Nitish, 2026, 14:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Cybersecurity knowledge Graph from Malware Attacks Action Reports

Nitish, Bijender, Suraj, Himanshu
Department of Computer Applications
Quantum University, Roorkee, Uttarakhand, India

Abstract- After Action Reports (AARs) provide intensive analysis of cyber incidents. Extraction of materials Cyber-knowledge from these sources will provide security researchers with reliable information, which. It can be used to identify or nd patterns of cyberattacks. In this paper we describe a framework to. extract information from AARs, combine similar organizations and aggregate the extracted information, and. They represent extracts from the Cybersecurity Knowledge Graph (CKG). We remove entities by We are creating a custom named entity detector called Malware Entity Extractor (MEE). Then we a neural network to predict how pairs of malware entities are related to each other. When we predicted For entity pairs and the relationships between them, we represent the entity-relation set in CKG. Our next one The step in the process is the consolidation of similar organizations, reforming our CKGs. This mixture helps to represent the intelligent fish Extracts from numerous papers and reports. Fused CKG is known from many AARs, as well. Inter- organizational relationships extracted from separate reports. Because of this mix, a security researcher. can handle questions and retrieve better answers on a hybrid CKG, than an unhybrid knowledge article. Wealso showcase the various logic capabilities that can be leveraged by the security analyst to bring us fused C. K. G

Keywords— Artificial intelligence, computer security, cyber threat intelligence, post-action reports, knowledge graphs, and semantic web.

I. INTRODUCTION

Every year, thousands of malware are created and subsequently used to attack organizations. In March 2018,. Nuance Technologies announced a loss of US$92 million Dollars Notpetya for attacks by malware [6]. At the same time, the hackers began a series of incidents of cyberattacks at several universities [48]. The attacker 31 terabytes of data were stolen and an estimated total loss It was about US$3 billion. spear-phishing emails Pre-filtering sensitive was also used to steal login credentials References [48]. To combat these malware-based attacks, security researchers take malware samples from the 'wild'. This model is then 'exploded' under controlled conditions and its behavior is 'written' [26], [27]. The implementation of this Behavioral data, security analysts map malware types toknown indicators and attack methods. Investigators and forensic experts from companies such as Mandiant are involved After an attack, where they look for the malware used, the tools and clues to the attack, and. The result of an attack. As a result of these studies , these. Safety analysts produce 'After Action Reports' (AARs), . detailing one particular example of fragrance, . The process and the results. These technical AARs are important source of Cyber Threat Intelligence (CTI) and potential growth To create additional sources of Open Source Intelligence (OSINT). Full picture of the attack. Later, this A.A.R they are used for defensive or modification techniques It was used to protect infrastructure, detect and prevent future attacks. Sometimes it is used to identify criminals and attribute this attack to a known hacker Different groups. The AAR contains data retrieved upon completion Investigation of attacks.

If security analysts base their primary source of knowledge on AARs, they can obtain more relevant information about attacks if. Not only does it help

them see their parts more clearly of unknown attacks, but also draws parallels between New attacks and previously known attacks. This is the power that it draws parallels to attacks faced in the past Gaining knowledge from the numbers enhances it of AARs involving hazardous substances, or similar malice. In addition, extracted knowledge can be automated Protective equipment In this paper, we aim to gain knowledge from these AARs by security researchers. going in numbers of cybersecurity blogs and AARs is not only difficult but. Also impossible for a short-term security professional. We exclude cybersecurity awareness from these reports, and. Do it in the Knowledge Base. This can then be used and should be interviewed by researchers to support defensive programs [35] . The cybersecurity sector is a niche sector in terms Vocabulary/language. and in general, natural comparison Language processing industry, there is not much data available To build data-driven knowledge extraction methods. Also decisions trained on commonly available corpuses do not work well for cybersecurity lessons. and one of the The most important contribution of our work is the cybersecurity curriculum corpus that helps build machine learning algorithms to.

Extract information from text about cybersecurity. They work on the cybersecurity corpus we build AAR, and other blogs and technical reports. (consider Section III-a). We created a new annotated data set for Cyber-knowledge mining. We reviewed 474 AARs and APTs (Increased Ongoing Threats) Reports, 550 security.

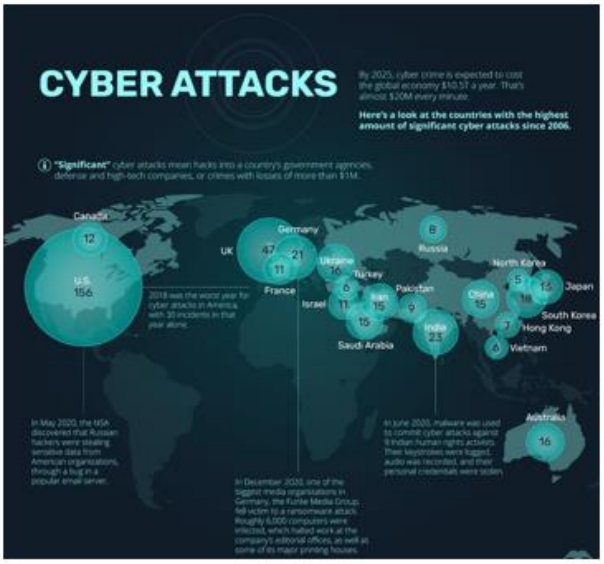
Our second major contribution is to create an aggregate tools, used in pipelines, to extract cybersecurity awareness From various texts. Extracting our cybersecurity expertise The processing pipeline had 3 components, one being a Malware Entity extractor (MEE), relation extraction (RelExt), and. end Cybersecurity Knowledge Graph (CKG). The MEE happened Trained in documented cybersecurity course sections, to predict Cybersecurity companies in the AAR. MEE is cybersecurity a designated employer (NER) (see Section III-B1). It does have some It was constructed using Conditional Random Fields, Gibbs' sampling, and regular expressions. RelExt (see Section III-B2) . is a novel deep learning based relationship extractor and. trained in cybersecurity issues, to predict the relationship In the entity pairs selected by the MEE. Once we get one These named entities are then filtered entity pairs That there can be no relationship between them as such Our plan for CKG. Then we cut the other two entities to the relationship exclusion. The relation extractor will try to take two names ('A' and 'B') as input predict a particular relationship as an outcome.

Then we fill people and our CKG (see Section III-C) entity relationship policies From the extracted data. CKG stands for unstructured Data in AAR: Towards a structured ontology. Once upon a time we in the entity-relationship set from RelExt, we can populate CKG with three ('Entity A'–in relation 'R'–). 'product b').

To improve the quality of the CKG, we also combine knowledge from different AARs describing the same firm. This fusion greatly benefits security analyst communities Cybersecurity knowledge pieces from many sources and. Understand how those pieces are connected. Security analysts It can then use this blend of knowledge to get better information Cybersecurity decisions. This helps him ask a question fused knowledge graph, and retrieve entities from different The sources of the sources.

This also reveals what was previously unknown Relationships. This synthesis of knowledge from AARs It is one of the main contributions of this work. We're doing a demonstration This power is contained in Section III-D. Our ability to reason and analyze is greatly improved Because of our CKG fusion.

The blending of knowledge from different AARs helps the security analyst achieve the best results His questions. This highlights the need for hybrid steps Correct the lower CKG. We manifest this power Executing the same query in Fused CKG, too Knowledge accounts before integration. Some sample questions found in Section IV-D.



Our system can be used by security researchers for analytical purposes Cyber intelligence data, as well as similarities, have been prepared between a new cyber event and a previously encountered cyber event already highlighted in our CKG. Some kind of security The analyst can use the SPARQL [46] endpoint to query above Ask them to graph and calculate the response of the complex in the system questions and context (see Section IV-D).

We organize the paper as follows: The second part has speech in the middle Details of AAR and similar studies by others The researcher. Part III, describes the construction in detail about our system and how we build the pipeline. Part IV, information about both our experimental observations and the properties of the individual components of our pipeline. Po V, 1999. It speaks to the scope for further research in this area.

Background and Related Work

In this section, we define AARs and discuss the same Research in this area.

After Action Reports (Aar) and Open Source Intelligence (OSINT)

The AAR contains information about the specific submitted action as ordered by the United States Department of Homeland Security [44]. The method of sharing exercises and research has been standardized by Homeland Safety Operations Monitoring Program, to ensure consistency of shared reporting across jurisdictions.

There are different areas in AAR in terms of cybersecurity Information on cyberattack detection exercises and information on mitigation strategies. apart Security companies from government agencies also reveal it ARS. Examples of such companies include 'Kaspersky'. [20] and 'Agninetram' These AARs contain detailed analyzes of various types of cyberattacks. Tools related to specific malware, AAR, threat-actor , or campaign costs To detect an attack from pointers that do nothing but. ngerprints left at the scene of the attack are disposed of by the agent The attack. It can also tell you what the weaknesses were targeted by the attacker, and whether the user, or a What he can do wrong, can do to stop the attack. AAR provides security analysts for cyberattacks Knowledgeable, picking and passing through one is given to the ordained mat, allowing this information to be used for research purposes Future attacks. AAR, is also a reliable source of info mation to mine intelligence, because they have shown themselves of cial security agency websites. This generates these reports A trusted source of intelligence against mining data From dark web circles, or social media as they have been shown Mittal and so on. [32] [34], where the accuracy of the info mation, cannot be true. Another reason for our choice The AAR should be our source of knowledge that these are open source. Security companies are abandoning the whole curve.

After Action Reports (AARs) are different from techni cal blogs have included more technical details about the malware attitude. AAR provides security analysts with technical information and knowledge of the cyber incident. Internet Safety Blogs, . is different from AARs and does not capture enough technical information Mation. Many of these blogs are meant for the average Audi ence, they had only super cial information about the attack. Another important point, in the context of our source material of knowledge, and that there should be blogs coming from indi viduals they may not know and have no need for Technical knowledge. Compared to their reports and blogs, A.A.R is released by recognized cybersecurity researchers and organizations zations.

They also have depth technical details whichmight it may be a useful source of knowledge for security researchers.

Cybersecurity Knowledge Graphs

The knowledge account consists of three sets of meanings, viz pairs of related entities. Cyber security Knowledge Graphics (CKGs) have been used for a long time To stand for Cyber Threat Intelligence (CTI). The first step in representing CTI in CKG is to identify organizations and. Relationships should be emphasized. We also use ontology called the Uni ed Cybersecurity Ontology (UCO) [45] for pro vide our program and cybersecurity field knowledge. Yuko Structured Threat Intelligence Language (STIX 2.0) [12] which provides a framework for representing cyber threats wise. CKGs have been developed from others as well Open data by Mittal et al. [32], CKG, respectively and also use knowledge statistics for concepts Analyte development process [18], [19], [21], [36], [40], . [42] . We then discuss 3 main aspects of our system named entity identifier, relationship exclusion, a system allows you to compare the number of malicious nodes in the CKG.

Named Entity Recognition

It is called Entity Recognition (NER) for Cyber Threat Information tion extraction is done with Conditional Random field (CRF), mechanical support vector [7], and neural networks Activity Ekbal and so on. Their paper [7] proposes that Non-language algorithm for named object search. Bi-directional LSTM is currently being used for detection Named Companies. Even in the realm of CTI, power removal. It was developed with the help of in-depth studies [4], a The measurements, in the CTI area, are also presented The use of neural networks in gestural objects a. gave good results [24], [41]. Despite its widespread use cations of Short-Term Speed Memory (LSTM), use CRF-based classifiers enable continuous extraction. Maintain the status quo [43]. Use of BiDirectional LSTMs, accessories in understanding (or forgetting) long-term context,. required to predict the class of objects. However, some Our favorite brands, like lenames , . IPAddresses , or hashes , do not need context information tion to be predicted to the appropriate class. take various types Context can also erode these categories Functionality of entity classification. Moreover, BiLSTMs over t and Limited data. So we create our entity extractor for Cybersecurity courses based on CRF and regular feedback (See Section III-B1)

II. RELATIONSHIP EXTRACTOR

Relationship extraction predicts interactions or relationships existing between pairs of objects excluded from our system. They have worked primarily on relationships extraction between entities. It can symbolize relationships Many-many-one, or one-one. TransH model [47] have worked to extract many-to- many maps Transfer of vector positions in hyperplanes. TransE examples[3]. Head-tree features were used to predict each map. Sparse vectors [15] have also been used to predict the relationship Ship design, which was an improvement over TransH and. TransE examples. We use the Relationship Extraction method proposed by Pingle et al. [39], which is our previous work This is the field. Relationship Extraction algorithm uses vector Encoding of individual entities created using word2vec [29]. These vector representations help capture context Cybersecurity entities in the text, which support rela work tionship prediction.

III. UTILIZING CKG FOR MALWARE COMPARISON

Significant research has been done in the field When you compare the negative. Some of these products use a device extract and study features associated with this malware [22]. An interesting method provides educational images indicators of goals to be achieved [2]. followed by the authors Machine learning algorithms were used for various types of soft classification Things that are not bad or malicious. Other graph-based methods That includes creating a malware-related social account, based on it system calls such as Park et al. [38] . after In constructing these diagrams, the authors used a technique that. subgraph matching calculated the similarity between malignancies. However, only the system would be affected this way Behavior at the scene of the assault.

It doesn't really take Broad terms such as, what is the target software, or if this The malware is part of a larger campaign. they should not take, Examples of aggressive attacks in natural language. The The CKG we build has technical issues, too As for the campaigns launched, the tools used, target software and so on. So we can use CKG, subtract it Three times about malware, and by comparison three times about Just another nasty piece of malware and compare them, and you've created a similar group They are terrible. Jiang et al. In his paper [16], it is suggested that a A method for reconstructing the semantic view of the host machine On virtual machines and running malware scans. To fix the problem of malware hiding VM detection software. How about in the paper, recreating… A semantic view of the host machine in the VM, is possible To detect hiding malware by le comparison. although other types of malware can be found Using a semantic perspective does not give us a higher level Details of the malware.

The vision of understanding that happened The reorganization of the paper focuses on speci c metrics. Some examples of these metrics include processes, memory, . Files types. Our STIX-based CKG not only saves the system speci c information, such as lenames and hashes, it can also capture a wealth of information that is helpful for secu rity researchers. For example, by conducting content analysis Infected device, we can collect information if le suspect tries to access a remote IP address and attempts to provide information. ARS, 9. List. If we have already examined this, it can confirm this System control results. They use malware. Because we collect our information from AARs, we choose these keywords, and we do CKG. So the security analyst can easily detect all the malware The command and control infrastructure of our CKG and. Fair results will be given.

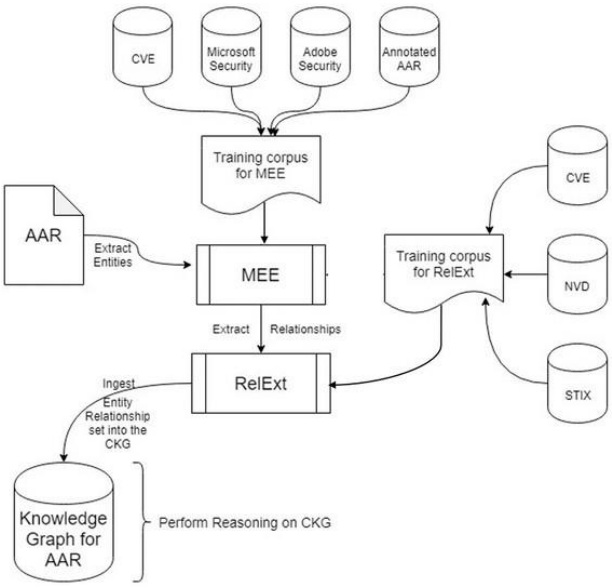
Strategy

Figure 2 depicts our proposed pipeline, which takes an AAR as input. The trained Malware Entity Extractor (MEE) extracts entities from AAR and sends the extracted set.

of entities into the Relationship Extractor (RelExt). The trained connection extractor estimates the optimal relationship between two entities. The relationship extractor generates an entity-relationship set, which the CKG module uses to create a knowledge graph.

When two AARs refer to the same virus, their entity relationships are combined. The system's fused output is asserted in the CKG. Our system includes the following essential components:

- ->MEE: a Malware Entity Extractor trained on annotated cybersecurity language, can predict cybersecurity entities in an AAR.
- ->RelExt: is a relationship extractor trained on cybersecurity data to anticipate relationships between pairs of elements extracted by MEE.



System Architecture with RelExt, CKG, and MEE.

->CKG: Cybersecurity Knowledge Graphs populate entity relationship sets from extracted data. The CKG organizes the unstructured material in the AAR into a structured ontology.

After Action Report

Our collection comprises AARs from many sources, including cybersecurity companies listed in Section II-A. We have compiled 474 reports detailing cyber-attacks. Our dataset is approximately 1 GB. All handpicked reports in our pipeline are in English. We picked reports that were consistent with the

language used to train our MEE and vector embeddings for RelExt's entities, which were trained on a corpus of cybersecurity text in English. Our corpus includes extensive analysis reports from the cybersecurity companies included in Section II-A. We contain notes on advanced persistent threats, including reports from government entities such as the Intelligence Research Team. The reports were in PDF format and processed as raw text. Images may be embedded in PDF documents. Our processing cannot detect the information in these photos. We extracted the raw text from

Use industry-standard tools to convert PDF files into a CKG format.

Extraction of Cybersecurity Knowledge From After Action Reports

A schema for representing cyber-threat intelligence was proposed by Syed et al. in their study on the Uni ed Cybersecurity Ontology [45]. STIX 1.2 [13], a standard for exchanging cyberthreat data, served as the foundation for UCO 1.0. STIX 2.0 [12] served as the foundation for UCO 2.0, which has now replaced the earlier iteration of STIX [39]. The classes and connections specified in UCO 2.0 serve as the foundation for our CKG's schema.

In addition to a few new classes that better reflect cyber-threat intelligence from an AARare, certain significant classes from UCO 2.0 have been utilized in our CKG.

- Software: An entity like Ofce or Adobe that is associated with an element of code that is usually used as a tool.
- A person or thing related to the attack site that usually gets attacked by malware, like Android, or an operating system, like Windows, has been identified as an exploit-target.
- Malware is a word used to refer to software and/or malicious code that is introduced into a system.
- An entity with a pattern that helps the administrator is known as an indicator.
- Vulnerability: A term used to describe a bug or vulnerability that could be exploited by malicious users.
- Course-of-action: A term used to describe a course of action or series of actions that either stop or neutralize an attack.
- Tool: A term used to describe trustworthy software that threat actors may utilize for nefarious purposes.
- Attack-pattern: A term used to describe actions that can contribute to an ongoing attack against a user or group of users.
- Campaign: A term used to describe a collection of activities that may result in a malevolent attack.
- Filename: A le that the malicious software uses to carry out the attack.
- Hash: An executable's SHA-256 hash that could be utilized to identify an attack.
- IP addresses: any number of IP addresses that malicious software could be using during an attack.

Malware Entity Extractor (MEE)

Each report is sent to the Malware Entity Extractor (MEE) following the compilation of the AAR corpus. Data from many sources, including Common Vulnerability and Exposures (CVE), security blogs, STIX datasets, AARs, dark web articles, etc., were used to train the MEE independently. The MEE has been taught so that, given a text passage, it is going to predict an entity class as output for each word. MEE seeks to identify the kind of knowledge that is included in the AAR. The basis of our knowledge graph schema is Uni ed Cybersecurity. UCO 2.0 Ontology [45].

Based on the Stanford NER [8], MEE attempts to match possible labels it might discover for a word with a similar context in the training set after receiving as input the context of a specific word for which it has been tasked to find the label. It also affects the sentence's structure. It is clear from the way MEE has been trained that the model performs better than a class predictor if that specific word has been observed before. Sentences are added to the training 211696 set, which is appended to the training corpus, for AARs. In addition to AARs, the corpus contains sentences from various additional cybersecurity sources.

In the parts that follow, we go into detail about MEE's performance. Despite not having seen a lot of AAR important in its training set, the model does well. In essence, this shows that our MEE can function well with unseen data, which is crucial as we want it to function well with novel AARs. The entities and their classes in the Cybersecurity Knowledge Graph (CKG) are asserted using these named entities. There are two sub-components in our MEE model.

A Conditional Random Field (CRF) component to identify different named entities and classes, as well as the regular expression model to extract le names, hashes, and IP addresses from the les (see Section III-B). These two subcomponents are further explained.

Regular Expressions

The first component employs regular expressions to identify textual fragments that correspond to classes that may be identified without a context. Some examples of these include IP address hash for the filename

We provide basic regular expressions that aid in IP address detection:
(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9]) (25[0-5]|2[0-4]
[0-9]|1[0-9][0-9]|[1-9]?[0-9])[0-9]|1 [0-9](25[0-5]|2[0-4])\.[0-
9]|[1-9]?[0-9])[0-9]|1[0-9][1-9]?[0-9]| [0-9])
and we can identify le hashes utilizing a regular expression: [0-9a-f] {5,40} \[0-9a-f]{10,64}.

Random Fields of Environment

To find the malicious entities, we employ a conditional random field (CRF) called the Entity Recognizer (NER) system from CoreNLP [25], which needs contextual information. Approximately 53,829 words from 6184 phrases were taken from a variety of sources, such as

- Microsoft Security Bulletins [28].
- Blogs on Adobe Security [1]
- CVEdescriptors [30]
- After-action reports with annotations (AARs)

To create a training set, we annotated some of them. In Section IV, we outline the different training divides.A. Manning et al. have explained in their study [8] how non-local information can be captured more easily when Gibbs sampling is used in place of the conventional Viterbi encoding. A probabilistic graphical model called CRF captures the bidirectional flow of data. The CRF model transforms the graphical model into linear chains in order to create Markov chains, which are necessary for Gibbs sampling to function. This ensures label uniformity and aids in gathering non-local information. There may not always be enough context for our model to deduce that the word "Adobe Reader" in a phrase indicates that it is software. However, even if there was non-local evidence indicating Adobe Reader was software, our MEE could correctly identify the proper word class with the use of a CRF model enhanced by Gibbs sampling. In Section IV, we go into MEE's and its components' performance.

A few of the relationships and categories that the VOWL visualizer displays.

Relationship Extractor (Relext)

The following step of our pipeline, which determines whether a relationship exists between a certain pair of entities, receives pairs of malware entities that our MEE has collected. In essence, the RelExt is a neural network that predicts a relationship between two vectors that represent two things that were extracted by our MEE. A Word2vec [29] model is trained independently, and the extracted items are fed into the model to produce their corresponding embeddings. The embeddings have a dimension of 200. Two embeddings are fed into the RelExt neural network. Its output consists of a softmax layer, three hidden layers, and an ab input layer. The input layer is 400 in size, while the hidden layers are 200, 100, and 50 in size, respectively. The output softmax layer is six in size.

Before sending entity pairs to RelExt, we post-process them based on our CKG's pre-defined schema. We only pass entity pairs with a credible relationship and delete those that do not have a relationship based on our CKG schema. Our CKG determines that there is no direct relationship between a Software and a Filename. Therefore,

entity-pairs of this type are not considered for the next stage. As we already have a schema for our CKG, we immediately filter out pairings of entities with no plausible relationships between them. Our pipeline features a neural network that predicts the connection between two things.

To extract relationships, the input elements are first represented as vectors, which neural networks can then conduct matrix multiplication on. The MEE output (refer to Section III-B1) is unsuitable for neural networks as it only produces a class type or entity type for a specific word. We produce word2vec embeddings [29] for each word in the corpus used for MEE training. Word2Vec is a way for representing words within a corpus.

The Word2vec technique collects the context of each word in the corpus and represents it with a vector of a specific length. We trained word2vec on cybersecurity text from NVD [37], CVE [30], and STIX data from TAXIIservers [39] to accurately represent each entity recorded by our MEE.

The relationship extractor's second step is to develop the neural network using NVD, CVE, and STIX datasets. We labeled pairs of entities and their relationships. Our dataset comprises of 33,000 labeled connections.

We divided the dataset into training and validation sets.

We employ various split ratios: 80 (train), 20 (validation); 70 (train), 30 (validation); and 90 (train), 10 (validation). Tables 3 and 4 present the experimental findings of RelExt on the validation set. After training the model to produce vector output for any input word, we use the predicted entities from the AARs to construct embedded data for different topics. We train the neural network model RelExt to identify the relationship between pairs of embeddings. We analyze our Relext model in Section IV-B.

The neural network in the RelExt component creates a relationship between cyber entities, such as Hellsing (Malware) and doubtful emails.

- Hellsing (Malware) uses doubtful emails.
- Hellsing (Malware) depends on 015915BB. (Hash) .

In the next stage, we will populate the CKG.

Cybersecurity Knowledge Graph (Ckg) Fusion Cybersecurity Knowledge Graph with Evaluation

After generating the entity-relationship set from our system's relationship extraction component, we can add it to our cybersecurity knowledge graph. Our knowledge graph schema is based on STIX (Structured Threat Intelligence) [12] and utilizes UCO 2.0 [45] for cybersecurity domain knowledge in the system. The relation extraction process yields semantic triples and entity types. A knowledge graph's schema is defined by its entity classes, relations, and specific classes that operate as domains or ranges for each relation.

It's a description of our CKG schema. Each relation in our CKG is listed below, along with its domain and range.

- Allocated to: domain:malware, tool, or vulnerability. Range:Campaign.
- indicates: Domain: Indicator. Range: Malware or tool.
- hasProduct: Domain:Software.
- Range: Software can help mitigate: Domain:Course-of- Action.
- Range: Malware, Tool, or Vulnerability.
- hasVulnerability: Domain: software or exploit-target. Range:Vulnerability.
- uses: Domain: Malware, Tool, or Attack Pattern. Range: Malware, Tool, or Vulnerability.

Our collection contains many AARs that describe the same attacks or malware. To construct a more robust CKG, we want to combine knowledge from multiple AARs that describe the same virus. If an AAR entity is already asserted in our CKG and there is an exact match, we utilize owl:SameAs to fuse the graph entities. We declare that nodes and subgraphs are the same. To determine the similarity between a newly discovered AAR object and previously processed AARs, we use TF-IDF scores. We compare string similarity between new and existing entities using edit-distance metrics. If there is a close match,

we merge them. Fusion identifies malware or cyber-entities across multiple AAR sources.This increases the robustness of our CKG by incorporating additional information about cyber entities.
Here's an example query for the unfusedCKG.

SELECT?x WHERE {
?x a CKG:Malware; CKG:
uses CKG:
588 f41bbc117346355113f.}.

The above query returns Hellsing.
The similar query for the fusedCKG yields: SELECT?x WHERE { ?x a FusedCKG: Malware; FusedCKG: uses; FusedCKG: 588 f41bbc117346355113f.}
The above query returns Hellsing and XWeber.

The Cybersecurity Knowledge Graph includes Hellsing malware entities.

The figure shows how the Hellsing virus and its properties are identified and asserted in the CKG. Hellsing employs various executables, including cmd.exe, test.exe, and xKat.exe. This was generated using entity relationship sets from two reports on the same virus. One report mentioned xkat.exe, while another mentioned xKat.exe. Using an edit-distance approach, we determined the similarity between two entities and used an owl:SameAs relation to confirm their identity. The owl:SameAs assertion ensures that all relationships stored by one node also apply to the other. We also see additional entities highlighted.

The Hellsing malware uses tools such as 7zip archives.

IV. CONCLUSION AND FUTURE WORK

By the effective development of a pipeline for creative systems, cybersecurity entities are automatically extracted from After Action Reports (AARs). The technology creates a Cybersecurity Knowledge Graph (CKG) by analyzing the relationships between each pair of those entities. The pipeline combines information from several AARs that detail the same attack with knowledge obtained from one AAR. Our CKG has reasoning capabilities, meaning it helps end users in running queries and identifying patterns among various cyberattacks. We used relatively shorter cybersecurity literature to train our MEE and RelExt.

includes xedstructure, such as text from Adobe Security Bulletins, Microsoft Security Bulletins, or CVEs.

We showed how to generate generalized the MEE and RelExt by including a few terms from AARs to our corpus of other relevant cybersecurity content. MEE and RelExt can extract entities, work successfully with unseen text data from AARs, and create the right connections between them. The fact that the text data in our training set comes from a variety of sources, strengthens our pipeline. We additionally showed that once we run the same queries on the fused CKG, we obtain more information than when we run the same query on the unfused knowledge graph.

This highlights how the fusion process increases the quality of the knowledge graph characterizing every AAR. A threat analyst can then take benefit from this improvement and use the information to better defend an enterprise. Without just applying these reports to train our extractor algorithms, we have shown that it is possible to extract information from AARs.

Neural models could one day be applied to calculate vector embeddings of the entities in our CKG and utilize them for malware attribution. Our CKG reasoner would be ready to come to better decisions if we extended the schemaofour ontology to include more info about cyberattacks.

REFERENCES

1. Adobe, “Adobe Security Blog.” [Online]. Available: https://blogs.adobe.com/security/
2. A. Author, "An interesting method provides educational images," Journal/Conference Name, Year. (Placeholder — original details missing)
3. A. Bordes, N. Usunier, A. Garcia- Durán, J. Weston, and O. Yakhnenko, “ Translating embeddings for modeling multi- relational data,” Advances in

4. Neural Information Processing Systems, vol. 26, 2013. B. Liu, M. Zhou, and C. Zhang, "Extracting event and entity information for cyber threat intelligence," Proceedings of the AAAI Conference on Artificial Intelligence, 2020.
5. Nuance Communications Inc., "Cyberattack disclosure report – Not Petya," 2018.
6. A. Ekbal and S. Saha, "Deep learning- based named entity recognition: A survey," ACM Computing Surveys, vol. 53, no. 4, 2020.A. Ekbal and S. Saha, "Deep learning- based named entity recognition: A survey," ACM Computing Surveys, vol. 53, no. 4, 2020.
7. [8] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. Mc Closky, " The Stanford Core NLP Natural Language Processing Toolkit," Association for Computational Linguistics (ACL) System Demonstrations, pp. 55– 60, 2014.
8. OASIS, "STIX Version 2. 0. Part 1: STIX Core Concepts," [Online]. Available: https://oasis-open.github. io/cti- documentation/ stix/ intro
9. OASIS, "STIX Version 1. 2. 1. Part 1: Overview," [Online]. Available: https://docs.oasis-open.org/cti/ stix/ v1. 2. 1/
10. R. Socher et al., "Reasoning With Neural Tensor Networks for Knowledge Base Completion,"Advances in Neural Information Processing Systems (Neur IPS), 2013.
11. M. Jiang, B. Yu, and J. Xie, "Reconstructing semantic views of host machines in virtualenvironments," IEEE Transactions on Information Forensics and Security, vol. 14, no. 6, pp. 1501–1513, 2019.
12. 18]–[21] Various works on knowledge graph construction in cybersecurity (details not explicitly given in the paper).
13. A. Author, "Feature extraction and classification of malware," Conference/ Journal, Year. (Placeholder)
14. Y. Goldberg, "Neural Network Methods for Natural Language Processing," Synthesis Lectures on Human Language Technologies, vol. 10, no. 1, pp. 1– 309, 2017. [26], [27] General references to malware sandboxing behavior analysis (details not given).
15. Microsoft, "Microsoft Security Bulletins." [Online]. Available: https://learn.microsoft.com/en-us/security-updates/securitybulletins/
16. T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv preprint arXiv: 1301. 3781, 2013.
MITRE, "Common Vulnerabilities and Exposures (CVE)." [Online]. Available: https://cve.mitre.org/
M. Mittal et al., "Open source intelligence for threat detection," Cyber Threat Intelligence Conference, 2019.
17. M. Mittal et al., "Social media mining for cyber threat detection," Journal of Cyber Security Technology, vol. 3, no. 1, 2019.
18. A. Author, "Supporting defensive programsvia threat intelligence," Journal/Conference, Year. (Placeholder)
19. 40], [41], [42] Related works on CKGs and neural methods in cybersecurity (partially referenced; full details missing).
20. NIST, "National Vulnerability Database (NVD)." [Online]. Available: https://nvd.nist.gov/
21. Y. Park et al., "Graph- based malware detection using system calls," IEEE Transactions onDependable and Secure Computing, vol. 15, no. 6, pp. 988– 1001, 2018.
22. S. Pingle et al., "Relationship extraction for cybersecurity knowledge graphs," Proceedings of the Workshop on Security and Privacy in NLP, 2020.
23. L. Yao, A. Mao, and C. Zhang, "Using LSTM for cyber entity classification," IEEE ICMLA, 2019.
24. K. Lafferty, A. Mc Callum, and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," Proc. ICML, 2001.
25. U. S. Department of Homeland Security, "Homeland Safety Operations Monitoring Program," 2020.
26. S. Syed et al., "UCO: A Unified Cybersecurity Ontology," Semantic Technology for Intelligence, Defense, and Security (STIDS), 2016.
27. SPARQL Query Language for RDF, W3C Recommendation. [Online]. Available:https://www.w3.org/ TR/ sparql11-query/

28. Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge Graph Embedding by Translating on Hyperplanes," AAAI Conference on ArtificialIntelligence, 2014.
29. Cyberattacks on Universities, 2018. (Exact source not listed; referenced in narrative)