2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

RelExt: Relation Extraction using Deep Learning approaches for Cybersecurity Knowledge Graph Improvement

Aditya Pingle, Aritran Piplai, Sudip Mittal, Anupam Joshi
University of Maryland, Baltimore County, Baltimore, MD 21250, USA
{adiping1, apiplai1, smittal1, joshi}@umbc.edu

James Holt, and Richard Zak
Laboratory for Physical Sciences
{holt, rzak}@lps.umd.edu

***Abstract*—Security Analysts that work in a 'Security Operations Center' (SoC) play a major role in ensuring the security of the organization. The amount of background knowledge they have about the evolving and new attacks makes a significant difference in their ability to detect attacks. Open source threat intelligence sources, like text descriptions about cyber-attacks, can be stored in a structured fashion in a cybersecurity knowledge graph. A cybersecurity knowledge graph can be paramount in aiding a security analyst to detect cyber threats because it stores a vast range of cyber threat information in the form of semantic triples which can be queried. A semantic triple contains two cybersecurity entities with a relationship between them. In this work, we propose a system to create semantic triples over cybersecurity text, using deep learning approaches to extract possible relationships. We use the set of semantic triples generated through our system to assert in a cybersecurity knowledge graph. Security Analysts can retrieve this data from the knowledge graph, and use this information to form a decision about a cyber-attack.**

***Index Terms*—cybersecurity, deep learning, knowledge graphs**

I. Introduction

Cyber-attacks aim to compromise the confidentiality, integrity, and availability of information. These attacks target individuals, small and medium enterprises. Today attacks such as a denial of services, malicious codes executed via a backdoor or trapdoors, malware, trojans are some of the most common attack types [2]. Every year researchers and cyber defense professionals find millions of attack and malware variants in the wild [6].

Now a 'Security Analysts' working in a 'Security Operations Center' (SoC) needs to keep up with variants of cyber threat intelligence that's available online so as to secure their organization. These analysts need to process this information keeping in mind their local defensive setup.

Knowledge about cyber-attacks and malwares specifically their means, target, etc. is available in the 'wild' as cyber threat intelligence. It is possible to develop a cyber informatics pipeline which scours the web for threat intelligence, extracts and mines knowledge from these intelligence samples and represents them as a scheme fit for Security information and event management (SIEM) systems.

Threat intelligence sources are generally of two types, *covert* and *overt*. Overt or 'open source intelligence' (OSINT) is available through various sources like, cybersecurity blogs, cybersecurity reports, CVE [20], CWE [21], National Vulnerability Datasets [27], and any cybersecurity text publicly available. Various SIEM systems, fetch, process, and store much of the cyber threat intelligence available through OSINT sources. Knowledge Graphs (KG) specifically to store cyber threat intelligence has been used by many cybersecurity informatics systems. These cybersecurity knowledge graphs can not only store but are also be able to retrieve data and answer complex queries asked by the security analysts.

The dependence of various cybersecurity informatics systems on various knowledge representation schemes makes it imperative that we develop systems that improve these representations. Some examples are the Intrusion Detection System Knowledge Graphs proposed by Undercoffer et al. [33] and Unified Cybersecurity Ontology (UCO) [38]. In case of a cybersecurity knowledge graph, this boils down to improving the quality of semantic triples generated from various cyber threat intelligence sources. An ontology can be used to provide a system with base cybersecurity classes and the relations that exist between them.

Semantic triple generation is a key component in the Knowledge Graph population(See Section II). A semantic triple for cybersecurity comprises of a pair of cybersecurity entities and a relationship between the entities. A cybersecurity entity can be a word or a group of words extracted from cybersecurity text. Generally, a Named Entity Recognizer (NER) is used to extract entities that form the base knowledge graph. Multiple such cybersecurity domain specific named entity recognizer currently exist [4][16][18][25][15][32][28].

The next step in cybersecurity knowledge graph creation is, predicting the right relationship between two specific entities extracted from a specific cyber threat intelligence source. As soon as we figure out the various relationships and entities, we can assert the semantic triple set in a knowledge graph.

In this paper, we propose the *RelExt* system that strives

ASONAM '19, August 27-30, 2019, Vancouver, Canada

ACM ISBN 978-1-4503-6868-1/19/08…$15.00
https://doi.org/10.1145/3341161.3343519

to improve various cyber threat representation schemes, especially cybersecurity knowledge graphs (CKG) by predicting relations between cybersecurity entities identified by cybersecurity named entity recognizer. These representation systems have been used to build various 'Analyst Augmentation Systems' that aid a security analyst. These augmentation systems help keep an analyst working in an SoC environment updated about various developments in the wild that can impact the security of her organization. We believe that improving the base cyber threat intelligence representation will help improve the overall quality and performance of these analyst augmentation systems.

RelExt is a feed-forward neural network model that predicts relationships between cybersecurity entities, that form up a triple. Existing models use features such as length of entities involved, words between the entities as features for the model. We leverage the contextual similarity between the entities to find out if the entities would make up a triple. In RelExt, we use contextual vector representation of cybersecurity entities identified by a cybersecurity specific NER.

The rest of the paper has been organized as follows - Section II discusses some related work. Section III talks about building an ontology for our Cybersecurity Knowledge Graph. We introduce the RelExt system in Section IV. Section V talks about the evaluation metrics. We conclude the paper in section VI.

II. Related Work

A. Knowledge Graphs for cybersecurity

Data management and exploration were for long supported by either keyword-based search technology or various stochastic matching approaches [29]. Content and knowledge-based approach in contrast to the former approach, add a layer of sophistication. Often termed as the specification of a conceptualization, Ontology forms the exclusive way to represent and communicate facts and relations with multiple agents [12]. Ontology is a set of classes with attributes and relationships between instances of various classes. One of the earliest available ontology for the cybersecurity, which was built as an Intrusion Detection System (IDS) was introduced by Undercoffer et al. [33]. Unified Cybersecurity Ontology, an extension of the IDS ontology, built by Syed et al. [38] is a more cybersecurity domain focused ontology based on STIX 1.2 [10]. We in this paper, extend some of the concepts from UCO 1.0, like various classes and relationships. Yan Jia et al. [15] discuss knowledge graph population approaches for the cybersecurity domain. The authors use various machine learning approaches to extract entities of interest from the cybersecurity domain. Relationships are then predicted between these entities by calculating formulas and path ranking algorithms. CyberTwitter [23] and Cyber-All-Intel [24] are other systems that create a cybersecurity knowledge graph.

Knowledge graphs consist of semantic triples which have a subject, predicate, and an object. A predicate is a relationship between the subject and the object. The semantic triple generation would thus require a relationship extractor. Alternatively, an existing knowledge graph is improved by looking for more relationships, or links, between entities, nodes, already available in the knowledge graph. Our paper concentrates on developing *RelExt*, which uses a feed forward neural network to predict relationships between subject-object entity pairs and subsequently improve the knowledge graph population.

B. Relationship Extraction

Relationships connect two entities found to be interesting by the system. Relationships could be symmetrical or asymmetrical. Entities can also end up with no relation to each other, depending upon the underlying ontology. Relationship extractors always end up with a lack of sufficient data to be trained with. Defining the relation might seem to be a straightforward task, but applying a relationship for two entities is an ambiguous task. Often, a difference of opinions and perspective to what a relationship means, end up with higher inner-annotation disagreement [31]. Based on the approach, relation extraction can be classified as global level or mention level relation extraction [31]. Mention level extraction expects a pair of entities to find a relationship for. Whereas, global level extraction only expects the corpus and ends up listing entities with the relationship. Relationship extractors can be either binary or n-ary classifiers. Binary classifiers try to predict whether a specific relation R holds between two entities. Whereas n-ary classifiers try to predict whether two entities hold one of the relationships from the predefined set [26].

Relation extraction can be classified down to three approaches, supervised, semi-supervised, and unsupervised. Supervised approaches are explored by Kambhatla et al. [17] and Zhao et al. [39]. Some of the well known semi-supervised approaches include the DIPRE system introducted by Brin [5], Snowball [1], and Knowitall [7]. Hasegawa [14] proposed one of the earliest approach for unsupervised clustering for relationship extraction. Knowitall is extended well to incorporate an unsupervised approach for relationship extraction by Rosenfeld et al. [30].

C. Knowledge Graph Improvement

Relation prediction or extraction is one elegant way to fill in missing links between entities of interests in a knowledge graph. A simple link prediction would not suffice this task, as nodes in KG carry the certain identity and the edges mean a certain connection between the identities [37]. Moreover, we need to know if a relationship exists between pairs of entities along with the type of relationship. Relationships between two entities, or between pair of the group of entities, can be classified based on their complexity and number of entities mapped. TransE [3] model predicts one-to-one relationships [36] based on the vector space for head and tail entities. TransH model, an improvement over TransE [36], based over translating vectors on hyperplane [36] works on predicting many-to-many relationships. The TranSparse model uses sparse matrices to address the heterogenity and

imbalance of knowledge graph which was missed out by TransE, TransH, TransR, and TransD models [13]. Our model, RelExt, is trained over vectors, specifically in the domain of cybersecurity, to extract relationships between pairs of cybersecurity entities.

III. Cybersecurity Knowledge Graph

A use-case of our system is to aid in the development of a cybersecurity knowledge graph (CKG). A CKG generally has two main parts, a schema and various cybersecurity-related semantic triples. In order to build a CKG, the first thing we need to do is to define a schema. Next, we use our system to extract relationships between cybersecurity entities, which are words or groups of words, extracted from cybersecurity text. Once we have the entity-relationship set, we can populate the data consistent with the schema.

Syed et al. in their paper [38] have developed an ontology for cybersecurity called Unified Cybersecurity Ontology (UCO 1.0). Their paper describes a prototype version of an ontology which encompasses various cybersecurity elements. The paper provided us with a starting point for the development of our system, as it mentions some classes and probable relationships that could exist between various classes.

Members of our research group have worked previously on creating cybersecurity knowledge graph from different open source intelligence (OSINT) sources [23], [24]. In this paper, we create a system 'RelExt' that improves such knowledge graphs by predicting relations between various cybersecurity entities, as they occur in the text. RelExt (See Section IV) leverages a deep neural net to automatically extract various relationships between pairs of entities to improve our CKG.

Defining entities and relationships in a cybersecurity ontology are paramount for building a CKG. We use the class definitions proposed by UCO [38] to define our entities of interest. The class definitions and the relationship definition present in UCO 1.0 were based on Structured Threat Information eXpression (STIX) 1.2. We use the relationships and classes defined by the Structured Threat Information eXpression (STIX) 2.0 to update UCO 1.0 and create UCO 2.0. [9]. Figure 1, provides an overview of the relationships between various entity types as defined in STIX 2.0.

A. Classes in UCO 2.0

Here we explain some important classes that are present in UCO 2.0:

- *Software* : An entity that relates to a piece of code usually used as tool such as Office or Unix operating system.
- *Malware* : An entity that refers to malicious code and/or software which is inserted into a system.
- *Indicator* : An entity that contains a pattern which helps the administrator to indicate an ongoing attack or malicious activity.
- *Vulnerability* : An entity that refers to a patch of bug or weakness that could be exploited by ill-intended users.
- *Course-of-action* : An entity that refers an action or set of actions that either prevents or responds to an attack.

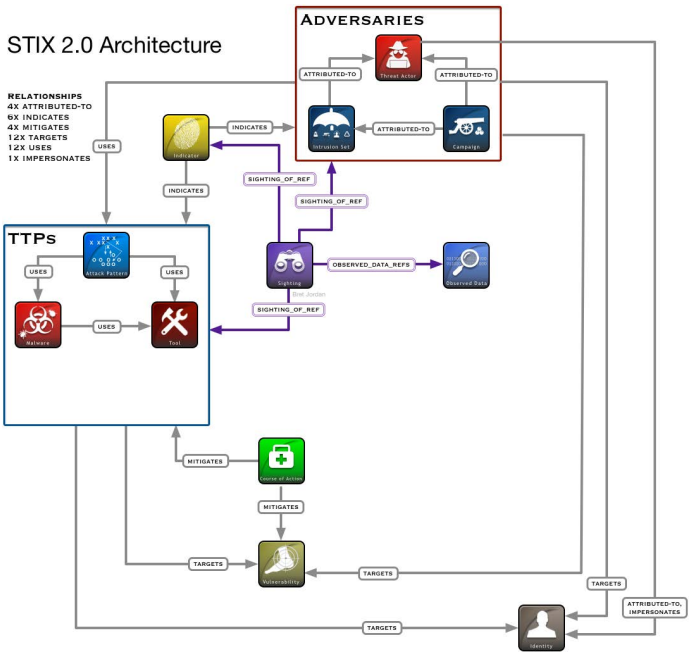

Fig. 1: Relationships and some classes defined in Structured Threat Information eXpression (STIX) 2.0 architecture [9].

- *Tool* : An entity that refers to a legitimate software that can be used by threat actors for malicious activities.
- *Attack-pattern* : An entity that refers to steps that could result in an active attack on an individual or group of users.
- *Campaign* : An entity that refers to grouping of activities that could lead to a malicious attack.

B. Relationships in UCO 2.0

In UCO 2.0 we have defined the following relationship, based on the STIX 2.0 definitions [9].

- *hasProduct* : Relationship where the subject and object entities, or just the object entity belong to software class.
- *hasVulnerability* : Relationship where the subject entity belongs to a software class and object entity belongs to a vulnerability class.
- *mitigates* : Relationship where the subject entity belongs to a course-of-action class and object entity belongs to malware or campaign class, wherein the subject entity aims to lessen the damages caused by object entity.
- *uses* : Relationship where the subject entity belongs to a campaign or malware class and object entity belongs to a tool or software class, wherein subject entity aims to leverage object entity to carry on an attack.
- *indicates* : Relationship where the subject entity belongs to a indicator class and object entity belongs to a malware or campaign class, wherein indicates. demonstrates presence or after effects of a object entity
- *attributed-to* : Relationship where the subject entity belongs to campaign or intrusion-set class and object entity belongs to threat actor class wherein subject entity is attributed to object entity.

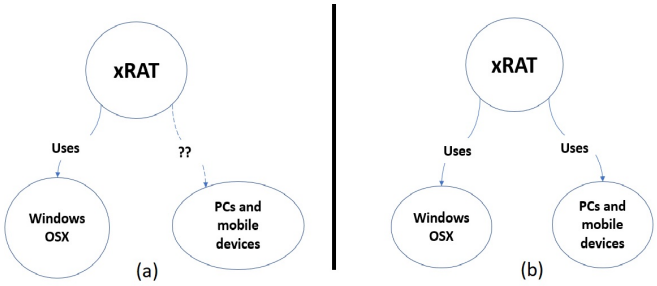

Fig. 2: Missing Relationships: The diagram on the left depicts how a particular relationship is not asserted in the existing knowledge graph. The one on the right shows, the relationship asserted by RelExt.

- *related-to* : Relationship wherein subject entity and object entity are related to each other.
- *located-at* : Relationship where entities are inter dependent or become a part of.

C. Baseline Knowledge Graph and missing relationships

In existing cybersecurity knowledge graph systems (See Section II-A) the resulting knowledge graph can be improved. A knowledge graph can have incorrect relationships between two cybersecurity entities, or may not even assert a relationship. In such a case, we can say that there are a few 'missing' relationships. Our system RelExt improves such knowledge graphs by validating relationships and asserting values for missing relationships.

For example, in Figure 2(a), there is a relationship which is not found by existing methods, hence we can say that this relationship is 'missing' from the knowledge graph. Figure 2(b), shows how RelExt can automatically suggest a value for the missing relationship.

IV. RelExt: System Architecture

In this section, we introduce our RelExt system, which extracts relationships between two named entities. The input to RelExt is a pair of named entities extracted from a particular cybersecurity text. RelExt outputs an entity relationship set.

We use a Named Entity Recognizer (NER) to extract cybersecurity entities from a piece of cybersecurity text. This NER was created and used in the CyberTwitter system [23]. The NER outputs a key-value pair, where the key is a cybersecurity entity and the value is a class as defined in UCO 2.0. A pair of such named entities serve as the input to RelExt.

RelExt's input can be further processed based on our cybersecurity knowledge graph's schema. This allows us to remove some entity pairs which are not consistent with UCO 2.0 and STIX 2.0. For example, we do not provide the relationship extractor with an input where the entity pairs are of the type *Campaign* and *Version*, because our schema suggests there should be no relationship between them. We also remove pairs of entities which are not in close proximity to one another in the pre-processing stage as explained in Figure 4. We have chosen a threshold of 35 words, which is used as a window-size limit. Two entities have to be within a window

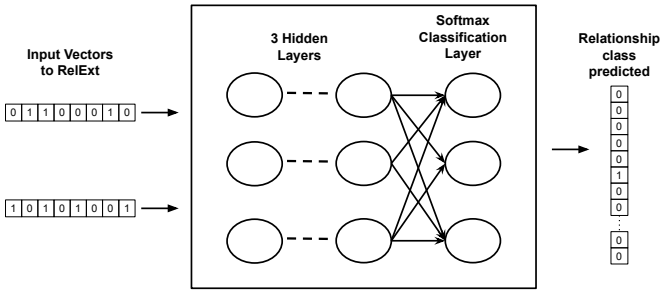

Fig. 3: Relationship Extractor - A Feed Forward Neural Network

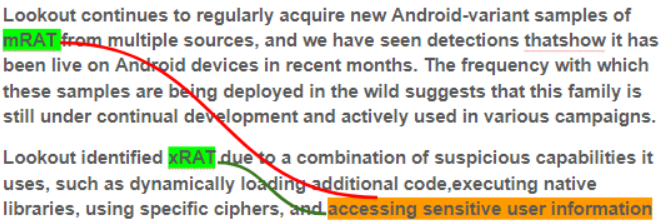

Fig. 4: The pair of entities connected by the red line (*mRAT*, *accessing sensitive information*) is not provided as an input to RelExt as they are not in close proximity. The pair connected by green (*xRAT*, *accessing sensitive information*), however, is preserved.

of 35 words, to be sent to the RelExt system for relationship prediction. We found that the average sentence length in the corpus was 14 words. We keep the window length as a function of the average length of sentences. We empirically chose 35 as the threshold for proximity. Having said that, this number is a heuristic, and it is open to experimentation. Users, of this system, can tune this parameter, depending upon their demand for precision and recall.

Once, we get the processed list of credible entity pairs, we can generate vector embeddings of these entities. We have another neural model, Word2Vec [19], which is trained on a cybersecurity corpus, which takes any particular word and then converts that word into a vector of fixed dimensions. We take two vectors, one for each entity, and provide them as an input to RelExt, which is a Feed-Forward Neural Network (FFNN) classifier. RelExt then predicts one out of the candidate relationships which we have assumed to exist between the named entity pairs from UCO.

The model (See Figure 3) has 3 hidden layers along with input and output layers. Hidden layers form the fully connected layers. Input layer accepts two entities in their vector representations. The neural model accepts the two resulting vectors of dimension 200.

These vector embeddings are concatenated at the initial layer. The concatenated embeddings are propagated to the hidden layers. Hidden layers are non-linearly activated using

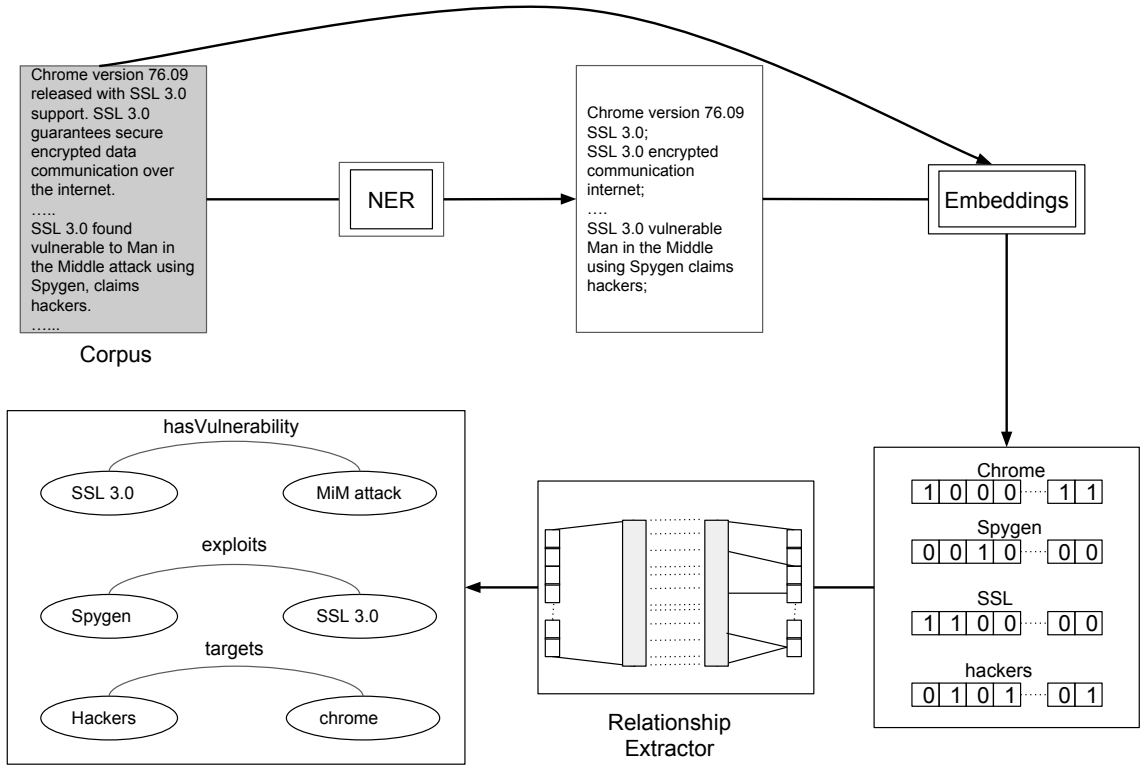

Fig. 5: A cybersecurity document is processed by the NER. NER extracts the entities based on classes defined in UCO 2.0. Further, the vector representations of the entities are generated using a Word2Vec model. A preprocessed set of entity pairs act as input for the RelExt. Based on the relationships predicted for the entities by RelExt, we assert it in the Cybersecurity Knowledge Graph (CKG).

the sigmoid function. Weights and biases are learned through training the neural model. The three hidden layers with 200, 100, 50 neurons respectively. The output is generated using a softmax layer.

1) Updating the CKG: The output of RelExt is a relationship set for every pair of entities which are provided as input to the model. So, for every pair of extracted entities from the cybersecurity text, we predict a relationship, from the candidate relationships. We take this entity-relationship set and assert it in our cybersecurity knowledge graph. As an example, in Figure 6, we showcase a part of our cybersecurity knowledge graph using a tool called Protege [8]. The asserted knowledge graph contains information captured from multiple sources. The example shows description of two malware attacks, *Dark Caracal* and *crossRAT*. We can see all the classes like Vulnerability, Course-of-Action, etc. and the individual entities of the classes which were found in the description. We also see the relationships asserted between the entity pairs, which are represented as lines connecting these entities. E.g., 'Wi-Fi Access Points' is a *Tool* which is used by 'xRAT' *Malware*. We can capture other information related to *Softwares* mentioned in the cybersecurity text. For example, we see *Software* 'Adobe Flash Player' *hasVulnerability Vulnerability* 'sensitive data'. We can also capture other information about the software, such as the other *softwares* 'Adobe Flash Player', in turn, uses. We can see *Software* 'Java SQL Date' is being used by 'Adobe Flash Player'. We also see information about the *exploit-targets* for respective *malwares*. We see 'Whatsapp' as an *exploit-target* which *hasVulnerability* 'phishing messages'. 'Windows OSX', 'PCs and mobile devices' are other *exploit-targets* asserted in the knowledge graph, which are affected by the 'xRAT' Malware. Exploiting 'PCs and mobile devices' can be seen as a part of a *campaign* 'multi-platform espionage campaign'. The information presented in the knowledge graph is useful in deriving similarities between two malware attacks. We can infer that both 'Dark Caracal' and 'xRAT' *use Tool* 'Java'. Further information about which Tools 'Dark Caracal' uses is also asserted in the knowledge graph. 'Pallas' is another Tool, used by 'Dark Caracal'.

V. Experimental Setup and Evaluations

In this section, we first describe our corpus and experimental setup. We then evaluate our RelExt system.

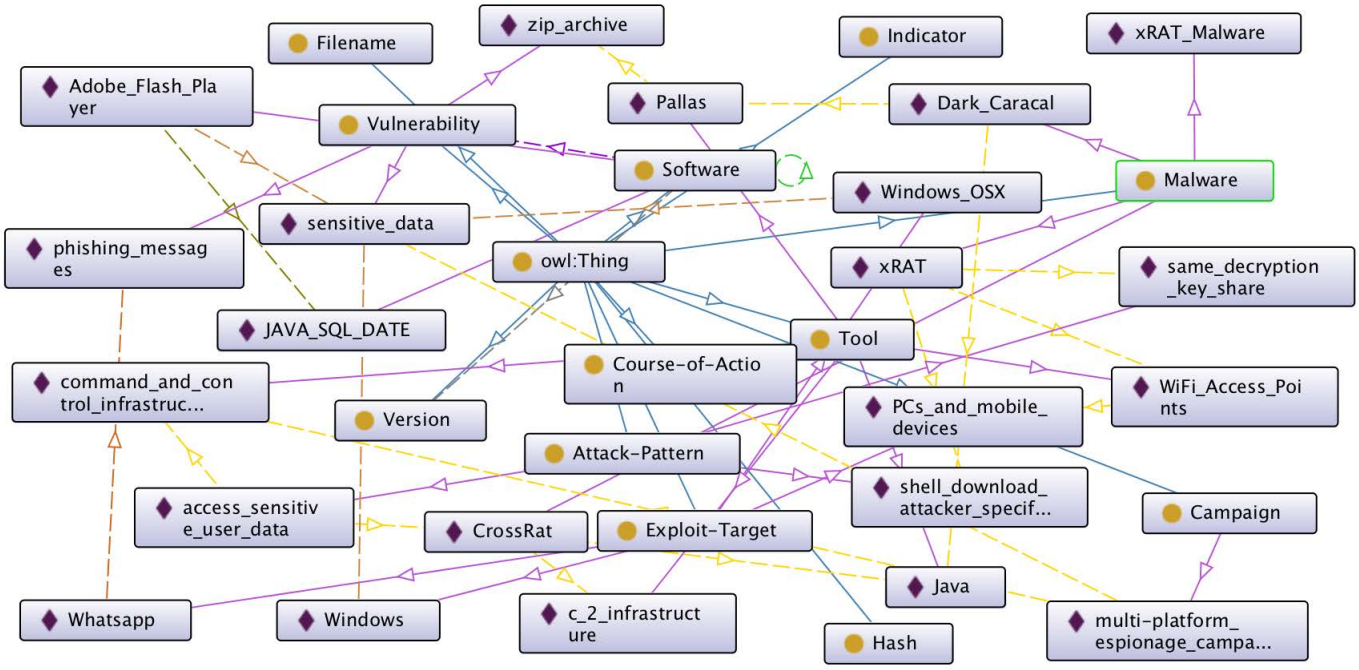

Fig. 6: Asserted Knowledge Graph with relations added using our RelExt system.

A. *Cybersecurity Corpus*

Our proposed system aims to find relations between cybersecurity entities. We construct our corpus from publicly available cybersecurity text. We have extracted information from cybersecurity bulletins, social media, National Vulnerability Datasets. Along with these publicly available information sources, we also collect detailed technical reports authored by cybersecurity specialists. Figure 4, is an extract from one of these technical reports.

However, sometimes this content is not available in raw text. The foremost action would be extracting raw text from these documents/reports. We have used open source libraries available on the internet to extract raw text from HTML, PDF, JSON, XML sources.

We next describe our training set:

- Cybersecurity blogs, cybersecurity technical reports, are publicly available for use over the internet. We procured about 2 GigaBytes of data from publicly available. The data had about 474 detailed technical reports and blogs. The reports have detailed analysis of what security analysts and researchers have learned after studying an attack. Since they are very detailed and size-able reports, they provide us with a large number of relationships that exist between various cybersecurity entities. They are also structurally quite different from one another, as opposed to smaller pieces of cybersecurity texts like Common Vulnerabilities and Exposures (CVE), tweets, etc. This gives us an opportunity to build a generalized model. Having a diverse corpus will help the system to avoid bias.

Splits	Accuracy
80-20	96.21%
70-30	91.88%
66-34	91.34%

TABLE I: Accuracy for various splits on training and testing data.

- *Common Vulnerabilities and Exposures (CVE) Corpus*: CVE JSON feeds have a textual description of the vulnerability present in a product. We collected approximately 90,000 JSON entities which are updated in the National Vulnerability Dataset [20].
- *Microsoft and Adobe Security Bulletins*: Microsoft and Adobe release security flaws and vulnerabilities in their respective software in the form of bulletins. Since these are publicly available and these two companies have produced applications which are ubiquitous, knowledge about them is valuable while detecting an attack. Thus, we also use them in our training set for relationship extraction.

  We can extract information such as *Software* entities that would help us build the *hasProduct* relationship. Moreover, we can also process the problems possessed by the product or a specific version of the product. The problem description contains information about the weaknesses or vulnerabilities known for the product. This information is sometimes represented by Common Weakness Enumeration (CWE) IDs [21]. We have also processed the CWE-IDs made available by MITRE [22]. Whenever a CWE-ID is met, we try to get a description

Relationship Classes	Precision	Recall	F-1 Score
hasProduct	49	97	65
hasVulnerability	92	74	82
uses	100	88	93
indicates	80	90	85
mitigates	55	70	62
related-to	92	74	66

TABLE II: Precision, Recall, and F-1 score for relationship classes.

Document	Entity Pairs	Relations Predicted
Dark Caracal	1287	1127
CrossRat	123	109

TABLE III: Number of Relationships predicted from the malware descriptions of Dark Caracal and CrossRAT malwares.

of the CWE-ID. Problem description ends up being the Vulnerability entities. These entities could form up a *hasVulnerability* relationship with Software entities.

- *STIX Corpus*: We also train our relationship extraction model with the semantic triples generated using the information from Trusted Automated eXchange of Indicator Information (TAXII) [11] servers. These triples are extracted in the source, relation, target format and processed.

Next we use this corpus to generate our Feed Forward Neural Network based RelExt system and evaluate it.

B. Evaluations

We train RelExt with relationships from the aforementioned corpus. Along with manually annotated relationships by cybersecurity experts and relationships that were extracted from the corpus, we have a training set that contains 33,000 relationships. The vector representations are generated using a pre-trained Word2Vec model, over our cybersecurity corpus. The training set is iterated over in a batch size of 100 with 50 epochs. Each element of the batch is a pair of vectors for subject and object entity.

We perform training with various splits like: 80-20, 70-30, and 66-34 and the accuracy metrics for these splits have been presented in Table I. Table II shows the precision and recall values for various relationship classes.

Figure 5, represents the cybersecurity pipeline. Dark Caracal and CrossRat malware descriptions are provided as input to the pipeline. The resulting knowledge graph is shown in Figure 6 and more details about the resulting knowledge graph have been discussed in Section IV-1.

Relationship Classes	Predicted
hasProduct	340
hasVulnerability	93
uses	307

TABLE IV: Various relationship classes found in Dark Caracal and CrossRat malware descriptions.

We have 1287 pairs of entities from Dark Caracal malware description and 123 pairs from CrossRat malware description after filtering out the rest of entity pairs by methods mentioned in section IV. Table III describes the relationships predicted by RelExt from the preprocessed input. Whereas, Table IV shows various classes of relationships predicted for the Dark Caracal and CrossRat malware descriptions.

VI. Conclusion

Cybersecurity Knowledge Graphs (CKG) are useful for storing a large number of semantic triples about cybersecurity entities. When we take an entity relationship set extracted by RelExt and assert it in a knowledge graph, we get access to significant information about various cybersecurity entities.

One such example is the use of Query Languages like SPARQL [34] or SWRL [35] to write queries which can be used to infer information about entities present in a knowledge graph. For example, we can use query languages, like SPARQL to ask the following question: *What software have exposed vulnerabilities?*

```
SELECT ?x WHERE {
?x :type :SOFTWARE.
?y :type :VULNERABILITY.
?x :hasVulnerability ?y.}
```

The above query will return 'Adobe Flash Player' as an answer.

In this paper, we develop the RelExt system that aids in the prediction of relationships between entity-pairs, using a neural network, specifically for cybersecurity. We describe a detailed pipeline, about how we prepare the corpus of cybersecurity texts, and process it further to provide as an input to our RelExt system. The preprocessing step removes entities which are not in proximity of each other as they occur in the cybersecurity text, or they cannot have a meaningful relationship between them. This removes a significant number of candidate semantic triples which are incorrect according to the STIX 2.0 schema. After training is complete, and our RelExt system has learned how to predict a relationship between pairs of entities, we use it to predict relationships from cybersecurity entities scraped from unseen cybersecurity text. This gives us a triple set for cybersecurity which can be asserted in a knowledge graph.

The RelExt system can be used to aid in the development of an entity-relationship set specifically for cybersecurity, which can then be asserted in a cybersecurity knowledge graph. We achieved an accuracy of 96.61 % over various data splits. RelExt successfully predicted over 700 relationships from Dark Caracal and CrossRat malware descriptions.

References

[1] Eugene Agichtein and Luis Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, DL '00, pages 85–94, New York, NY, USA, 2000. ACM.

[2] Andreea Bendovschi. Cyber-attacks – trends, patterns and security countermeasures. *7th INTERNATIONAL CONFERENCE ON FINANCIAL CRIMINOLOGY 2015*, 2015.

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 2787–2795, USA, 2013. Curran Associates Inc.

[4] Robert A. Bridges, Corinne L. Jones, Michael D. Iannacone, and John R. Goodall. Automatic labeling for entity extraction in cyber security. *CoRR*, abs/1308.4941, 2013.

[5] Sergey Brin. Extracting patterns and relations from the world wide web. In Paolo Atzeni, Alberto Mendelzon, and Giansalvatore Mecca, editors, *The World Wide Web and Databases*, pages 172–183, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[6] Symantec Corp. Symantec cyber report. *https://www.symantec.com/security-center/threat-repor*, 2019.

[7] Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Web-scale information extraction in knowitall: (preliminary results). In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, pages 100–110, New York, NY, USA, 2004. ACM.

[8] Stanford Center for BMIR. Protege. https://protege.stanford.edu/, May 2018.

[9] Oasis group. Stix 2.0 documentation. https://oasis-open.github.io/cti-documentation/stix/examples.html, May 2013.

[10] Oasis group. Stix 1.0 documentation. https://stixproject.github.io/documentation/, May 2018.

[11] Oasis group. Taxii. https://oasis-open.github.io/cti-documentation/taxii/intro, May 2019.

[12] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2):199–220, June 1993.

[13] Shizhu He Jun Zhao Guoliang Ji, Kang Liu. Knowledge graph completion with adaptive sparse transfer matrix. *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*, 2016.

[14] Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. Discovering relations among named entities from large corpora. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[15] Yan Jia, Yulu Qi, Huaijun Shang, Rong Jiang, and Aiping Li. A practical approach to constructing a knowledge graph for cybersecurity. *Engineering*, 4(1):53 – 60, 2018. Cybersecurity.

[16] Corinne L. Jones, Robert A. Bridges, Kelly M. T. Huffer, and John R. Goodall. Towards a relation extraction framework for cyber-security concepts. In *Proceedings of the 10th Annual Cyber and Information Security Research Conference*, CISR '15, pages 11:1–11:4, New York, NY, USA, 2015. ACM.

[17] Nanda Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions*, ACLdemo '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[18] Ravendar Lal. Information Extraction of Security related entities and concepts from unstructured text. Master's thesis, May 2013.

[19] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[20] MITRE. Cvelist project. https://github.com/CVEProject/cvelist, May 2013.

[21] MITRE. Cwe mitre. https://cwe.mitre.org/data/index.html, May 2018.

[22] MITRE. Mitre. https://www.mitre.org/, May 2018.

[23] Sudip Mittal, Prajit Kumar Das, Varish Mulwad, Anupam Joshi, and Tim 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*, pages 860–867. IEEE Press, 2016.

[24] Sudip Mittal, Anupam Joshi, and Tim Finin. Thinking, fast and slow: Combining vector spaces and knowledge graphs. *arXiv preprint arXiv:1708.03310*, 2017.

[25] Varish Mulwad, Wenjia Li, Anupam Joshi, Tim Finin, and Krishnamurthy Viswanathan. Extracting information about security vulnerabilities from web text. In *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Volume 03*, WI-IAT '11, pages 257–260, Washington, DC, USA, 2011. IEEE Computer Society.

[26] Sameer Badaskar Nguyen Bach. A review of relation extraction. *To be filled in yet*, 2015.

[27] NisT. Nvd. https://nvd.nist.gov/.

[28] Manikandan R, Krishna Madgula, and Snehanshu Saha. TeamDL at SemEval-2018 task 8: Cybersecurity text analysis using convolutional neural network and conditional random fields. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 868–873, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[29] Victor Raskin, Christian Hempelmann, Katrina E. Triezenberg, and Sergei Nirenburg. Ontology in information security: a useful theoretical foundation and methodological tool. In *NSPW*, 2001.

[30] Benjamin Rosenfeld and Ronen Feldman. Ures : an unsupervised web relation extraction system. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 667–674, Sydney, Australia, July 2006. Association for Computational Linguistics.

[31] Pushpak Bhattacharyya Sachin Pawar, Girish K. Palshikar. Relation extraction : A survey. *Elsevier*, 2017.

[32] Benjamin Strauss, Bethany Toma, Alan Ritter, Marie-Catherine de Marneffe, and Wei Xu. Results of the WNUT16 named entity recognition shared task. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 138–144, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[33] Jeffrey Undercoffer, John Pinkston, Anupam Joshi, and Timothy Finin. A target-centric ontology for intrusion detection. In *IN PROCEEDING OF THE IJCAI-03 WORKSHOP ON ONTOLOGIES AND DISTRIBUTED SYSTEMS. ACAPULCO, AUGUST 9 TH*, pages 47–58. Morgan Kaufmann Pub, 2004.

[34] W3. Sparql query language. https://www.w3.org/TR/rdf-sparql-query/.

[35] W3. Swrl. https://www.w3.org/Submission/SWRL/.

[36] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, 2014.

[37] Maosong Sun Yang Liu Xuan Zhu Yankai Lin, Zhiyuan Liu. Learning entity and relation embeddings for knowledge graph completion. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[38] Tim Finin Lisa Mathews Zareen Syed, Ankur Padia and Anupam Joshi. Uco: A unified cybersecurity ontology. *The Workshops of the Thirtieth AAAI Conference on Artificial Intelligence Artificial Intelligence for Cyber Security: Technical Report WS-16-03*, 2016.

[39] Shubin Zhao and Ralph Grishman. Extracting relations with integrated information using kernel methods. In *ACL*, 2005.