Extracting Rich Semantic Information about Cybersecurity Events

Taneeya Satyapanich, Tim Finin and Francis Ferraro
Computer Science & Electrical Engineering
University of Maryland, Baltimore County
Baltimore, MD 21043
taneeya1@umbc.edu, finin@umbc.edu, ferraro@umbc.edu

Abstract—Articles about cybersecurity events like data breaches and ransomware attacks are common, both in general news and technical sources. Automatically extracting structured information from these can provide valuable information to inform both human analysts and computer systems. In this paper we describe how cybersecurity events can be described via semantic schemas, examined through an initial set of five event types. Using a collection of 1,000 news articles annotated with these event types, including their semantic roles, arguments, realis, and coreference, we detail a modular, deep-learning based information extraction (IE) pipeline, which extracts useful event information with high accuracy. We argue that the event argument set considered here can support many other cybersecurity events, facilitating the extension to new cybersecurity event types, such as distributed denial of service and SQL injection attacks.

Index Terms—cybersecurity, event detection, information extraction, event schema

I. INTRODUCTION

An event information extraction system analyzes documents to identify descriptions of specific events and construct a semantic representation of them from the information in the text. More specifically, it aims to detect event triggers, participants, location, and other properties and to classify the event as one of a fixed set of relevant types.

Cybercrimes are increasingly widespread and common. The cybersecurity news contains many details expressed in both common and technical terms. Event extraction techniques can be applied to extract useful information from such cybercrime news. Useful information includes who performed the attack, what steps they used, who was the target, when the crime happened, who will be affected by the discovered vulnerability, and when a patch will be released. Given the following text, the system can produce the information shown in Table I.

The phishing scheme which may have circulated to 1 million Gmail users is particularly effective because it fooled users with a dummy app that looked like Google Docs. Recipients who received the email were invited to click a blue box that said "Open in Docs." Those who did were brought to an actual Google account page that asks them to handover Gmail access to the dummy app.

If we can provide such useful and simple information to people, they can be aware of the same phishing technique in

TABLE I
AN EVENT FRAME FILLED WITH A PHISHING EVENT. MULTIPLE INSTANCES ARE SEPARATED BY SEMI-COLONS (;)

Event Info	Details
Event type	Phishing
Attacker	N/A
Attack-Pattern	click a blue box that said "Open in Docs."
Number of Victim	1 million
Purpose	handover Gmail access
Tool	a email; dummy app; Google account page
Trusted-Entity	Google Docs
Victim	Gmail users; Recipients

the near future. These can be applied to other cybersecurity events as well. While the general task of extracting information about events has been studied for common events involving people and organizations (e.g., [20], [37]), very little work has been done on events in the cybersecurity domain.

In this paper, we discuss the problem of cybersecurity event information extraction. Our goal is to analyze any cybersecurity news article and present useful information in a structured form, such as an event frame, that can be easily understood by both non-technical people and cybersecurity experts and can be used to populate a semantically-grounded knowledge graph.

Our analysis centers around cybersecurity event schemas designed for an initial set of five event subtypes of cyberattack and vulnerability-related events [32]. Each event in this schema has been defined with an event definition frame that represents the event's semantic roles and argument types that can fill them, including a *realis* attribute that captures whether or not an event instance has actually occurred. To the best of our knowledge, this is the broadest range of cybersecurity events, covering five event subtypes.

Using 1,000 cybersecurity news articles and annotated with this event schema [32], we built the cybersecurity event information extraction system capable of identifying the events, their arguments, event realis, and event coreference, which groups the different mention of the same event in a document. This collection of annotated articles is publicly available [31]. In this paper we present an in-depth exploration into effective feature and system design for cybersecurity information and

event extraction. We believe that these contributions will enable future researchers to develop more targeted, rich, information extraction systems within the cybersecurity domain.

The rest of the paper is organized as follows. Section I gives an introduction to our work. Section II provides background knowledge and briefly discusses related work on event detection, especially in the cybersecurity domain. Section III describes our current cybersecurity event schema. Section IV discusses our document corpus and the annotation process and provides statistics about the results. Section V presents our current system for each subtask of a cybersecurity event information extraction system, with their experimental results shown in Section VI. The final section gives a conclusion and direction for future work.

II. BACKGROUND

A. Event Detection Terminologies

The following are definitions of common terms used in event detection research and in this paper. Some of these terms were introduced by the Text Analysis Conference event track in 2015 [20]. An event **trigger** (also sometimes called a **nugget**) is a word or phrase that represents or invokes the event. An event **argument** is a word or phrase representing a participant in an event, or a modifying aspect of the event (such as a temporal modification). Event nuggets and arguments can be annotated with **realis**, which captures if the event actually happened or not. Three values of realis are used: *actual*, to indicate a specific event did happen; *generic*, to refer to that type of event in kind, without referring to any specific events; and *other*, which refers to specific events that did not happen or complete. Finally, multiple mentions of an event that refer to the same core event are said to **corefer** with one another.

B. Cybersecurity Incidents Resources

With few exceptions, e.g., the 2018 SemEval task 8 [27], little previous work has been done on labeling cybersecurity events with a rich annotation scheme. Many online resources exist that summarize the frequency and associated trends of different cybersecurity events, e.g.,

- PrivacyRights[1] is a repository for data breach incident reports,
- Hackmageddon[2] is a website which collects public reports of cybersecurity incidents,
- Databreaches.net[3] is a website that collected databreach incidents,
- Cyberwire[4] is a cyber security-focused news service which provides daily briefing of cybersecurity news.

While these resources create reports that show the relative frequency of events but do not provide any comprehensive details about them. Some resources categorize cybersecurity events by event type, attack pattern, and type of malware.

[1] https://www.privacyrights.org/data-breaches
[2] https://www.hackmageddon.com/
[3] https://www.databreaches.net/
[4] https://thecyberwire.com/

This information is summarized in the document level, not the token (word) level, and is generally insufficient to train an automatically information extraction system. They might be useful for analyzing trends or predicting similar events.

C. Related Work

Event extraction has been a challenging task in language understanding systems for decades. Most previous work has focused on extracting information about common events involving a person or organization, such as birth or being arrested for a person and being created or merged for an organization. Many approaches have been used for this task, such as the features based models in state-of-the-art systems like [5], [9], [10], novel methods such as modified Convolutional Neural Networks [2], [22]–[24] and applied attention mechanism [14], [15], [25].

All of these systems were developed and evaluated as part of two significant IE programs/challenges: the Automatic Content Extraction program (e.g., ACE2005 [37]) and the NIST Text Analysis Conference Knowledge Base Population track (e.g., [20]) and used document collections which were annotated with the rich event schema [12], [13] representing life events for people and organizations.

There has been some recent work on extracting cybersecurity events. [29] discussed extracting cyberattack information from Chinese news articles by comparing different features modeled with a maximum entropy classifier. The system can detect event triggers and event arguments as subject and actor, but no information was given about its schema, such as event types and event arguments.

[21] presented a framework to analyze Twitter messages about cybersecurity and to issue timely threat alerts. They collected tweets that mentioned cybersecurity vulnerabilities terms and linked these to a semantic knowledge graph. [33] proposed an automatic, self-learned framework that can detect, geolocate, and categorize cybersecurity events in near-real time over the Twitter stream. [6] and [30] used social media as a crowdsourced sensor to monitor ongoing cyber-attacks, including DDoS attacks, data breaches, and account hijacking. [1] used social media text to detect mentions of Denial-of-Service attacks and establish when the attack started. They modeled the problem as a binary classification task and used a neural network approach. [38] also detected references to cybersecurity events from the Twitter data stream, but did not determine a specific type of cybersecurity attack.

[11] proposed an analysis that annotated documents in a collection of "advanced persistent threat" (APT) reports with attribute labels from the Malware Attribute Enumeration (MAEC) vocabularies. Their work was interested in identifying a malware-related token, its relation, and associated attribute labels. [26] proposed an approach to predict future cybersecurity events using *probabilistic soft logic* rules. They detected explicit mentions of events in a cyberattack kill chain using lexical matching and named entity recognition.

While [29] and [26] describe a schema and dataset similar to what we use here, their underlying schemas are not as

semantically rich or detailed. [1], [6], [21], [30], [33], [38] used keyword searching on Twitter data stream to detect or predict cybersecurity events. They differ from our work, which focuses on extracting the comprehensive information about cybersecurity events. [11] applies language understanding techniques to cybersecurity text but does not deal with event detection or modeling at all.

III. CYBERSECURITY EVENT SCHEMA

In this section, we provide an overview of the rich event schema presented by [32]. In providing this overview, we aim to make it clear how such schemas can be defined and applied to new situations.

This schema encodes five different cybersecurity events. These were chosen as they provided diversity, occurred frequently, were often mentioned in news articles, and are thought to have important consequences and impact. The five events types are:

- **Attack.Databreach**: an attacker compromises a system to exfiltrate data to sell or publish.
- **Attack.Phishing**: an attacker impersonates a trusted or benign entity to entice a victim to access a malicious website or download malicious attachments.
- **Attack.Ransom**: an attacker hacks into a system to encrypt data and demand a ransom payment in exchange for decrypting the data.
- **Discover.Vulnerability**: a security researcher or software company reveals the discovery of a software vulnerability.
- **Patch.Vulnerability**: a software company releases or describes an update to fix the known vulnerability.

These can be categorized as belonging to two more general super-types: Attack and Vulnerability-related cybersecurity events.

The cybersecurity event schema also defines arguments, which represent potential participating entities, attributes, role fillers set, and realis value. The current event argument catalog [32] includes 20 argument types: *Capabilities, CVE, Data, Device, File, GPE, Money, Number, Organization, Patch, Payment Method, Person, PII, Purpose, System/Software, Time, Malware, Website, Version*, and *Vulnerability*. When an argument is associated with an event, it will fill a semantic role in that event. For example, an argument that is a type of organization could fill either the *victim* or *attacker* role in an attack. The roles an event can have depends on the event's type and is summarized in Table II.

The number of event arguments and roles for each event is shown in Table III. As noted, this set of events can be seen as a set of prototypes that can be easily expanded to other types of cybersecurity events in the future. For example, Distributed Denial of Service (DDoS) attack can use the same event argument set of **Attack.Databreach** with an addition of a role *'Bot'* to be filled by an event argument of type *'Device'*.

IV. DATA COLLECTION

The data we use to build an IE system was collected from approximately 5,000 news articles published between 2017 and 2019 from the selected reading list of the Cyberwire[4] website. Each text article has XML-based metadata, such as its title, publication date, and URL.

A. Annotation Process

The Brat Rapid Annotation tool [34] was used to annotate data following a five-step process, as documented in our annotation guidelines, and briefly described below and in more detail in [32].

1) Identifying event nugget and its type: An event nugget is a noun, verb, noun phrase, or verb phrase that signals the presence of an event. If it is a phrase, it should as short as possible while still containing enough information to represent the event. The event nugget should be annotated whether any event arguments are found or not. The main difference between our cybersecurity events and events in previous domains is that a cybersecurity event often has multiple actions associated with it, as shown by the following examples.

> *E1:* Computer hackers $\textbf{\textit{have stolen}}_{Attack.Databreach}$ private files from the company.

> *E2:* WikiLeaks $\textbf{\textit{is posting}}_{Attack.Databreach}$ thousands of files Tuesday.

Both examples are annotated as **Attack.Databreach** nuggets, but represent two different actions associated with the event: obtaining the data and making the data public.

2) Labeling event arguments: Event arguments can be an entity or attribute that can be annotated with a type selected from a catalog of 20 argument types. The chosen one should be compatible with its nearest event type, as defined in Table II. If there is no annotated event nugget in the context (the target sentence and its preceding and following ones), the event argument is not annotated. The event arguments can be noun, noun phrase, or verb phrase of any length, as long as it has relevant information. However, the spans of two events (includes an event nugget and its arguments) cannot overlap. The challenge of annotating an event nugget and arguments is illustrated by these examples:

> *E3:* "This $\underline{vulnerability}_{Vulnerability}$ $\underline{allows\ an}$ $\underline{attacker\ to\ steal\ FTP\ accounts}_{Capabilities}$", $\text{Kim}_{Discoverer}$ $\textbf{\textit{noted}}_{Discover.Vulnerability}$.

> *E4:* "This $\underline{vulnerability}_{Vulnerability}$ allows $\underline{an}$ $\underline{attacker}_{Attacker}$ to $\textbf{\textit{steal}}_{Attack.Databreach}$ $\underline{FTP}$ $\underline{accounts}_{CompromisedData}$", $\underline{\text{Kim}}_{Discoverer}$ $\textbf{\textit{noted}}_{Discover.Vulnerability}$.

The event argument of type *'Capabilities'* in **Discover.Vulnerability** might be wrongly annotated as the event of **Attack.Databreach**. But we do not allow annotation when their text descriptions overlap, so the annotation given in *E4* is incorrect.

TABLE II

ARGUMENT TYPES AND ROLES FOR THE CYBERSECURITY EVENTS [32, REPRODUCED WITH PERMISSION] USED IN OUR IE SYSTEMS. FOR EXAMPLE, THE **ATTACK.DATABREACH** EVENT HAS A "VICTIM" ROLE, WHICH IS FILLED BY AN ARGUMENT OF TYPE "DEVICE." MULTIPLE ROLES OF THE SAME TYPE ARE SEPARATED BY SEMICOLONS.

Argument Types	Roles, Listed per Event				
	Attack.Databreach	**Attack.Phishing**	**Attack.Ransom**	**Discover.Vulnerability**	**Patch.Vulnerability**
Capabilities	AttackPattern	AttackPattern	AttackPattern	Capabilities	IssuesAddressed
CVE	–	–	–	CVE	CVE
Data	CompromisedData	TrustedEntity	–	–	–
Device	Victim	-	Victim	VulnerableSystem	VulnerableSystem
File	Tool	TrustedEntity; Tool	Tool	-	-
GPE	Place	Place	Place	–	–
Malware	Tool	Tool	Tool	–	–
Money	DamageAmount	DamageAmount	DamageAmount; RansomPrice	–	–
Number	NumberOfVictim; NumberOfData	NumberOfVictim	NumberOfVictim	–	–
Organization	Attacker; Victim	Attacker; Victim; TrustedEntity	Attacker; Victim	Discoverer; VulnerableSystemOwner	Releaser; VulnerableSystemOwner
Patch	–	–	–	–	Patch
PaymentMethod	–	–	PaymentMethod	–	–
Person	Attacker; Victim	Attacker; Victim; TrustedEntity	Attacker; Victim	Discoverer; VulnerableSystemOwner	Releaser; VulnerableSystemOwner
PII	CompromisedData	TrustedEntity	–	–	–
Purpose	Purpose	Purpose	–	–	–
System; Software	Victim	TrustedEntity	Victim	VulnerableSystem	VulnerableSystem
Time	Time	Time	Time	Time	Time
Version	–	–	–	VulnerableSystemVersion	PatchNumber; VulnerableSystemVersion
Vulnerability	–	–	–	Vulnerability	Vulnerability
Website	Victim	TrustedEntity; Tool	Victim	VulnerableSystem	VulnerableSystem

3) Linking an event nugget to event arguments: Each argument is annotated with a role it fills for the event defined by the nugget. Depending on their type, the cybersecurity events have between nine and eleven possible roles (see Table V). A given argument type may be able to fill several roles and a particular role may accept arguments of several different types. For example, a *'Person'* argument might fill the *'Attacker', 'Victim', 'TrustedEntity', 'Discoverer', 'Releaser', 'VulnerableSystemOwner'* roles (Table II shows more information.) An argument is constrained in only being able to fill a single role in its associated event, which makes the role assignment process more manageable. While this constraint is too strong in general (e.g., John fills two roles in *John hit himself*), we did not find it to be so in this domain.

4) Specifying event realis: Each event nugget must be annotated with a realis value that represents the event's truth value. There are three possible realis values: *Actual, Generic,* and *Other.* An *actual* realis value indicates that a specific instance event is believed to have happened.

> *Actual:* Facebook have admitted they ***were conned***$_{Attack.Phishing}$ out of an alleged $100 million in ***a phishing scam***$_{Attack.Phishing}$.

The *other* realis value is used to identify an event that has not actually occurred, such as a failed event, one set in the future, and hypothetical events that are part of a conditional statement.

> *Other:* Apple ***has not yet clarified***$_{Discover.Vulnerability}$ whether the latest versions of macOS and iOS ***are vulnerable***$_{Discover.Vulnerability}$.

A *generic* value indicates that the event is a general description of a class of events and does not describe a specific event instance is believed to have happened.

> *Generic:* ***Phishing***$_{Attack.Phishing}$ has grown to be one of the most serious threats to healthcare organizations.

5) Event coreference: Some events may be mentioned multiple times in a document and their mentions should be grouped into an *event hopper* [13]. The event mentions are coreference if:

- They have the same event type;
- They share one or more event arguments;
- Each shared event argument has the same role; and
- They have the same temporal and location scope.

In our system, cybersecurity events with different realis values can be coreferenced, which is similar to the event hopper defined in DEFT Rich ERE Annotation guidelines [13]. This decision was made to handle the fact that cybersecurity events often consist of multiple actions, such as parts of a "killchain." In those cases, it is possible that one of the actions in the chain has failed, as in the following example.

> *E5:* Library management ***refused to pay***$_{Attack.Ransom}$ the \$35,000 ***demanded as ransom***$_{Attack.Ransom}$.

Both event nuggets in *E5* trigger the same **Attack.Ransom** event and should be grouped in the same hopper even if they have different realis values. The first event nugget, *'refused to pay'* has the realis value as *'Other'* since the event did not actually happen while the second nugget, *'demanded as ransom'* has realis value *'Actual'*.

In some cases, there is disagreement about whether or not an event happened, as in the following.

> *E6:* "The headlines that say 2 million messages ***were leaked***$_{Attack.Databreach}$ on the internet are completely false," Meyers said.

> *E7:* "A malicious actor would only be able to ***access***$_{Attack.Databreach}$ a customer's voice recording if they managed to guess the password", he said.

Both *E6* and *E7* mention the same **Attack.Databreach** event. Even though *E6*'s realis is *'Actual'*, while *E7*'s is *'Other'*, both events are grouped in the same coreference set.

TABLE III
NUMBER OF ARGUMENTS AND ROLES DEFINED FOR EACH EVENT.

Event types	Arguments	Roles
Attack.Databreach	15	11
Attack.Phishing	14	9
Attack.Ransom	13	9
Discover.Vulnerability	10	9
Patch.Vulnerability	11	10

B. Statistics

[32] described an annotation process for 1,000 articles according to the above schema. The annotation was done by three experienced computer specialists with the final annotation chosen by a majority vote, and a reported Cohen's Kappa of 0.81. Table V gives the frequency breakdown of our annotation by event types. The highest number of event types found is 955 events with 1,564 instances of **Attack.Phishing** event. The **Discover.Vulnerability** has a lower number of events with the highest number of event nuggets, 2,122 instances. The average length of an event nugget is between 1.57 and 2.29 words. The average number of roles per event varies from two to three types. The statistics for event roles are also relevant to event arguments. Since an argument can fill just one role, the numerical data show that a complete set of event information cannot always be found in a single mention.

Table IV shows event argument type statistics. The number of event argument instances varies from 168 to 3,805. The least frequent argument type is *'GPE'* (Geopolitical Entity), which we attribute to the fact that a relevant *'GPE'* is often found in different sentences which can be relatively far from the event nugget, making it ineligible for assignment to the nugget.

The most frequent argument type is *'Person'*, with 3,805 mentions. Its popularity can be explained by the fact that this argument type is a possible filler for all of our event types. The average event argument length varies by type from 1.0 to 5.2 words, with the *'Capabilities'* type having the longest at 5.2. The *'Capabilities'* type characterizes an attacker's action or what vulnerability was exploited and is often impossible to describe in just a few words. For example, "downloading a dangerous attachment" is a *'Capabilities'* and fills the *'AttackPattern'* role in a **Attack.Phishing** event and "allow an attacker to achieve root shell" is a *'Capabilities'* argument that fills the *'Capabilities'* role in a **Discover.Vulnerability** event.

V. CYBEREVENT EXTRACTION TASKS

Our core contributions are the development and analysis of an information extraction system that can identify and extract information described by the above described rich schema and annotated data. The following tasks are the major elements in our model for extracting cyberattack events from a document:

- An **event nugget detection** system predicts whether a token is part of an event mention

TABLE IV
STATISTICS OF EVENT ARGUMENTS IN OUR ANNOTATED CORPUS SORTED BY THE NUMBER OF INSTANCES. (# IS NUMBER OF INSTANCES, AVG IS AVERAGE LENGTH OF EVENT ARGUMENT IN WORDS)

Arg Type	#	Avg	Arg Type	#	Avg	Arg Type	#	Avg	Arg Type	#	Avg
GPE	168	1.4	Money	413	1.6	Device	757	2.0	Capabilities	1,742	**5.2**
PaymentMethod	195	2.5	Version	463	1.9	Patch	772	2.1	System/Software	2,283	2.2
Number	260	1.4	Malware	484	1.7	Data	980	1.8	Vulnerability	2,554	2.4
CVE	317	**1.0**	Purpose	555	4.2	PII	1,076	2.1	Organization	3,511	1.8
Website	325	2.3	File	607	2.2	Time	1,403	2.0	Person	**3,836**	1.6

TABLE V
STATISTICS OF EVENT TYPES IN OUR ANNOTATION CORPUS

Event types	Number of events	Number of event nuggets	Avg nugget length (words)	Avg roles/ event
Attack.Databreach	916	1,780	1.83	3.16
Attack.Phishing	955	1,564	1.95	2.79
Attack.Ransom	944	1,585	2.29	2.68
Discover.Vulnerability	528	2,122	1.57	3.02
Patch.Vulnerability	560	1,419	1.66	2.90

- An **event argument detection** system predicts whether a token is part of an argument type or not.
- An **event realis identification** system identifies the truth value of the event, assigning a realis value to the event nugget only, not the event argument.
- An **event argument and role linking** system assigns an event argument to a semantic role, choosing among the options if more than one role can accept the argument. For example, an argument type of type *'Money'* can only fill the *'DamageAmount'* role in an **Attack.Databreach** or **Attack.Phishing** event, making the assignment task easy.
- An **event coreference resolution** system groups mentions of the same event in the same document.
- **Mapping events to a knowledge graph** produces output in the knowledge graph form. It performs additional cross-document entity clustering and links entities where appropriate to entities in the DBpedia and Wikidata knowledge graphs.

These systems are designed to replace the laborious human annotation process which is similar to other event detection research. There are some constraints which differ from those found in other event annotation efforts due to the nature of cybersecurity events (See section IV-A).

VI. EXPERIMENTAL RESULTS

In this paper, we have focused so far on describing the event schema and the annotation corpus that underlies our system. For additional details on the major tasks needed to extracting cybersecurity events, please see [32]. We developed these sub-systems using our annotation corpus which breaks the data into two data sets; 900 documents for training the system with 8-fold cross-validation and 100 documents for testing.

We used an evaluation metric from [7] and report scores in our experiments as the average from five runs.

The text of each document is processed to get linguistic and other features, e.g., part-of-speech, syntactic dependencies using Stanford CoreNLP [18], and extra named entities that are found by DBpedia Spotlight [3]. We also used Wikidata [36] to map entities to their fine-grained classes, such as software company, software version, or city.

A. Event Nugget Detection System

We form the module as a multi-class classification system using Bi-directional Long Short Term Memory (Bi-LSTM) with CRF as the output layer. The network architecture consists of four layers: embedding layers which are concatenated by word embeddings and other syntactic features, Bi-LSTM layer, a fully-connected layer, and the CRF as an output layer. The number of nodes in each layer is equal to half of the size of the previous layer.

Results. We conducted the experiments by using two types of word embedding techniques; context-free (Word2vec [19]) and context-dependent (BERT [4]). For Word2vec, we trained the word embeddings with size 100, called Domain-Word2vec, on the 5,000 cybersecurity news articles previously collected using a random initialization. We did not start from a pre-trained word embedding model because the cybersecurity text contains technical terms and our corpus is small (with 102K words), which cannot change the pre-trained vector (with 3M words) much. For BERT, we prepared the word vector from averaging Word-piece from BERT word vectors. The BERT word vectors are in the fourth-to-last layer, which gave the best performance on our validation data set. The results are shown in Table VI.

Considering the micro average of all events, Domain-Word2vec yielded the highest precision while BERT gave the best recall. We found that the system performs very well in identifying the **Attack.Ransom** event and both embeddings gave their lowest scores for the **Discover.Vulnerability** event. Our assessment showed that poor performance resulted from **Discover.Vulnerability** event nuggets were typically very common and general verbs like 'said', 'tell', and 'reported'. However, the overall average performance of the system is good, demonstrating that each event can be recognizable automatically and that our cybersecurity event types are well defined and annotated.

TABLE VI
SCORES OF EVENT NUGGET DETECTION SYSTEM BY EVENT TYPES

Event types	Domain-Word2vec			BERT		
	P	R	F_1	P	R	F_1
Attack.Databreach	85.61	72.70	78.63	75.20	82.29	78.58
Attack.Phishing	83.29	68.75	75.32	81.57	77.36	79.41
Attack.Ransom	83.87	83.57	**83.72**	84.74	80.17	**82.39**
Discover.Vulnerability	84.31	63.58	72.49	76.04	81.75	78.79
Patch.Vulnerability	89.68	67.30	76.89	84.66	79.17	81.82
micro avg	**85.00**	70.78	77.24	79.60	**80.28**	**79.94**

B. Event Argument Detection System

We implemented the event argument detection system using a multi-class classification approach. This system is invoked after the event nugget detection system has found a candidate set of event nuggets. Its neural network consists of five layers and has an architecture similar to that used in the event nugget detection system, except for the attention layer, which is inserted between the Bi-LSTM layer and the fully-connected layer, and the output layer, which has a different size.

Results. We ran experiments similar to those used in the event nugget detection system, with the results shown in Table VII. These demonstrate that BERT gave higher performance than Domain-Word2vec. The lowest performance is the event argument of type *'Capabilities'* for Domain-Word2vec, while BERT gave slightly higher performance. This is due to the length and its form of *'Capabilities,'* which has an average length of 5.2 words (See Table IV), one of the longest event argument type. Moreover, *'Capabilities'* can be considered as a composition of another event (See *E3* and *E4* in section IV-A). Another example of low performance is found in the event argument of type *'GPE,'* which has an F_1 of 58.55 from Domain-Word2vec, and 44.71 from BERT. We believe that this is the result of there being a relatively low number of instances in our training corpus compared to other event argument types (see Table IV). The result can likely be improved during annotation by increasing the context size, which will introduce more GPE annotation samples.

The highest performance for an event argument belongs to *'Money'* for both word embedding techniques. The other high-performance event arguments are *'CVE'* and *'Number'*. While the numbers of instances for these argument types are low, their mentions fall into a few well-defined patterns, allowing a named entity recognizer to more easily find them. In summary, the average performance of event argument detection is about 74.76, which is a high rate for classifying 20 catalogs of event argument types. Some event arguments have lower performance because of their complex structure. This can be improved in future work by acknowledging the overlapping of events.

C. Event Argument and Role Linking

Role assignment depends on the argument type and the type of event it participates in. The event argument and role linking system is invoked once we have the event nugget and arguments. We designed a neural network to assign roles to

TABLE VII
SCORES OF EVENT ARGUMENT DETECTION SYSTEM

Arg Type	Domain-Word2vec			BERT		
	P	R	F_1	P	R	F_1
Capabilities	52.35	45.78	**48.84**	45.22	59.05	**51.22**
CVE	86.85	89.26	88.04	85.55	86.30	85.92
Data	69.05	69.52	69.28	77.64	67.63	72.29
Device	60.73	48.55	53.96	65.16	63.95	64.55
File	77.24	67.39	71.98	78.99	74.77	76.82
GPE	72.39	49.15	58.55	76.03	31.67	**44.71**
Malware	74.31	49.55	59.46	68.67	51.33	58.75
Money	92.28	92.71	**92.49**	93.98	92.81	**93.39**
Number	75.40	67.11	71.02	87.38	83.56	85.43
Organization	76.99	64.56	70.23	75.05	84.22	79.37
Patch	81.46	81.22	81.34	81.70	80.40	81.04
PaymentMethod	80.14	78.37	79.25	78.32	83.74	80.94
Person	79.85	75.77	77.76	76.71	85.16	80.71
PII	77.79	80.25	79.00	78.71	83.66	81.11
Purpose	56.75	51.99	54.27	65.20	53.68	58.88
Product	67.34	56.80	61.62	67.25	71.45	69.29
Time	78.13	75.34	76.71	76.42	85.13	80.54
Version	65.28	52.24	58.03	69.87	66.53	68.16
Vulnerability	82.93	79.84	81.36	86.10	81.11	83.53
Website	45.64	50.61	**48.00**	54.91	67.98	60.75
micro avg	**73.82**	67.45	70.50	72.88	**76.74**	**74.76**

TABLE VIII
SCORES OF EVENT ARGUMENT AND ROLE LINKING SYSTEM. (VS MEANS VULNERABLESYSTEM.)

Role type	P	R	F_1	Role type	P	R	F_1
Attacker	0.83	0.77	0.80	TrustedEntity	0.75	0.77	0.76
Discoverer	0.83	0.87	0.85	VS	**0.94**	**0.96**	**0.95**
PatchNumber	**0.72**	**0.45**	**0.55**	VS_Owner	0.74	0.77	0.75
Releaser	0.82	0.88	0.84	VS_Version	0.54	0.75	0.62
Tool	0.71	0.92	0.80	Victim	0.87	0.89	0.88

the token sequences that are identified as event arguments. The input to the network is the event argument only. The same neural network is built for each event type. Each neural classifier consists of two layers: an embedding layer and a fully-connected layer with Softmax output.

Results. We tested the event argument and role linking system and reported the weighted average scores of all event types. The word embeddings used in the experiment is Domain-Word2vec. The results are given in Table VIII and show that the event argument and role linking systems perform well. The highest score is 0.95 for the *'VulnerableSystem'* role and the lowest score is 0.55 for the *'PatchNumber'* role. Each argument was assigned to a correct role, even if there are multiple candidates (see Table II). This demonstrates the utility of a domain-specific event schema with a well-considered, reasonable set of event arguments and roles, namely that it can yield effective information extraction components.

D. Event Realis Identification

We assign a realis value to the event nugget only, not the event argument. Hence, the input of the event realis identification system is only the token which is predicted as

TABLE IX
SCORES FOR IDENTIFYING EVENT REALIS ATTRIBUTES

Realis	P	R	F_1
Actual	0.83	0.86	0.85
Generic	0.65	0.63	0.64
Other	0.67	0.58	0.62

TABLE X
SCORES OF EVENT COREFERENCE SYSTEM

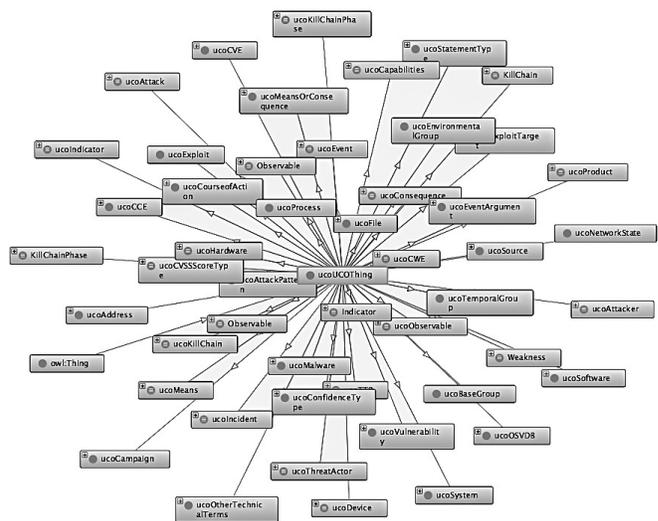| MUC	B^3	CEAFe	BLANC	Average
80.06	79.79	59.60	72.57	73.00

Fig. 1. Information extracted about cybersecurity events is represented as a knowledge graph in RDF using terms from the Unified Cybersecurity Ontology.

the event nugget. We break the realis identification into two steps. First, we predict the realis label of *Generic* or *Non-Generic*. Second, the Non-Generic cases are further analyzed to predict a label of *Actual* or *Other*. The neural networks are built for each step to classify the realis for the event nugget using an architecture with three layers: an embedding layer of event nugget and its context of size seven within the same sentence, a Bi-LSTM layer, and a fully-connected layer with Softmax output.

Results. We tested the system using Domain-Word2vec embeddings and report the scores in Table IX. There is no score of *'Non-Generic'* because all of the samples predicted to be *'Non-Generic'* were predicted again and reported as the realis labels of *'Actual'* and *'Other'*. Note that the Table shows that the scores of *'Actual'* is high while *'Generic'* and *'Other'* are much lower. This stems from the fact that we limited the context to text in the same sentence. In some cases, the evidence for an event having realis values of *'Generic'* and *'Other'* is found in a different sentence.

E. Event Coreference Resolution System

To find the hopper or coreference set of events, we use an agglomerative clustering approach. Each event mention pair is given a distance vector that measured the distance between their properties. Event pairs lower than a threshold are merged. The distance vector is comprised of weighting the properties nugget surface distance, realis difference, number of the redundant argument type, coreference of arguments, normalized number of sentences between nuggets, number of argument mismatch, and time difference. We ran a grid search to find the best threshold and the weight vector.

Results. We evaluated the system using the standard evaluation script V8.01 [28], which has also been used in other event coreference research [8], [16], [20]. The results are in Table X. Our system produced a good result compared to other systems' performances on the benchmark dataset [17].

F. Mapping Event Information to our Knowledge Graph

Our cyber-event extraction pipeline produces a set of JSON objects that represent the events found in a document collection along with their associated entities and role fillers. These are processed by a module that maps these into a knowledge graph using terms from the Unified Cybersecurity Ontology

[35] (UCO). Some of UCO's top-level types are shown in Figure 1.

UCO provides a common understanding of cybersecurity domain and unifies the most commonly used cybersecurity standards. Unlike existing independent and isolated cybersecurity ontologies, UCO has been mapped to publicly available ontologies and data models in the cybersecurity domain such as the OASIS STIX and NIST National Vulnerability Database schema models and hence offers more coverage. In addition to that, UCO is also mapped to concepts in general world knowledge sources to support diverse use cases.

Where possible, we link entities participating in a cybersecurity event to the corresponding entity in a knowledge graph of background knowledge, currently Wikidata, using their mentions and type. For example, the organization "Adobe" is mapped to the Wikidata entity *Q11463* and the DBpedia entity *Adobe_Systems*.

VII. CONCLUSION AND FUTURE WORK

We defined a new suite of cybersecurity event extraction tasks, demonstrated across five event types, their semantic roles, and the argument types that can fill them. Our focused work targets the key tasks in an event detection system: detecting event nuggets and arguments, predicting event realis and linking arguments and nuggets with roles. We developed our system and performed our analysis on a collection of 5,000 news articles discussing cybersecurity events, where 1,000 of them have been annotated with the rich event schema [32]. We developed an information extraction system trained using the annotated data and evaluated its performance. We show that neural methods with linguistic and word embedding features can extract cybersecurity event information with high accuracy.

Ongoing work is focused on re-arranging the semantic roles and arguments to new types of cybersecurity events, improving annotation by allowing overlapping of events extents, and deriving cross-document event coreference using the knowledge graph. There are multiple types of cybersecurity events that can be covered by assigning new roles using existing arguments. The event argument detection might be improved by allowing overlapped annotation of event extents. But it will need an additional process in event nugget detection. Moreover, we can find more information, such as evidence for cyberattack campaigns, advanced persistent threats, and impending attacks, that can be derived from events represented in the knowledge graph.

REFERENCES

[1] N. Chambers, B. Fry, and J. McMasters, "Detecting Denial-of-Service Attacks from Social Media Text: Applying NLP to Computer Security," Proc. Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018.

[2] Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao, "Event extraction via dynamic multi-pooling convolutional neural networks," In Proceedings of ACL, 2015, pp. 167176.

[3] J. Daiber, M. Jakob, C. Hokamp and P.N. Mendes,"Improving Efficiency and Accuracy in Multilingual Entity Extraction,",Proceedings of the 9th International Conference on Semantic Systems (I-Semantics), 2013.

[4] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805 (2018).

[5] H. Ji, and R. Grishman, "Refining event extraction through cross-document inference," In Proc. ACL-08, HLT, 2008, pp. 254262.

[6] R.P. Khandpur, T. Ji, S. Jan, G. Wang, C.T. Lu, and N. Ramakrishnan, "Crowdsourcing cybersecurity: Cyber attack detection using social media," In Proc. ACM Conf. on Information and Knowledge Management, 2017, pp. 1049-1057.

[7] Language Technologies Institute - Carnegie Mellon University, "Event Nugget Detection and Coreference Scoring v.27," Technical report, 2015.

[8] H. Lee, M. Recasens, A. Chang, M. Surdeanu, and D. Jurafsky, "Joint entity and event coreference resolution across documents," In Proc. Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, 2012, pp. 489-500.

[9] Q. Li, H. Ji, and L. Huang, "Joint event extraction via structured prediction with global features," In Proceedings of ACL, 2013, pp. 7382.

[10] S. Liao, and R. Grishman, "Using document level cross-event inference to improve event extraction," In Proceedings of ACL, pp. 789797 , 2010.

[11] S.K. Lim, A.O. Muis, W. Lu, and C.H. Ong, "Malwaretextdb: A database for annotated malware articles," In Proc. 55th Annual Meeting of the Association for Computational Linguistics, 2017, pp. 1557-1567.

[12] Linguistic Data Consortium. ACE (Automatic Content Extraction) English Annotation Guidelines for Events Version 5.4.3., 2005.

[13] Linguistic Data Consortium. DEFT Rich ERE Annotation Guidelines: Events v.2.6. Technical report, February, 2015.

[14] J. Liu, Y. Chen, K. Liu, and J. Zhao, "Event detection via gated multilingual attention mechanism," In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[15] S. Liu, Y. Chen, K. Liu,and J. Zhao, "Exploiting argument information to improve event detection via supervised attention mechanisms," In 55th Annual Meeting of the ACL, Vancouver, 2017. pp. 17891798.

[16] Z. Liu, T. Mitamura, and E. Hovy, "Graph-Based Decoding for Event Sequencing and Coreference Resolution," arXiv preprint arXiv:1806.05099 (2018).

[17] J. Lu, and V. Ng, "Event Coreference Resolution: A Survey of Two Decades of Research," In IJCAI, 2018, pp. 5479-5486.

[18] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP Natural Language Processing Toolkit," In 52nd Annual Meeting of the ACL: System Demonstrations, 2014, pp. 55-60.

[19] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," In Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'13), 2013.

[20] T. Mitamura, , Z. Liu, and E.H. Hovy, "Overview of TAC KBP 2015 Event Nugget Track," In TAC, November, 2015.

[21] S. Mittal, P. K. Das, V. Mulwad, A. Joshi, and T. Finin, "Cybertwitter: Using twitter to generate alerts for cybersecurity threats and vulnerabilities," In Proc. IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining, IEEE Press, 2016, pp. 860-867.

[22] T.H. Nguyen, and R. Grishman, "Event detection and domain adaptation with convolutional neural networks," In Proc. 53rd Annual Meeting of the Association for Computational Linguistics and the 7th Int. Joint Conf. on Natural Language Processing (Volume 2: Short Papers), 2015, pp. 365-371.

[23] T.H. Nguyen, and R. Grishman, "Modeling skip-grams for event detection with convolutional neural networks," In Proc. Conf. on Empirical Methods in Natural Language Processing, 2016, pp. 886-891.

[24] T.H. Nguyen, and R. Grishman, "Graph convolutional networks with argument-aware pooling for event detection," In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[25] J.W. Orr, P. Tadepalli, and X. Fern, "Event detection with neural networks: A rigorous empirical evaluation," arXiv preprint arXiv:1808.08504, 2018.

[26] I. Perera, J. Hwang, K. Bayas, B. Dorr and Y. Wilks,"Cyberattack Prediction Through Public Text Analysis and Mini-Theories,", IEEE International Conference on Big Data (Big Data), 2018, pp. 3001-3010.

[27] P. Phandi, A. Silva and W. Lu, SemEval-2018 Task 8: Semantic Extraction from CybersecUrity REports using Natural Language Processing (SecureNLP), in Proc. 12th International Workshop on Semantic Evaluation, Association for Computational Linguistics, 2018.

[28] S. Pradhan, X. Luo, M. Recasens, E. Hovy, V. Ng, and M. Strube,"Scoring Coreference Partitions of Predicted Mentions: A Reference Implementation," Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, (Volume 2: Short Papers), June 2014, pp.30-35.

[29] X. Qiu, and X. Lin, "Feature Representation Models for Cyber Attack Event Extraction," 2016 IEEE/WIC/ACM International Conference on Web Intelligence Workshops (WIW), 2016, pp. 29-32.

[30] A. Ritter, E. Wright, W. Casey, and T. Mitchell, "Weakly supervised extraction of computer security events from twitter," In Proc. 24th Int. Conf. on World Wide Web, 2015, pp. 896-905.

[31] T. Satyapanich, "CASIE Repository," https://github.com/Ebiquity/-CASIE, 2019.

[32] T. Satyapanich, "Modeling and extracting information about cybersecurity events from text". Ph.D. dissertation, University of Maryland, Baltimore County, December 2019.

[33] Q.L. Sceller, E.B. Karbab, M. Debbabi, and F. Iqbal, "Sonar: Automatic detection of cyber security events over the twitter stream," In Proceedings of the 12th International Conference on Availability, Reliability and Security, ACM, August, 2017, pp. 23.

[34] P. Stenetorp, S. Pyysalo, G. Topi, T. Ohta, S. Ananiadou, and J. Tsujii, "BRAT: a web-based tool for NLP-assisted text annotation," In Proc. Demonstrations at the 13th Conf. of the European Chapter of the ACL, 2012, pp. 102-107.

[35] Z. Syed, A. Padia, T. Finin, L. Mathews, and A. Joshi, UCO: A unified cybersecurity ontology. In *Workshop on AI for Cyber Security*, 195–202. AAAI. 2016.

[36] D. Vrandei, and M. Krtzsch, "Wikidata: a free collaborative knowledge base," 2014.

[37] C. Walker, S. Strassel, J. Medero, and K. Maeda, ACE 2005 multilingual training corpus. Linguistic Data Consortium, 2006.

[38] S. Yagcioglu, M. S. Seyfioglu, B. Citamak, B. Bardak, S. Guldamlasioglu, A. Yuksel, and E. I. Tatli, "Detecting Cybersecurity Events from Noisy Short Text," arXiv preprint arXiv:1904.05054 (2019).