U.P.B. Sci. Bull., Series C, Vol. 84, Iss. 2, 2022 ISSN 2286-3540

EXTRACTING EXPLOITS AND ATTACK VECTORS FROM CYBERSECURITY NEWS USING NLP

Cristian SANDESCU[1], Alexandra DINISOR[2], Cristina-Veronica VLADESCU[3], Octavian GRIGORESCU[4], Dragos CORLATESCU[5], Mihai DASCALU[6], Razvan RUGHINIS[7]

Cybersecurity has an immense impact on society as it enables the digital protection of individuals and enterprises against an increasing number of online threats. Moreover, the rate at which attackers discover and exploit critical vulnerabilities outperforms the vendors' capabilities to respond accordingly and provide security patches. As such, open-source intelligence data (OSINT) has become a valuable resource, from which details on zero-day vulnerabilities can be retrieved and timely actions can be taken before the patches become available. In this paper we propose a method to automatically label articles on vulnerabilities and cyberattacks from trusted sources. Using Named Entity Recognition, we extract essential information about new vulnerabilities, such as the exploit's public release and the environment in which the attack's exploitation is possible. Our balanced dataset contains 1095 samples out of which 250 entries are from cybersecurity articles; the rest of the articles were crawled and annotated from the U.S. Government's Vulnerability Database, whereas automated text augmentation techniques were also considered. Our model built on top of spaCy obtained an overall performance of 75% recall on the Exploit Available task. When considering the Attack Vector metric, the model achieved the following recalls: Network 72%, Local 78%, and Physical 92%.

Keywords: Zero-days Attack, Exploit, Attack Vector, Entity Labeling, spaCy

[1] PhD student, Dept. of Software Development, CODA Intelligence SRL, Romania, e-mail: cristian.sandescu@codaintelligence.com
[2] MSc student, Dept. of Informatics, Technical University of Munich, Germany, e-mail: alexandra.dinisor@tum.de
[3] MSc student, Dept. of Computer Science, University POLITEHNICA of Bucharest, Romania, e-mail: cristiana.vladescu@stud.acs.upb.ro
[4] PhD student, Software Development, CODA Intelligence SRL, Romania, e-mail, e-mail: octavian.grigorescu@codaintelligence.com
[5] PhD student, Dept. of Computer Science, University POLITEHNICA of Bucharest, Romania, e-mail: dragos.corlatescu@upb.ro
[6] Prof, Dept. of Computer Science, University POLITEHNICA of Bucharest, Romania, e-mail: mihai.dascalu@upb.ro
[7] Prof, Dept. of Computer Science, University POLITEHNICA of Bucharest, Romania, e-mail: razvan.rughinis@upb.ro

1. Introduction

In recent years, the accelerated digitalization process resulted in an intensifying number of vulnerabilities and cyber-attacks. A fitting example is the 67 percent increase in security breaches over a range of five years [1], whereas most attacks are people-based and lead to information leakage. Moreover, Talalaev [2] highlighted that 73 percent of cybercriminals consider traditional antivirus security to be irrelevant when distributing their Trojans or hijacking internet connected devices.

In terms of preventing data breaches, early notifications on urgent patches, as well as information to mitigate the risk of exposure, should be a top priority. However, the average delay in applying patches by employees in a company is 102 days [3]. This alarming protection gap is a result of misinformation and lack of concern on the abundant patches and updates released by the software vendors. Nevertheless, the ever-changing threat landscape greatly impacts users trying to protect themselves against attackers.

Discovered by security researchers or bug bounty hunters, vulnerabilities are commonly published and discussed on the Internet. Therefore, important articles and in-depth reports are written and issued on niche magazines by cybersecurity analysts and reporters, thus providing support until vendors release corresponding security patches.

Our research aims to aggregate data from trusted cybersecurity news platforms (i.e., a central OSINT source) and to automatically extract information regarding vulnerabilities exploited in the wild as zero-days. . The dataset used for this project has 1000 manually labeled cybersecurity and can be downloaded from here [8]. Moreover, the enlarged sequence labeled dataset with the EVE feature can be downloaded from here.[9]

We summarize our core contributions as follows:

- Employing a dataset focused on two particular vulnerability characteristics, particularly the exploit’s public release and the attack vector (i.e., the attack’s operating environment), which were manually labeled and converted to IOB format.
- Expanding the manually collected dataset with samples from the National Vulnerability Database of the U.S. government and applying pre-trained contextual embedding with BERT for data augmentation to prevent overfitting behavior.
- Building a Named Entity Recognition model from spaCy v3 in conjunction with the afore-mentioned custom named entities to extract the key

[8] https://yggdrasil.codaintelligence.com/dataset.xlsx
[9] https://yggdrasil.codaintelligence.com/dataset-spacy.xlsx

characteristics based on sequence labeling.

- Obtaining qualitative outcomes in the field of zero-day attacks, such as the Attack Vector vulnerability metric with recall over 90% and the detection of proof-of-concepts for early exploits with recall over 75%.

The next section describes the current state of the fields of cybersecurity with the focus on early detection of vulnerabilities, followed by Natural Language Processing with data augmentation and sequence labeling. The third section presents the method in which the datasets used in our experiments are presented, as well as the applied algorithms. The results are analyzed in the fourth section, whereas the fifth presents conclusions and future experiments, together with envisioned improvements.

2. State of the Art

This section introduces state-of-the-art methods for the early detection of vulnerabilities, followed by relevant Natural Language Processing (NLP) techniques, namely data augmentation to enhance our dataset, and named entity recognition for labeling text segments.

2.1 Early Detection of Vulnerabilities

Detection techniques for software vulnerabilities are widely discussed in the scientific literature. For example, Mittal et al. [4] and Queiroz et al. [5] correlated user-generated data from social media posts with the identification of new vulnerabilities. Social networks represent a valuable OSINT source, where details about software or hardware-related vulnerabilities are discussed even before being officially disclosed by the vendors.

Moholth et al. [6] used Reactive Programming as a method to extract vulnerability-relevant tweets. Their experiment relied on filtering data with human involvement to ensure that the conclusions were correct. Moreover, a certain set of keywords was considered as not all tweets described a recent issue actively exploited in the wild. New critical flaws (i.e., zero-days vulnerabilities) were identified using a large number of retweets within a short time interval.

Another approach on identifying vulnerabilities that are likely to be exploited was presented by Tavabi et al. [7] who considered a neural language model whose input texts were extracted from the dark web and deep web, as well as security blog posts. The examination of dark web posts separated the available information relevant to the cybersecurity field from posts on illegal activities, such as the drug trafficking and the resale of stolen merchandise. The experiment included a Skip-Gram model with Negative Sampling. A further classification task for exploit prediction was carried out with a SVM classifier with Radial Basis Function kernel, which produced the best results for detecting the high-severity flaws, with an F1 score of .80.

Collecting Twitter posts for vulnerability detection is a widely considered perspective. As there can be millions of tweets describing past and recent cyber-attacks, Trabelsi et al. [8] proposed a machine learning component named SMASH (Social Media Analysis for Security on HANA) to be integrated in a security management platform. The two mentioned tasks focus on detecting zero-day exploits in tweet posts, as well as on separating new CVE assignations from CVE updates. Irrelevant terms were removed, and bag-of-words representations were considered when applying a clustering algorithm. In their experiment, Linux kernel vulnerabilities about zero-day exploits were assessed. The correlation between their findings and the delay in CVE Update detection argues that software developers and companies are in general one step behind vulnerability discovery, thus exposing their products to malicious actors.

2.2. Natural Language Specific Tasks

Data augmentation in NLP is a more sensitive task in comparison to the same strategy applied in other machine learning subfields, such as computer vision. Given precise disciplines like biology or cybersecurity, a single word might affect the meaning of the entire phrase. As a result, careful testing of the text augmentation techniques [9], such as synonym replacement, random swap, random insertion, and random deletion, is recommended.

Using pre-trained embeddings when augmenting text is a widely employed approach to preserve the context of the input sentence. Contextual embeddings prove to be reliable sources of preventing language models from overfitting. The Bidirectional Encoder Representations from Transformers (BERT) [10] provided state-of-the-art results for a variety of NLP tasks, while outperforming classic sequential models. BERT was trained on a large plain text corpus (about 3300 million words) from two popular sources: English Wikipedia and BookCorpus [11]. BERT as a contextual word embedding augmenter was applied on datasets from a variety of scientific fields [12]. Moreover, the cased model keeps the text’s original accents and marks, allowing a Named Entity Recognition [13] model to benefit during training. Other BERT-based experiments, such as fine-tuning BERT in the result of a conditional version for text augmentation [14], were conducted with promising outcomes.

Named Entity Recognition (NER). NLP techniques were frequently used to identify sequences that reference a specific entity. Bidirectional long-short term memory with conditional random field layer on top (BiLSTM-CRF) proved to be beneficial in a variety of real-life situations, including geoscience entity detection [15] or social media message recognition [16]. N-grams and Convolutional Neural Networks were also used in experiments from the biomedical field [17].

The most frequently used format in terms of labeling data for Named Entity Recognition tasks is the IOB scheme in which chunks of texts are labelled as: beginning (B-tag), center (i.e., I-tag inside), and the outside component (O-tag). In various experiments from the cybersecurity domain, Conditional Random Field classifiers [18] benefitted from this approach. Another successful labeling strategy consists of annotating data with customized relational labels based on their importance in a cyber-attack, which aids in sentence classification and malware-related token prediction [19].

Other approaches for detecting typical tokens in a text, such as the name of a person or a city, consider pre-trained models from spaCy[10]. SpaCy [20] is an open-source Python library that supports various NLP tasks. SpaCy is known for its processing speed in parsing large-scale data and it offers pretrained language models for more than 15 languages. SpaCy implements a state-of-the-art architecture for NER based on Bloom embeddings [21] and residual CNNs [22].

‘Tok2vec’ is an independent layer from spaCy that can be shared between components. It is frequently applied in conjunction with another component, such as ‘ner’, and is set as the first layer to generate suitable dynamic vectors. The built-in spaCy components ‘tagger’ and ‘parser’ do not share their features with the ‘ner’ component. As a result, they are often disabled during training for custom NER models. The named entity recognition system includes several NLP Transformer models, out of which roBERTa [23], an optimized BERT implementation, was considered in our experiments as part of the spaCy version 3 NER pipeline.

3. Method

Our overarching aim is to detect software and hardware vulnerabilities from OSINT news. As such, our method extracts specific relevant references to zero-day attacks called EVEs (Early Vulnerability Exposures). First, we required a dataset on exploitable vulnerabilities and cyber-attacks. Starting from an initial corpus of articles introducing vulnerabilities [24], we introduce two new datasets created for the task at hand, namely the EVE and NVD-CVE corpora. Afterwards, text augmentation techniques were applied to extend our datasets, followed by machine learning models trained on the security-relevant data.

[10] https://spacy.io/

3.1 Initial Corpus of Articles Introducing Vulnerabilities

Cybersecurity reporters from trusted websites provide useful information for our task. However, articles need to be filtered out since part of them consider events outside our scope of vulnerability identification – for example, bug bounty programs, hacked government websites, or data breaches for certain systems without mentioning the actual exploited vulnerability.

A dataset containing 1000 manually labeled cybersecurity articles was gathered by Iorga et al. [24] with the aim of performing text classification on whether an article introduces a new vulnerability or not. The articles were extracted from four cybersecurity news platforms, namely: The HackerNews[11], Ars Technica[12], Security Affairs[13], Threatpost[14]. The selected article provided insights into the most recent critical vulnerabilities or easy-to-exploit injection flaws.

The scraping of the articles was performed using Newspaper [25]. Additional features were considered to ensure a rigorous text analysis and a precise labeling. The following annotations were also collected, if mentioned in text: the CVE-ID [26] (i.e., the identifier corresponding to a Common Vulnerability and Exposure), its CVSS (i.e., Common Vulnerability Scoring System) score [27], affected product and version, patched version, or mentioned related products. The final version of this dataset consisted of 596 security-relevant articles and 404 security irrelevant articles.

3.2 EVE Corpus

The time lag between crucial moments, such as vulnerability discovery, CVE assignation, patch release, and CVE publish date, is of utmost importance. Even though relevant articles were filtered from the security irrelevant ones, a further division was necessary. As such, we introduce the EVE (Early Vulnerability Exposure) corpus which focuses on zero-day exploits, i.e., cyberattacks with and without patched versions available from the affected vendors, with no published CVE. Out of the 596 relevant articles, 250 EVE articles were manually selected and annotated by one cybersecurity expert.

Two new annotations were considered for each article: Exploit Available and Attack Vector. The annotation of examples focused on labeling only the most relevant and shortest sequences; thus, the chunk's first token had to be in strong relation to the annotated metric.

First, the Exploit Available annotation consist of labeling text spans where proof-of-concepts (PoC – e.g., video demonstration, or GitHub exploit code) were mentioned. References to security blog posts containing details about the unpatched flaw or detailed explanations about working PoC exploits were categorized as

[11] https://thehackernews.com/
[12] https://arstechnica.com/
[13] https://securityaffairs.co/wordpress/
[14] https://threatpost.com/

relevant for this feature. In some cases, a comprehensive description of the exploit technique was provided by the security writer of the article. The annotated articles with Exploit Available represent 22.4% of the EVE articles (see Table 1 for training set samples).

Table 1

Annotated samples for Exploit Available Task

Annotated sample (Exploit available text span)	Exploit available
"… an anonymous hacker with an online alias "SandboxEscaper" today released proof-of-concept (PoC) exploit code for a new zero-day vulnerability…"	YES
"… a security researcher today publicly disclosed details and proof-of-concept exploits for two 'unpatched' zero-day vulnerabilities …"	YES
"Researchers illustrated and demonstrated four attack scenarios, as explained below …"	YES
"Researchers have no plans to release the proof-of-concept code for these attacks …."	NO

Second, the annotation for Attack Vector consists of four different classes, together with the corresponding relevant text spans. This annotation relates to the exploitability metric from the CVSS v3 score. This vulnerability metric highlights the context in which the vulnerability was already exploited in the wild or potentially exploited. The four categories are:

- 'NETWORK': the article mentions that the attacker had full remote control over the system or device. The vulnerability was remotely exploitable also in cases of flaws linked to a browser component, such as plugins, browser, extensions, or add-ons.
- 'ADJACENT': the attacker was connected to the same network.
- 'PHYSICAL': the article explicitly mentions that the attacker required physical access to the targeted machine. Part of attacks included the use of peripheral devices, such as USB drives.
- 'LOCAL': the attacker accesses the targeted system locally by using scripts from the console or the keyboard.

Table 2 introduces examples for each Attack Vector type, alongside the EVE corpus summary statistics for the Attack Vector. Due to insufficient samples, the security articles describing 'Adjacent' attacks were not taken into consideration into the subsequent machine learning approaches.

Table 2

Annotated samples for Attack Vector NER Task

Attack Vector type	Annotated sample (Attack vector text span)	Total samples
Network	"… one of which could allow remote hackers to take complete control …"	202

Attack Vector type	Annotated sample (Attack vector text span)	Total samples
Physical	"Although exploiting the issue requires physical access, Sintonen explained …"	15
Local	"… could allow a local attacker to gain and run code with administrative system privileges on the targeted machines …"	6
Adjacent	"… could have allowed an attacker, connected to the same network as the victim … "	3

All NER experiments required the conversion of the original texts into inside–outside–beginning (IOB) tagged texts, as the entities had variable length. Entity tokens refer in this particular case to relevant sequences related to the chosen vulnerability features, namely Exploit Available and Attack Vector. The Attack Vector entities represent only 10% of the tokens, whereas the Exploit Available text spans cover only 3% of all the tokens from a news article, thus making the NER approaches very challenging.

As a result, further experiments considered a trimmed dataset containing only the sentences with labeled entities ('OnlyTagSent') and disregarding all other sentences from the news article.

3.3. NVD-CVE Corpus

Given the limitation of the EVE dataset that is neither large nor diverse enough, more annotations were collected from multiple sources related to cybersecurity attacks. As such, vulnerability data feeds from the National Vulnerability Database[15] of the U.S. government were used. It is worth mentioning that the exploitability metrics of each discovered vulnerability since 2002 are offered and published in a JSON format, having the advantage of already being correctly annotated. This additional dataset was highly imbalanced with 800 Network, 200 Local, and 150 Physical examples and was collected by the same expert as in the previous stage.

3.4. `Text augmentation on the NVD-CVE corpus

For the spaCy experiment described later on in detail, a text augmentation approach was chosen for the Local and Physical Attack Vector types, particularly for the training examples belonging to the NVD data feed with the aim of having a balanced dataset. These Attack Vector types have a reduced number of examples both in the EVE corpus and in the NVD data feed. A first attempt considered NLPAug[16] for synonym replacement. However, this approach damaged highly specific cyber-security terms (e.g., 'buffer overflow vulnerability' – 'cowcatcher overflow vulnerability', 'attacker with physical access' – 'assailant with forcible

[15] https://nvd.nist.gov/vuln/data-feeds
[16] https://github.com/makcedward/nlpaug

access') and was therefore abandoned.

A second experiment was conducted using TextAttack[17] [28], more specifically an Embedding Augmenter in Command-Line Interface to change a specific number of words per input. This approach encountered the same issue with the security vocabulary (e.g., 'Cross-site scripting (XSS) vulnerability' – 'Cross-site scripting (XSS) fragility').

The final and successful solution was a word-level augmentation with insertion. A pre-trained contextual embedding with BERT was used for a semantically suitable insertion in the original paragraphs (e.g., 'root login may allow upon a reboot' - 'unauthenticated root commit login may also allow upon a quick reboot'. The augmented contextual BERT texts implied the NVD-CVE Local and Physical examples were added to the training set. The goal after sampling was to obtain a balanced dataset (noted as *EVE +NVD-CVE +Augmented*), which in the end contained 395 Network, 400 Local, and 300 Physical examples.

The evaluation metrics of Precision, Recall and F1-score are presented for all experiments. We emphasize the importance of Recall, as our focus is to maximize the number of correctly detected sequences that represent a vulnerability. Our overarching goal is to build a framework that can be used in the decision-making process of a cybersecurity expert.

3.5 Sequence Labeling / BiLTSM?

Bidirectional word-level Long-Short Term Memory (BiLSTM) networks were built with Tensorflow and Keras instead of a standard LSTM, because these networks consider the previous and post information after a specific sequence. The input vocabulary required a tokenizer for words and their corresponding tags (IOB scheme). The model architecture implied an embeddings layer, a BiLSTM layer with a dropout rate of 0.2, and a time distributed layer applied on Dense layers with ReLU activation corresponding to each IOB tag (see Fig. 1 for architecture). No pre-trained embeddings were chosen due to the small-scaled preliminary dataset.

In terms of hyperparameters, the configuration implied an Adam optimizer with a learning rate of 0.005, the model was trained for 20 epochs with a batch size of 32. The performance on the training data was measured with categorical cross-entropy as a loss function.

[17] https://github.com/QData/TextAttack

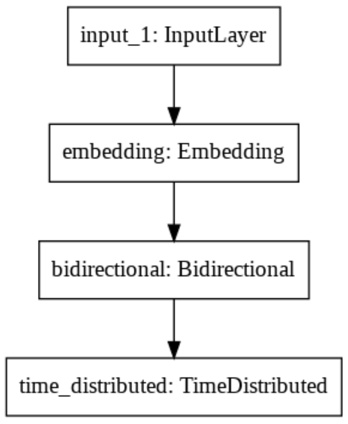
input_1: InputLayer
embedding: Embedding
bidirectional: Bidirectional
time_distributed: TimeDistributed

Fig. 1. BiLSTM network architecture.

3.6 spaCy

Our second approach for the NER tasks considered a custom NER model built using spaCy v3. The spaCy's underlying model architecture is not detailed publicly, but it considers an embedding strategy with subword features and Bloom Embeddings Input data was transposed into the required spaCy v3 training format with IOB tags, while dropping sentence and document indexes. Essentially, the Language class, the Vocab, and the Doc object are the three most important data structures in spaCy. The Language class is responsible for parsing text and converting it into a Doc object, being stored as an 'nlp' variable. The Doc object owns the token sequence along with all the annotations. With the aim of guaranteeing a unique truth source, the Vocab object unifies strings, word vectors, and lexical properties.

Due to the fact that spaCy 'ner' component inside the 'nlp' pipe provides default entities related to persons, organizations, time expressions or monetary values, we employ a custom Named Entity Recognition model with custom entity types for each EVE feature. the.Therefore, the specific new entity labels were added before setting up the pipeline and the entity recognizer. The model started from an English spaCy trained pipeline optimized for CPU, updated in spaCy version 3 and the custom entity labels were added to the entity recognizer. The initial NLP spaCy processing pipeline contained additional pipeline components, such as "tok2vec" or "tagger", that did not affect the NER component; as such, they were disabled during the training process. No static values were used in the spaCy training configuration. In terms of hyperparameters, the NER model was trained for 20 epochs on shuffled data to reduce the bias generated by the order of the training examples with compounding batch sizes from 4 to 32, a spaCy optimizer, and a dropout rate of 0.2.

4. Results

4.1 Bi-LSTM

The first experiments with the BiLSTM network exhibited problems in identifying the text spans which describe the EVE tasks in cybersecurity articles: references to a published proof-of-concept exploit video and to an attack vector metric. For the machine learning approaches, the datasets were divided statically into two subsets: train and test data with a split percentage of 80% and 20%, which were later used as a benchmark for the spaCy approach.

The model's performance in the Exploit Available task was evaluated on the entire corpus. The classification report from Table 3 indicates that the model clearly overfits the data because the O value, outside-chunk entity, represents 99,6% out of the total entities. It was interpreted as a clear overfitting behavior, because most of the sentences were not relevant for our task

Table 3

Classification performance on the entire Exploit corpus

Named entity	Precision	Recall	F1 Score
B-exploitAvailable	.00	.00	.00
I-exploitAvailable	.00	.00	.00
O	1.00	1.00	1.00

The cybersecurity articles were excessively long, and the named entities were sparse. As a result, due to the poor performance of the first trial, experiments continued with shortened versions of the EVE features. The Bi-LSTM model was further trained on the corresponding Exploit Available dataset type, 'OnlyTagSent' dataset. Considering that this corpus contained 15.5% named entity tokens, the model showed a high overfitting condition after 20 epochs (see the classification report in Table 4). The model only extracted outside chunk entities (i.e., 'O') in the same way it did in the previous trial.

Table 4

Classification performance for Exploit Available 'OnlyTagSent' with Bi-LSTM

Named entity	Precision	Recall	F1 Score
B-exploitAvailable	.00	.00	.00
I-exploitAvailable	.00	.00	.00
O	.93	1.00	.96

Next, the experiments for determining the Attack Vector features were conducted only on 'OnlyTagSent' corpus, given the weak performance of the Bi-LSTM approach on identifying the Exploit Available labels; however, the Bi-LSTM model exhibited overfitting as presented in the classification report from Table 5.

Table 5

Classification performance for Attack Vector 'OnlyTagSent' with Bi-LSTM

Named entity	Precision	Recall	F1 Score
B-Network	.01	.02	.01
I-Network	.10	.01	.03
B-Local	.00	.00	.00
I-Local	.00	.00	.00
B-Physical	.00	.00	.00
I-Physical	.00	.00	.00
O	.97	.99	.98

Considering the presented cases, this approach was abandoned in favour of the state-of-the-art Named Entity Recognition tool, spaCy. Further alteration and improvements of the cybersecurity datasets were specifically conducted for this machine learning approach.

4.2 spaCy

Experiments were conducted with variations of both datasets (i.e., trimmed versions or diverse augmentations); however, the most promising results with the spaCy version 3 model are represented by the trimmed version ('OnlyTagSent') for the Exploit Available feature and the augmented corpus for Attack Vector. The evaluation was conducted using the spaCy scorer.

Table 6 introduces the results with our spaCy model obtained when testing with new article sentences from the 'OnlyTagSent' EVE dataset. No NVD-CVE data was added since the exploit's release is not related to the CVSS score and thus could not be extracted from the National Vulnerability Database.

Table 6

spaCy evaluation on the EVE corpus

Named entity	Precision	Recall	F1 Score
B-exploitAvailable	.642	.750	.692
I-exploitAvailable	.625	.409	.495

Table 7 introduces the results of our spaCy model on both the EVE+NVD-CVE corpus), followed by the performance of the balanced model with augmentation (see Table 8). The poor performance on the first dataset is argued by the high imbalance between the classes.

Table 7

spaCy evaluation on EVE+NVD-CVE corpus

Named entity	Precision	Recall	F1 Score
B-Network	.224	.083	.122
I-Network	.059	.125	.206
B-Local	.750	.333	.461
I-Local	.760	.308	.439
B-Physical	.571	.461	.510
I-Physical	.896	.594	.715

Table 8

spaCy evaluation on the EVE +NVD-CVE +Augmented corpus

Named entity	Precision	Recall	F1 Score
B-Network	.670	.724	.696
I-Network	.718	.658	.686
B-Local	.537	.773	.634
I-Local	.641	.784	.705
B-Physical	.918	.730	.813
I-Physical	.881	.928	.904

4.3 Sample Use Cases

When tested on new articles from ZDNet[18] outside the EVE corpus, the spaCy model identified quite well the Exploit Available entities. Table 9 highlights three samples, the latter having no 'ExploitAvailable' named entity and being correctly identified as such.

Table 9

spaCy model evaluation on new entries

Test samples
"'Here's How the Attack Works? The attack involves exploitation of three vulnerabilities via iTunes and the App Store's iOS Notify function."
"'Nelson later released proof-of concept code for the first Steam zero-day, and also criticized Valve and HackerOne for their abysmall handling of his bug report."
"Security researchers and regular Steam users alike are mad because Valve refused to acknowledge the reported issue as a security flaw, and declined to patch it."

Tests were performed on newly gathered paragraphs from security articles recently. The test samples annotated by spaCy are presented in Table 10. The Network test samples indicate that additional tokens were erroneously tagged, as the recall for the beginning chunk is approximately .72. Both test samples for Physical denote adequate identifications, with a minor incorrect labeling with the Local category in the first examples. The Local class has the best overall recall and correctly labels multiple sequences in the last example.

5. Conclusions and future work

This paper introduces a method to automatically label articles on vulnerabilities and cyberattacks from trusted sources, while focusing on the exploit's public release and the attack vector.

Experiments with two different approaches were considered. The bidirectional word-level LSTM model showed poor results for both tasks, whereas our custom model built using spaCy with Bloom embeddings [21] and residual CNNs [22] achieved promising results. Besides the EVE corpus with annotated

[18] https://www.zdnet.com/

cybersecurity articles, the NVD data feeds with the CVEs from 2017 to 2021 were added into the final corpus. Text augmentation techniques with contextualized word embeddings from BERT were employed for two of the three Attack Vector classes (i.e., Physical and Local) to ensure a balanced dataset.

As a future development, cybersecurity information should be crawled and annotated from other sources such as exploit kits sold on the Dark Web, blogs of cybersecurity experts, or the Google Hacking Database. Additional references for other exploitability metrics from the CVSS v3 score should be retrieved using NER models - for instance, the User Interaction (e.g., "requires user interaction", "victim needs to open the malicious iWork file") and Privileges Required (e.g., "attacker requires no privileges", "non-privileged user can initiate"). In terms of follow-up processing, we envision creating a dashboard with a newsfeed in which articles can be filtered depending on user needs and on the owned infrastructure.

Table 10

Samples and spaCy output

Type	Test sample
Network	"Lastly, an extension named Rainbow Fart was ascertained to have a zip slip vulnerability, which allows an adversary to overwrite arbitrary files on a victim's machine and gain remote code execution."
Network	"The disclosure of the CODESYS flaws comes close on the heels of similar issues that were addressed in Siemens SIMATIC S7-1200 and S7-1500 PLCs that could be exploited by attackers to remotely gain access to protected areas of the memory and achieve unrestricted and undetected code execution."
Network	"The vulnerability exists because the software lacks proper authentication controls to information accessible from the web UI. An attacker could exploit this vulnerability by sending a malicious HTTP request to the web UI of an affected device."
and	"The USB Mass Storage in Microsoft Windows Vista SP2, Windows Server 2008 SP2 and R2 SP1, Windows 7 SP1, Windows 8.1, Windows Server 2012 Gold and R2, Windows RT 8.1, and Windows 10 Gold and 1511 allows to execute arbitrary code by , aka "USB Mass Storage Elevation of Privilege Vulnerability."
	"Setup Wizard in Android 5.1.x before LMY49H and 6.x before 2016-03-01 allows to bypass the Factory Reset Protection protection mechanism and delete data via unspecified vectors, aka internal bug 25955042."
	" Avamar Data Store (ADS) and Avamar Virtual Edition (AVE) in EMC Avamar Server before 7.3.0-233 allow to obtain root privileges by leveraging admin access and entering a sudo command."
	"An elevation of privilege vulnerability in Mediaserver in Android 4.x before 4.4.4, 5.0.x before 5.0.2, 5.1.x before 5.1.1, 6.x before 2016-11-01, and 7.0 before 2016-11-01 could enable a to execute arbitrary code within the context of a . This issue is rated as High because it could be used to gain to elevated capabilities, which are not normally accessible to a third-party application."

Acknowledgment

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS – UEFISCDI, project number 2PTE⁄2020, YGGDRASIL – "Automated System for Early Detection of Cyber Security Vulnerabilities" and by the internal UPB program Proof of Concept.

R E F E R E N C E S

[1]. *K. Bissel, R. M. Lasalle and P. D. CIN*, "Ninth Annual Cost of Cybercrime Study", 2019.

[2]. *A. Talalaev*, "Website Hacking Statistics You Should Know in 2021", https://patchstack.com/, 2021.

[3]. *Barkly*, "Study Reveals 64% of Organizations Experienced Successful Endpoint Attack in 2018", https://www.businesswire.com/, 2018.

[4]. *S. Mittal, P. K. Das, V. Mulwad, A. Joshi and T. Finin*, "Cybertwitter: Using twitter to generate alerts for cybersecurity threats and vulnerabilities", in proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), San Francisco, CA, USA, IEEE, pp. 860–867, 2016.

[5]. *A. L. Queiroz, S. Mckeever and B. Keegan*, "Eavesdropping hackers: Detecting software vulnerability communication on social media using text mining", in proceedings of the The Fourth International Conference on Cyber-Technologies and Cyber-Systems, pp. 41-48, 2019.

[6]. *O. C. Moholth, R. Juric and K. M. McClenaghan*, "Detecting cyber security vulnerabilities through reactive programming", in proceedings of the Proceedings of the 52nd Hawaii International Conference on System Sciences, Grand Wailea, Maui, Hawaii, USA, ScholarSpace, pp. 1–10, 2019.

[7]. *N. Tavabi, P. Goyal, M. Almukaynizi, P. Shakarian and K. Lerman*, "Darkembed: Exploit prediction with neural language models", in proceedings of the Conference on Artificial Intelligence, New Orleans, Louisiana, USA, AAAI Press, pp. 7489–7854, 2018.

[8]. *S. Trabelsi, H. Plate, A. Abida, M. M. B. Aoun, A. Zouaoui, C. Missaoui, S. Gharbi and A. Ayari*, "Monitoring software vulnerabilities through social networks analysis", in proceedings of the 2015 12th International Joint Conference on e-Business and Telecommunications (ICETE), Colmar, France, IEEE, pp. 236–242, 2015.

[9]. *J. Wei, K. Zou*, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks", in proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference Natural Language Processing, Hong Kong, China, Association for Computational Linguistics, pp. 6381–6387, 2019.

[10]. *J. Devlin, M.-W. Chang, K. Lee and K. Toutanova*, "BERT: Pre-training of deep bidirectional transformers for language understanding", in proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, Association for Computational Linguistics, pp. 4171–4186, 2019.

[11]. *Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba and S. Fidler*, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books", in proceedings of the Proceedings of the IEEE international conference on computer vision, Santiago, Chile, IEEE Computer Society, pp. 19–27, 2015.

[12]. *V. Atliha, D. Šešok*, "Text augmentation using BERT for image captioning", in Applied Sciences, **vol. 10**, no. 17, 2020, pp. 5978.

[13]. *G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami and C. Dyer*, "Neural architectures for named entity recognition", in proceedings of the The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, The Association for Computational Linguistics, pp. 260–270, 2016.

[14]. *X. Wu, S. Lv, L. Zang, J. Han and S. Hu*, "Conditional bert contextual augmentation", in proceedings of the International Conference on Computational Science, Faro, Portugal, Springer, pp. 84–95, 2019.

[15]. *Q. Qiu, Z. Xie, L. Wu, L. Tao and W. Li*, "BiLSTM-CRF for geological named entity recognition from the geoscience literature", in Earth Science Informatics, **vol. 12**, no. 4, 2019, pp. 565–579.

[16]. *B. Y. Lin, F. F. Xu, Z. Luo and K. Zhu*, "Multi-channel bilstm-crf model for emerging named entity recognition in social media", in proceedings of the Proceedings of the 3rd Workshop on Noisy User-generated Text, Copenhagen, Denmark, Association for Computational Linguistics, pp. 160–165, 2017.

[17]. *Q. Zhu, X. Li, A. Conesa and C. Pereira*, "GRAM-CNN: a deep learning approach with local context for named entity recognition in biomedical text", in Bioinformatics, **vol. 34**, no. 9, 2018, pp. 1547–1554.

[18]. *A. Sirotina, N. Loukachevitch*, "Named entity recognition in information security domain for russian", in proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019), Varna, Bulgaria, ACL, pp. 1114-1120, 2019.

[19]. *P. Phandi, A. Silva and W. Lu*, "SemEval-2018 task 8: Semantic extraction from CybersecUrity REports using natural language processing (SecureNLP)", in proceedings of the 12th International Workshop on Semantic Evaluation ((SemEval-2018)), New Orleans, Louisiana, USA, ACL, pp. 697–706, 2018.

[20]. *M. Honnibal, I. Montani*, "spacy 2: Natural language understanding with bloom embeddings", in convolutional neural networks and incremental parsing, **vol. 7**, no. 1, 2017, pp.

[21]. *J. Serrà, A. Karatzoglou*, "Getting deep recommenders fit: Bloom embeddings for sparse binary input/output networks", in proceedings of the Proceedings of the Eleventh ACM Conference on Recommender Systems, Como, Italy, ACM, pp. 279–287, 2017.

[22]. *K. Zhang, W. Zuo, Y. Chen, D. Meng and L. Zhang*, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising", in IEEE transactions on image processing, **vol. 26**, no. 7, 2017, pp. 3142–3155.

[23]. *Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov*, "Roberta: A robustly optimized bert pretraining approach", in CoRR, **vol. abs/1907.11692**, 2019, pp.

[24]. *D. Iorga, D. Corlătescu, O. Grigorescu, C. Săndescu, M. Dascălu and R. Rughiniş*, "Early Detection of Vulnerabilities from News Websites using Machine Learning Models", in proceedings of the 2020 19th RoEduNet Conference: Networking in Education and Research (RoEduNet), Online, IEEE, pp. 1–6, 2020.

[25]. *L. Ou-Yang*, "Newspaper3k: Article scraping & curation", 2021.

[26]. *The MITRE Corporation*, "CVE", https://cve.mitre.org/, 2021.

[27]. *National Institute of Standards and Technology*, "National Vulnerability Database", 2021.

[28]. *J. X. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin and Y. Qi*, "Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp", in proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online, Association for Computational Linguistics, pp. 119–126, 2020.