Threat Class Predictor: An explainable framework for predicting vulnerability threat using topic and trend modeling

François Labrèche[1], Serge-Olivier Paquette[1]

[1]Secureworks

Abstract

Every day, an increasing number of new software is found to be vulnerable to exploitation. Such vulnerabilities are disclosed through publicly available databases, such as the National Vulnerability Database (NVD). However, the rate of disclosures now far outpaces the ability of any single research team or remediation team to handle them all. In this paper, we present a framework that not only predicts the vulnerabilities that will be exploited by malicious actors or malware, but also which vulnerabilities can go under the radar, escaping the trending discussions of online cybersecurity communities. This is achieved by leveraging topic modeling in a novel way, combining a threat score and a trend score. The interpretable nature of such topic models enables security teams to dig deeper into the predictions of our model, making it a valuable tool for their remediation and investigative work.

Keywords

Attack prediction, Exploit prediction, Vulnerability prioritization

1. Introduction

We present an explainable machine learning framework to predict threats associated with disclosed vulnerabilities and better inform security professionals on potentially overlooked critical vulnerabilities. We first apply topic modeling to vulnerability descriptions to build a semantic representation of vulnerabilities. Using this representation, we train a multi-label threat prediction classifier for recently disclosed vulnerabilities. The model provides two independent threat predictions; a probability of either having a proof-of-concept/weaponized exploit code published, and/or of being included in malware. We combine these to obtain a threat score for each vulnerability. This score can be used to prioritize the remediation or investigation of vulnerabilities.

We also use the same topic model to create a novel trend score from online infosec discussions. This trend score, used in conjunction with the threat score, can inform security researchers on where to focus their attention, i.e., on the most interesting and potentially overlooked vulnerabilities. We do this by joining the two independent scores, the threat score and the trend score, visually, in a two-dimensional plane. Given the interpretable nature of topic models

CAMLIS'22: Conference on Applied Machine Learning in Information Security (CAMLIS), October 20–21, 2022, Arlington, VA

flabreche@secureworks.com (F. Labrèche); spaquette@secureworks.com (S. Paquette)
0000-0001-6027-3017 (F. Labrèche)

and our novel visual representation, we believe that our framework brings new value to the cybersecurity community by offering a method of prioritizing investigative work.

Our contributions are the following:

- We build a semantic representation of vulnerabilities based on the underlying concepts of all descriptions, which represents them in a more holistic way than what was previously done.
- Using this new representation, we compute an explainable threat score and trend score.
- We provide a threat dashboard which helps visualizing vulnerability trends in relation to the likelihood of an attack leveraging them.

The rest of the paper is as follows:

1. Section 2 presents prior work done in predicting threats and exploit publication using machine learning.
2. Section 3 describes the methodology used for this approach. The corresponding results are presented as the methodology is discussed. We first present the topic model, used both by the threat model and the trend model, before presenting each model respectively. This section closes with the combination of the two scores in a visual dashboard.
3. Section 4 explores the trained models, their features and their explainable nature.

2. Related Works

A number of previous studies have built exploit prediction models using vulnerability features [1, 2, 3], such as the CVSS score and its sub-components, the Common Weakness Enumeration (CWE), the references, the description and the vulnerable products. While feature encodings vary, they all use supervised machine learning trained on NVD data to predict vulnerabilities labeled with exploits. Suciu *et al.* [4] employ a similar approach, but with the goal to predict over time the likelihood that a functional exploit will be developed. Other approaches [5, 6] explore using social network data and dark web discussions as additional features to predict the likelihood of an exploit targeting a vulnerability. Huang *et al.* [7] use Latent Dirichlet Allocation (LDA) to identify important words through six topics built on vulnerability descriptions, combined with a classifier that labels tweets as cybersecurity-related or not. Xiao *et al.* [8] employ community detection over botnet IP activity to identify if a vulnerability is being exploited. Additionally, however, Bullough *et al.* [9] identify key methodological errors in some of these previous works, most notably incorrect metrics used for evaluating an imbalanced dataset. Finally, others model the vulnerability description to predict the publication of an exploit or an attack, such as using tf-idf [10], neural networks [11], deep learning with CNNs [12] or a BERT pre-trained model [13, 14]. In this work, we build a vulnerability representation using topic modeling, which we then use to predict multiple threat classes and identify trending vulnerabilities. Contrary to previous approaches employing deep learning on vulnerability descriptions, our use of topic modeling provides an explainable framework which can provide insights into how different types of threats are linked to vulnerabilities.

3. Methodology and Results

3.1. Topic Model

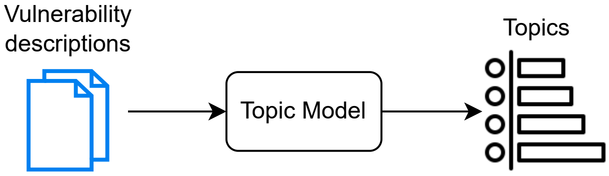

Figure 1: We extract topics from vulnerability descriptions using a topic model.

We obtain a topic model by training LDA [15] on the textual descriptions of 152,585 published vulnerabilities from the 1st of January 2008 to the 1st of August 2022. We prepare the corpus by removing all stop words, common words, and URLs. We lemmatize and tokenize the documents to obtain a bag-of-words representation to feed to the model. The number of topics is selected using a coherence score [16], a measure to compute the strength of the similarity of words inside a topic. A coherence score provides a robust way to evaluate topic models, in regards to interpretability by humans [17]. We obtain an optimal model with 30 topics, 50 iterations and 10 passes.

With this trained topic model, we now have a list of 30 topic probabilities $V^i = (v_1^i, ..., v_{30}^i)$ with real numbers $v_j^i \in [0, 1]$ and $\sum_{j=1}^{30} v_j^i = 1$ representing each topic probability j for every vulnerability V^i in our dataset. Examples of six extracted topics, visualized as word clouds with weighted words, are presented in Figure 2. Each topic corresponds to a set of words, where larger (higher probability) words are more salient inside the topic.

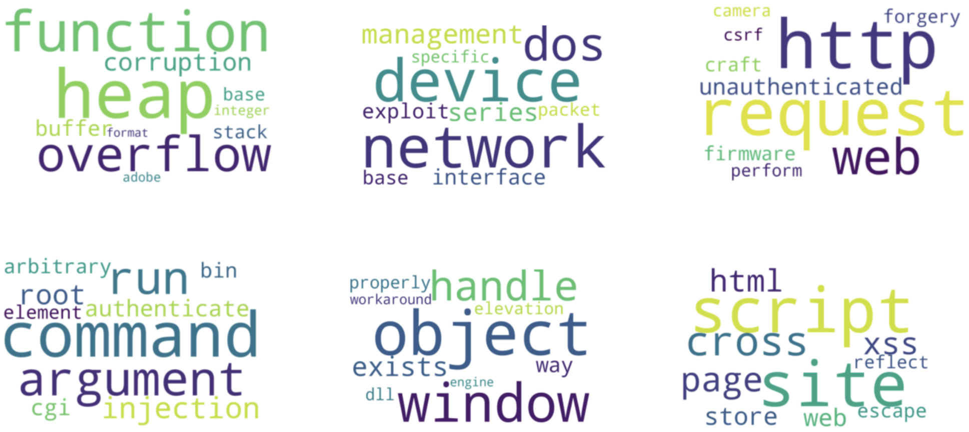

Figure 2: Six topics extracted from the topic model trained on vulnerability descriptions.

3.2. Threat Class Prediction Model

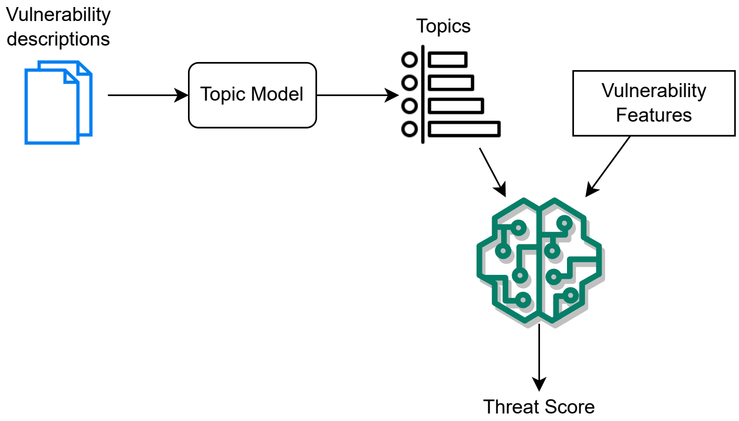

Figure 3: Predicting threats using topics and vulnerability features.

3.2.1. Feature Selection

We build a threat class predictive model[1], using the topics from the descriptions above, and details from vulnerability disclosures on the National Vulnerability Database (NVD)[2] as features. Categorical features are encoded as dummy variables.

The list of additional features used is the following and follows previous works[1, 5, 2, 3, 4, 7]:

- The length of the description,
- The number of references available for the vulnerability at the time of publication,
- The number of software configurations affected by this vulnerability,
- The CVSSv2 score,[3]
- The CVSSv2 metrics.

3.2.2. Dataset

As previously mentioned, the dataset used to build our features is the NVD. The two threat classes that we predict are exploit publication and malware inclusion. These two classes have been chosen because they represent key cybersecurity threats and labels for them can be found openly. Although they do overlap, they do not do so completely. Each threat class uses its own datasets for labels. Exploit publications are labeled from exploitDB, Packetstorm and a Github repository listing POCs[4], all of which are publicly available. ClamAV [18] signatures are used

[1] Patent pending

[2] https://nvd.nist.gov/

[3] There is a larger body of vulnerabilities published with a CVSSv2 score, however, the CVSSv3 score can also be used

[4] https://github.com/nomi-sec/PoC-in-GitHub/blob/master/README.md

for malware labels, which are also publicly available. We join these signatures to a database of malware threat intelligence reports from the Counter Threat Unit™ (CTU)[5].

The datasets are summarized in Table 1. There are 835 vulnerabilities that overlap between the exploits and malware labels. The classes are highly imbalanced and are not mutually exclusive, hence we train two independent classifiers, which both output a probability between 0 and 1. To obtain a single threat score, we add their outputs to obtain a value between 0 and 2.

Table 1
Datasets summary

Dataset	N samples
CVE Database	152585
ExploitDB	22441
Packetstorm	5471
Github POCs	3219
ClamAV	2956
CTU	184

3.2.3. Model Selection

We train a classifier for each of the two threat classes. We tested a Logistic Regression model, a Support Vector Machines and a Random Forest classifier, using 10-fold cross-validation. Table 2 shows the results for each of the three classifiers.

Table 2
Averaged results per model

Algorithm	Accuracy	Precision	Recall	F1-Score	F2-Score
Random Forest	98.68	54.78	82.31	65.41	74.45
SVM	83.65	23.92	81.96	36.16	53.34
Logistic Regression	83.52	23.74	81.81	35.95	53.12

By far, the random forest model exhibits the best performances. To compensate for the class imbalance, we used class-weight optimization and threshold-moving based on the F2-score, a performance metric that optimizes for recall on the minority class, which is suitable for our need. Threshold-moving lets us choose the threshold on which to assign the model output class using the class probability. A high recall measures the ability of the model to predict the positive class and to avoid false negatives, but at the price of potentially having more false positives, which is an acceptable cost for identifying true attacks.

An important note is that our label datasets, while sufficient for training models that can correctly identify a majority of our samples, do not include all exploits and malware samples in

[5] Although we obtained better results using the CTU™ database, one can get similar but slightly lower results with the ClamAV database alone, or potentially with other public sources.

the wild, hence the true precision of the model is likely higher given that many false positives are in fact true positives.

Using grid search over the model parameters and 10-fold cross validation, we obtained a final model for which the performance and parameters are presented in Table 3.

Table 3
Threat Prediction Evaluation Results

Exploit Publication		**Malware Inclusion**	
Metric	**Value**	**Metric**	**Value**
Accuracy	**88.81% (+/- 0.16%)**	Accuracy	**98.01% (+/- 0.13%)**
Recall	**79.92% (+/- 0.99%)**	Recall	**87.96% (+/- 2.08%)**
Precision	36.92% (+/- 0.49%)	Precision	47.77% (+/- 1.76%)
F1-Score	50.51% (+/- 0.60%)	F1-Score	61.90% (+/- 1.71%)
F2-Score	64.82% (+/- 0.75%)	F2-Score	75.27% (+/- 1.71%)
Threshold	0.34	Threshold	0.46
Parameter	**Value**	**Parameter**	**Value**
max-depth	30	max-depth	50
min-samples-leaf	8	min-samples-leaf	6
min-samples-split	22	min-samples-split	16
n-trees	300	n-trees	200

3.2.4. Features Must be Chosen Carefully

Our initial prediction model, which was discarded, included a number of time-sensitive features inspired by the literature :

- The published date of the vulnerability,
- The date of its last modification,
- The number of online discussions related to the vulnerability.

Although this version of the model performed better in our training phase, with a higher accuracy and recall than our current model, it performed poorly in a real implementation by not predicting any instances of the positive classes. After investigation, three of the top five most impactful features were time-sensitive features, which skewed our model to better predict older vulnerabilities (i.e., newly published vulnerabilities rarely have a modification date and the publication date is always recent). In the end, including time-sensitive features was found irrelevant to our task of predicting threats for new vulnerabilities, and were discarded, even though the model performed better when evaluated on historical data.

3.3. Trend Model

We then compute a trend score $T(V^i, D^j) \in [0, 1]$ for each published vulnerability V^i on day D^j based on how closely its computed description topics match those in online discussions from that day. This numerical value is obtained by generating trending topics using the same LDA model previously trained for the threat class predictor.

Each day, we apply the topic model to a set of relevant online discussions, social media posts and dark web forum posts related to hacking and cybersecurity, in order to obtain an average for each topic value over all posts in a 30 day time window.

3.3.1. Dataset

We obtain these discussions through the Twitter API, Reddit API and Flare[6] API, a data provider who specializes in crawling dark web forums[7]. In the first 6 months of 2022, we searched for the following keywords on Twitter, Reddit and 90 dark web forums: *CVE-2013* to *CVE-2022*, *#infosec #vulnerability*, *#infosec #exploit*. We searched the hashtag keywords only on Twitter, and in pairs, in order to avoid noise and unrelated comments. Out of this, we obtained 512,347 tweets, 13,114 dark web forum posts and 36,598 Reddit posts mentioning CVE ids or hashtag pairs.

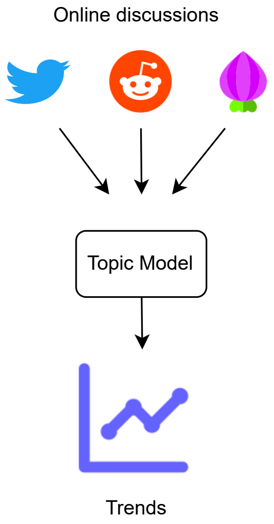

Figure 4: A trend predictor using the pre-trained topic model.

3.3.2. Obtaining a Stable Trend Score

Every day, we apply the LDA model to each sample, obtaining a topic weight vector $S^k = (s_1^k, ..., s_{30}^k)$. To obtain a raw trend value for a day $\hat{D^j}$ we average over all n topic vectors for that day j.

$$\hat{D}^j = \frac{1}{n}\sum_{j=1}^{n} S^k = \frac{1}{n}(\textstyle\sum_{j=1}^{n} s_1^j, ..., \sum_{j=1}^{n} s_{30}^j) \tag{1}$$

This process gives a trend vector of dimension 30 for each day, indicating the relevance of each topic to infosec discussions for that day. In order to dampen the variability between each day and to encode the momentum of evolving trends, we instead use a 30-day rolling average of the trend vector for each day.

$$D^j = \frac{1}{30}\sum_{k=j-30}^{j} \hat{D^k} = (d_i^j, ..., d_{30}^j) \tag{2}$$

The daily trend score of a single vulnerability $t(V^i, D^j) \in [0, 1]$ is obtained by computing the dot product of the 30-day averaged trend vector $D^j = (d_i^j, ..., d_{30}^j)$ with the vulnerability topic weight vector $V^i = (v_1^i, ..., v_{30}^i)$, which is a real number between 0 (not matching online discussions) and 1 (perfectly matching online discussions).

$$T(D^j, V^i) = \frac{1}{30}\sum_{k=i}^{30} d_k^j * v_k^i \tag{3}$$

[6] https://flare.systems/

[7] The most important ones are the exploit.in forum, xss.is, pediy, nulled.to and RaidForums.

A simplified version of this process is graphically presented in Figure 5.

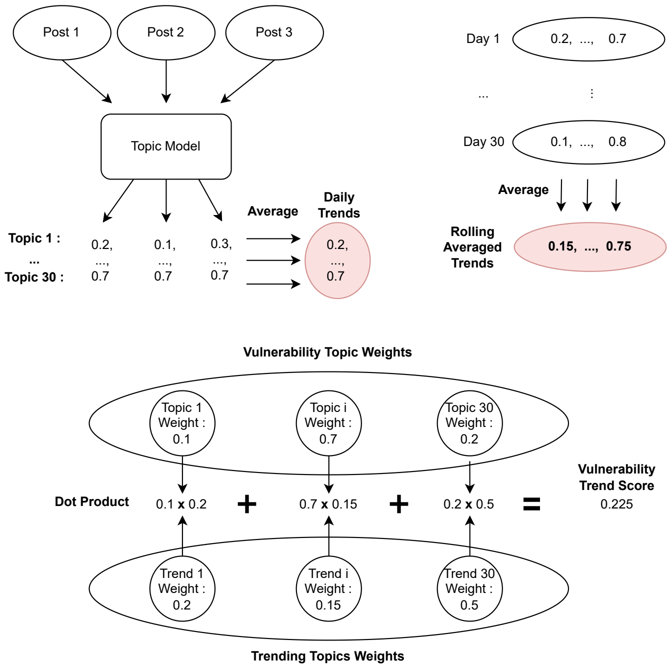

Figure 5: A simplified graphical representation of the trend computation.

3.3.3. Combining the Trend Scores and Threat Scores

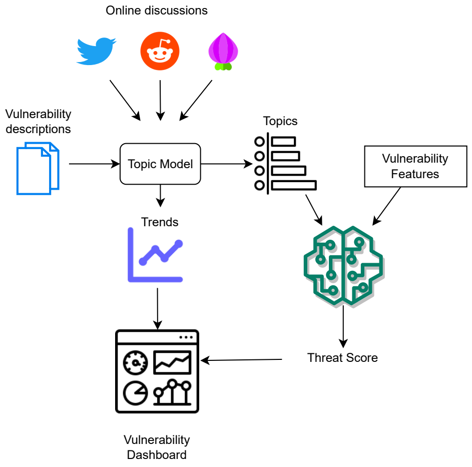

Figure 6: Combining the threat score and the trend score.

We also show the three most important topics according to their trend score at the time of writing, as well as their evolution through 2022 in Figure 7. A deeper analysis of the causes of the evolution of those trends is outside of the scope of this paper, and will be presented in a future study.

Finally, we wish to combine these two predicted values together to inform further research on the vulnerability disclosure. We obtain a visual representation by plotting, for each vulnerability, the two scores against each other in a two-dimensional plane, where the X-axis is the threat score and the Y-axis is the trend score, normalized from 0 to 1. An example of a subset of published vulnerabilities in the month of July 2022 is presented in Figure 8. Vulnerabilities on the right side are those predicted likely to have an exploit published and/or to be included in malware, while vulnerabilities in the lower half of the graph are those who do not match the trending online discussions. We believe those are the most interesting vulnerabilities for a researcher.

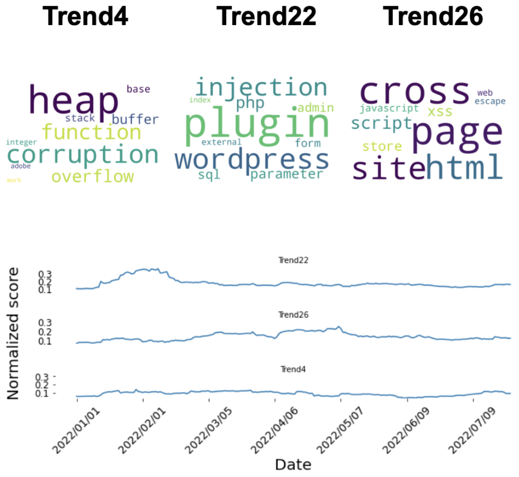

Figure 7: The evolution of three trends over time with the associated words.

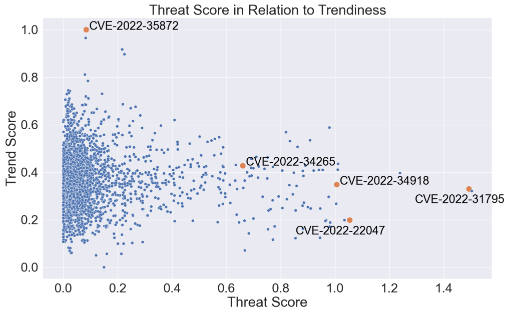

Figure 8: Threat score of vulnerabilities published in July 2022.

The following vulnerabilities have been correctly identified as having exploits published: CVE-2022-34265[8], CVE-2022-34918[9], CVE-2022-31795[10]. These vulnerabilities had exploits available outside of our datasets, and were identified by our prediction model. Additionally, the following vulnerability was identified in malware after its prediction: CVE-2022-22047[11]. An example of a vulnerability closely matching currently trending topics of remote code execution vulnerabilities is also shown: CVE-2022-35872. While some of the vulnerabilities identified are false positives, the total number of vulnerabilities to investigate has been considerably lowered and true attacks were successfully identified.

4. Discussion

4.1. An Explainable Framework

In this section, we show how a human can understand the decisions made by the models described in this approach. Our threat class predictor uses our generated topics as input features when predicting specific types of threats. For this reason, we can explore which concepts drive the fitted model, through the topics that come out as top features. More importantly, these topics vary per fitted model: they change depending on the type of threat we wish to predict. Below are shown, in Figure 9 and Figure 10, the top features for each model.

[8] https://github.com/aeyesec/CVE-2022-34265

[9] https://www.openwall.com/lists/oss-security/2022/07/05/1

[10] https://research.nccgroup.com/2022/05/27/technical-advisory-fujitsu-centricstor-control-center-v8-1-unauthenticated-command-injection/

[11] https://www.forbes.com/sites/daveywinder/2022/07/15/new-0day-hack-attack-alert-issued-for-all-windows-users

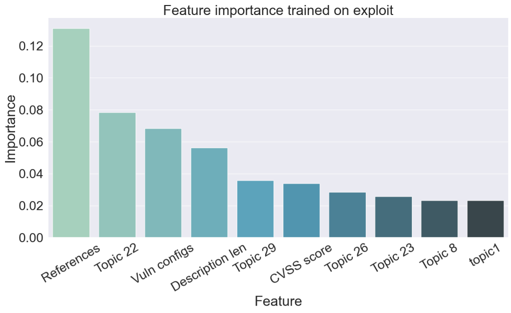

Figure 9: Top features when predicting exploit publications

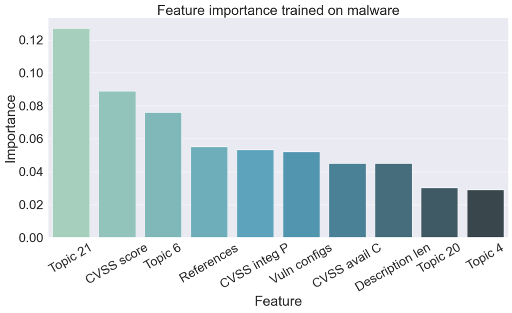

Figure 10: Top features when predicting malware inclusion

As can be observed, when predicting exploit publications, the number of references, the number of vulnerable configurations and the length of the description impact the model. The six most impactful topics, along with their most salient words, are:

1. **Topic 22** - Parameter, Plugins and SQL injections (*plugin, wordpress, injection, php, parameter, sql, admin*)
2. **Topic 29** - Google and OAuth Vulnerabilitiess (*prior, google, extension, convince, vector, agent, unknown, storage*)
3. **Topic 26** - Cross-Site Scripting (XSS) vulnerabilities (*page, cross, html, site, script, xss, store, javascript*)
4. **Topic 23** - Denial of Service (DOS) vulnerabilities (*service, cause, denial, null, pointer, dereference, crash, craft*)
5. **Topic 17** - Web vulnerabilities (*request, http, web, perform, forgery, unauthenticated, csrf, craft*)
6. **Topic 8** - Vulnerabilities centered around network attacks (*series, interface, network, device, management, dos*)

The top topic, understandably, refers to command injections, which is a common way of exploiting a vulnerability. We see as other topics more common techniques used in public exploits.

The top features used to predict malware are different, with Topic 21 about Windows handles appearing as most impactful. The following topics are the most influential in the prediction of vulnerabilities included in malware:

1. **Topic 21** - Vulnerabilities including the use of Windows handles (*object, window, engine, exists, handle, git, dll*)
2. **Topic 6** - PDF vulnerabilities (*module, update, upgrade, pdf, reader, zone*)
3. **Topic 4** - Heap and buffer overflow vulnerabilities (*heap, corruption, function, overflow, buffer, stack*)

Vulnerabilities centered around the exploitation of processes are more impactful when predicting malware, which contrasts with the prediction of exploits where specific types of attacks influence the prediction model.

The explainability of the framework goes even further, as one can obtain the trend score of a given vulnerability for each topic. A security researcher can thus explore the topics of a vulnerability or set of vulnerabilities and identify if it appears overlooked in online infosec discussions. A vulnerability can also be identified as part of a hype wave with respect to certain semantic characteristics.

5. Conclusion

In this research, we presented a coherent and explainable framework to predict the threat associated with a vulnerability, both from an exploitability perspective and from a semantic tendency perspective. Our results showcase vulnerabilities with a high likelihood of being included in real attacks that may appear overlooked by the cybersecurity community. The results of this paper show that we can easily achieve this using mainly open source data, with well-known and interpretable techniques.

References

[1] M. Bozorgi, L. K. Saul, S. Savage, 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, 2010, pp. 105–114.

[2] M. Edkrantz, A. Said, Predicting cyber vulnerability exploits with machine learning., in: SCAI, 2015, pp. 48–57.

[3] J. Jacobs, S. Romanosky, I. Adjerid, W. Baker, Improving vulnerability remediation through better exploit prediction, Journal of Cybersecurity 6 (2020) tyaa015.

[4] O. Suciu, C. Nelson, Z. Lyu, T. Bao, T. Dumitras, Expected exploitability: Predicting the development of functional vulnerability exploits, arXiv preprint arXiv:2102.07869 (2021).

[5] C. Sabottke, O. Suciu, T. Dumitraș, Vulnerability disclosure in the age of social media: Exploiting twitter for predicting {Real-World} exploits, in: 24th USENIX Security Symposium (USENIX Security 15), 2015, pp. 1041–1056.

[6] N. Tavabi, P. Goyal, M. Almukaynizi, P. Shakarian, K. Lerman, Darkembed: Exploit prediction with neural language models, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 32, 2018.

[7] S.-Y. Huang, T. Ban, Monitoring social media for vulnerability-threat prediction and topic analysis, in: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), IEEE, 2020, pp. 1771–1776.

[8] C. Xiao, A. Sarabi, Y. Liu, B. Li, M. Liu, T. Dumitras, From patching delays to infection symptoms: Using risk profiles for an early discovery of vulnerabilities exploited in the wild, in: 27th USENIX Security Symposium (USENIX Security 18), 2018, pp. 903–918.

[9] B. L. Bullough, A. K. Yanchenko, C. L. Smith, J. R. Zipkin, Predicting exploitation of disclosed software vulnerabilities using open-source data, in: Proceedings of the 3rd ACM on International Workshop on Security and Privacy Analytics, 2017, pp. 45–53.

[10] M. Almukaynizi, E. Nunes, K. Dharaiya, M. Senguttuvan, J. Shakarian, P. Shakarian,

Proactive identification of exploits in the wild through vulnerability mentions online, in: 2017 International Conference on Cyber Conflict (CyCon US), IEEE, 2017, pp. 82–88.

[11] Y. Fang, Y. Liu, C. Huang, L. Liu, Fastembed: Predicting vulnerability exploitation possibility based on ensemble machine learning algorithm, Plos one 15 (2020) e0228439.

[12] A. Okutan, M. Mirakhorli, Predicting the severity and exploitability of vulnerability reports using convolutional neural nets, in: 2022 IEEE/ACM 3rd International Workshop on Engineering and Cybersecurity of Critical Systems (EnCyCriS), IEEE, 2022, pp. 1–8.

[13] J. Yin, M. Tang, J. Cao, H. Wang, Apply transfer learning to cybersecurity: Predicting exploitability of vulnerabilities by description, Knowledge-Based Systems 210 (2020) 106529.

[14] J. Yin, M. Tang, J. Cao, H. Wang, M. You, Y. Lin, Vulnerability exploitation time prediction: an integrated framework for dynamic imbalanced learning, World Wide Web 25 (2022) 401–423.

[15] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, Journal of machine Learning research 3 (2003) 993–1022.

[16] M. Röder, A. Both, A. Hinneburg, Exploring the space of topic coherence measures, in: Proceedings of the eighth ACM international conference on Web search and data mining, 2015, pp. 399–408.

[17] J. Chang, S. Gerrish, C. Wang, J. Boyd-Graber, D. Blei, Reading tea leaves: How humans interpret topic models, Advances in neural information processing systems 22 (2009).

[18] T. Kojm, Clamav, 2004.