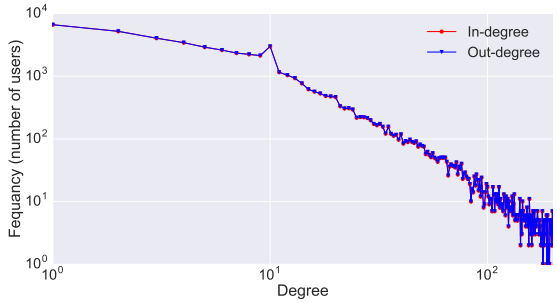

Figure 6: In-degree and out-degree distribution of users in G.

insensitive) across different D2web sites to belong to the same person(s) [19]. This allows for generating one network comprising a large number of D2web sites as opposed to a social network for each site [**?**].

4.1 Users' Social Graph

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

Considering only posts between January 2010- June 2016 (747,351 posts under 109,413 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.

Hence, there is no self-loops, only one edge can be created from v_i to v_j (i.e., v_i posted after v_j) under topic t_i on date d. 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.

Degree distributions, for both incoming and outgoing edges, of G are found to resemble power-law distribution as depicted in figure 6. 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 [25].

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 user. These measures describe the type of activities in which 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, Betweeness 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 the in-neighbors such that each of v_i's in-neighbors passes the value of his/her centrality divided by the number of outgoing edges. *Pagerank* is the algorithm used by Google search engine to rank results, relevant to search keywords, based on their importance [18].

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., $vulnUsers \subset V$), (2) the subgraph of G induced by $vulnUsers$, $G_{vulns}(vulnUsers, vulnEdges)$, (s.t., $vulnEdges \subset E$), and (3), a subgraph induced by $vulnUsers$ as well as all their in-neighborers 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 dynamics of user connectivity for the subset of users mentioned vulnerabilities (s.t., $vulnUsers$, less than 1% of the total population) as opposed to the total user population.

These individuals, or *vulnUsers*, 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 *vulnUsers* 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 *vulnUsers* is exposed to a larger population of hackers than a normal hacker. Additionally, we observe that a hacker from *vulnUsers* 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

Table 2: Statistics for graph G with all users with at least one edge (in-edge or out-edge), the subset of users discussed vulnerabilities *vulnUsers*, subgraph ($G_{vulns}(vulnUsers, vulnEdges)$)– induced by users *vulnUsers*, and subgraph ($G_{vulNei}(vulNeis, vulNeiEdges)$)– induced by *vulnUsers* as well as all their in-neighborers and out-neighbors

Property	D2web users	Users mentioned vulnerabilities		
		as a subset of nodes	as a subgraph	as a subgraph with all 1-hub neighborers
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	$2.59e^{-4}$	$9.68e^{-4}$	$1.12e^{-2}$	$1.84e^{-3}$
Betweeness 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	$1.90e^{-5}$	$1.27e^{-4}$	$2.74e^{-3}$	$9.6e^{-5}$

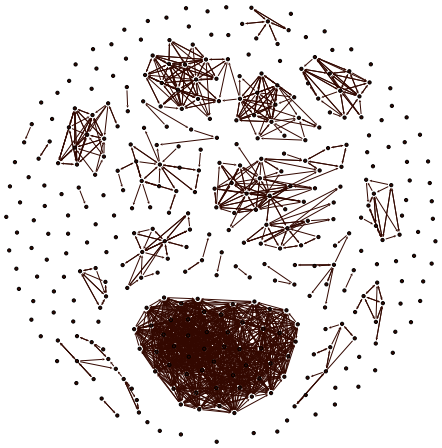

Figure 7: The subgraph of G induced by the set of users mentioned vulnerabilities in their postings.

other *vulnUsers* (about 1%); whereas he/she would reply to posts by about 51 users and make about 64 other users reply to his/her post - in both cases, this is less than 0.1% of the total population. For these reasons, *vulnUsers* 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 *vulnUsers* have no communication history with other users within the same group. Figure 7 shows a visualization for the subgraph G_{vulns}, which confirms the two observations. Finally, about 25% of *vulnUsers* 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 because these can 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 compared, but we only report the performance achieved by the best performing classifier, which is Random Forest (RF) [5]. RF generates an ensemble of decision trees, each trained on a randomly sampled subset of the training data - the achieved performance of different runs slightly differs; thus, for each of the experiments, we report the average of five runs. The said decision trees are then used in combination to classify vulnerabilities, each tree provides an independent classification opinion; collectively producing the confidence score. We use a RF classifier that combines bagging [5] for each tree with random feature selection at each node of the tree to split the data. The confidence score, along with other measures (e.g., the cost of patching, the estimated impact if the vulnerability is exploited) can be used for patch prioritization, or a threshold can be set as a decision boundary.

6.1 Performance Evaluation

We evaluate the performance of the models 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 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 negative.

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

We also report Receiver Operating Characteristics (ROC) curve as well as 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 various thresholds of the confidence scores the classifier outputs.

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). The classification performance achieved by our RF model is at averaged precision of 0.57, recall of 0.93, and F1 of 0.67. The same classifier is able to achieve an averaged AUC of 0.95. The lower score of precision is attributed to the fact that Symantec's data is biased towards reporting exploits targeting vulnerabilities exist in software products from certain software vendors such as Microsoft, Adobe [20]. Since our model is found to predict vulnerabilities as being exploited from other vendors as well, we believe that some false positives were actually exploited in the wild but never detected by Symantec (the third timeline in Section 2.2 is an example).

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 - 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 averaged F1 achieved is 0.72, with a precision of 0.61 and a recall of 0.89, and with AUC of 88%. To measure the impact of individual feature sets on the overall classification performance, we apply two tests: (1) an ablation test (depicted in Figure 8) 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 9) where the classification performance is reported for models trained on only one set features at a time. In both tests, the set of social network measures shows some performance decrease when it is removed from model, and large improvement compared to other sets of features and compared to the simple classifier - which labels all vulnerabilities are *exploited*, resulting in a precision of 0.16, a recall of 1, at an F1 of 0.27 and an AUC of 0.5.

6.3 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 assigning values to the different components of the vector is manually created by domain experts[14]. 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 - which account for the 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 to potentially over-fit the biased ground truth. In all experiments, social network measures demonstrated their viability as predictors of potential cyber threats.

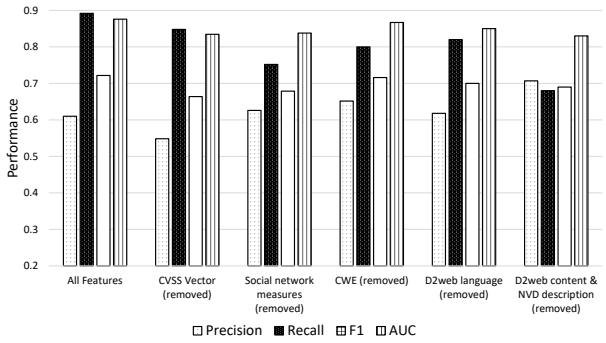

Figure 8: Classification performance achieved by applying ablation test with 5-fold cross-validation.

7 RELATED WORK

7.1 Vulnerability Exploitation Prediction

Vulnerability remediation involves crucial decisions. For example, applying patches can disrupt important business processes if it involves taking some systems down to apply patches. Thus, applying unimportant patches is undesirable. A recent study found that CVSS base score metrics, the most popular reference measures of vulnerability severity, are poor indicators of exploitation [2]. Many vulnerabilities are assigned high scores, resulting in very high false positive rates (long lists of vulnerabilities to patch, most will never be exploited). Additionally, methods have been proposed recently to assess the vulnerability exploitation likelihood using vulnerability disclosure data as well as other data sources, a problem we tackle in

[14]See https://www.first.org/cvss/v2/guide.

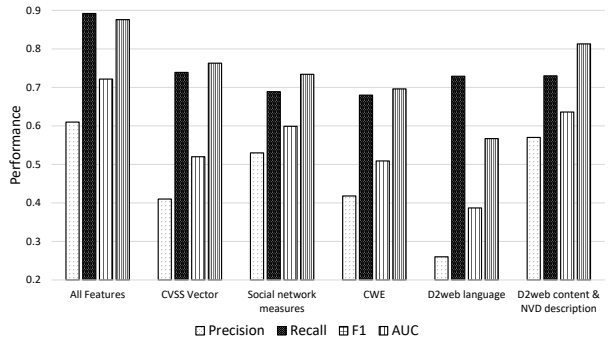

Figure 9: Classification performance achieved by individual feature sets.

this work. Some of these methods leverage machine learning techniques to predict whether a vulnerability will be exploited. Sabottle et al. [20] 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 [6] to measure their impact on real-world proactive exploitation prediction settings. The work in [6] replicates the experiments done in [20] (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. When examined under real-world conditions, the value of such data feeds is found to be questionable [6]. In both papers, the dynamics of user connectivity are not studied.

7.2 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 [4, 8, 16, 23]. For example, Cao et al. [7] propose a method called *SybilRank* that relies on social network measures to identify fake accounts (Sybils) in large scale social media platform. *SybilRank* is demonstrated to outperform other used approaches by Tuenti, a social media platform that was popular in Spain, by 18 times increase in recall. 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 [3, 13, 21]. However, the dynamics of connections within the hacker communities have not been quantified as predictors for vulnerability exploitation. We find that the measures computed from social connections to be promising predictors of future cyber attacks.

8 CONCLUSION

Our work contributes towards the understanding the dynamics of user connectivity in D2web forums, and extracting measures that serve as predictors of cyber attacks. We use the task of predicting the vulnerability likelihood of 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. 2012. A preliminary analysis of vulnerability scores for attacks in wild: the ekits and sym datasets. In *Proceedings of the 2012 ACM Workshop on Building analysis datasets and gathering experience returns for security*. ACM, 17–24.

[2] 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.

[3] 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.

[4] 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.

[5] Leo Breiman. 2001. Random Forests. *Machine Learning* 45, 1 (2001), 5–32.

[6] 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.

[7] 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.

[8] Qiang Cao, Xiaowei Yang, Jieqi Yu, and Christopher Palow. 2014. Uncovering large groups of active malicious accounts in online social networks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 477–488.

[9] Zakir Durumeric, James Kasten, David Adrian, J Alex Halderman, Michael Bailey, Frank Li, Nicolas Weaver, Johanna Amann, Jethro Beekman, Mathias Payer, and others. 2014. The matter of heartbleed. In *Proceedings of the 2014 Conference on Internet Measurement Conference*. ACM, 475–488.

[10] Michel Edkrantz and Alan Said. 2015. Predicting Cyber Vulnerability Exploits with Machine Learning.. In *SCAI*. 48–57.

[11] FIRST. Last Accessed: May 2017. A Complete Guide to the Common Vulnerability Scoring System. https://www.first.org/cvss/v2/guide. (Last Accessed: May 2017).

[12] Thomas J. Holt and Eric Lampke. 2010. Exploring stolen data markets online: products and market forces. *Criminal Justice Studies* 23, 1 (2010), 33–50. DOI:https://doi.org/10.1080/14786011003634415 arXiv:http://dx.doi.org/10.1080/14786011003634415

[13] 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.

[14] 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

[15] Kartik Nayak, Daniel Marino, Petros Efstathopoulos, and Tudor Dumitraş. 2014. Some vulnerabilities are different than others. In *International Workshop on Recent Advances in Intrusion Detection*. Springer, 426–446.

[16] Mariam Nouh 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.

[17] 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.

[18] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web.* Technical Report. Stanford InfoLab.

[19] John Robertson, Ahmad Diab, Ericsson Marin, Eric Nunes, Vivin Paliath, Jana Shakarian, and Paulo Shakarian. 2017. *Darkweb Cyber Threat Intelligence Mining*. Cambridge University Press.

[20] 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.

[21] 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.

[22] Jana Shakarian, Andrew T Gunn, and Paulo Shakarian. 2016. Exploring malicious hacker forums. In *Cyber Deception*. Springer, 261–284.

[23] Gianluca Stringhini, Pierre Mourlanne, Gregoire Jacob, Manuel Egele, Christopher Kruegel, and Giovanni Vigna. 2015. EVILCOHORT: Detecting Communities of Malicious Accounts on Online Services.. In *Usenix Security*. 563–578.

[24] Wikipedia. Last Accessed: May 2017. WannaCry ransomware attack. https://en.wikipedia.org/wiki/WannaCry_ransomware_attack. (Last Accessed: May 2017).

[25] Reza Zafarani, Mohammad Ali Abbasi, and Huan Liu. 2014. *Social media mining: an introduction.* Cambridge University Press.