UCO: A Unified Cybersecurity Ontology

Zareen Syed, Ankur Padia, Tim Finin, Lisa Mathews and Anupam Joshi
University of Maryland, Baltimore County, Baltimore, MD 21250
{zsyed, ankurpadia, finin, math1, joshi}@umbc.edu

Abstract

In this paper we describe the Unified Cybersecurity Ontology (UCO) that is intended to support information integration and cyber situational awareness in cybersecurity systems. The ontology incorporates and integrates heterogeneous data and knowledge schemas from different cybersecurity systems and most commonly used cybersecurity standards for information sharing and exchange. The UCO ontology has also been mapped to a number of existing cybersecurity ontologies as well as concepts in the Linked Open Data cloud (Berners-Lee, Bizer, and Heath 2009). Similar to DBpedia (Auer et al. 2007) which serves as the core for general knowledge in Linked Open Data cloud, we envision UCO to serve as the core for cybersecurity domain, which would evolve and grow with the passage of time with additional cybersecurity data sets as they become available. We also present a prototype system and concrete use cases supported by the UCO ontology. To the best of our knowledge, this is the first cybersecurity ontology that has been mapped to general world ontologies to support broader and diverse security use cases. We compare the resulting ontology with previous efforts, discuss its strengths and limitations, and describe potential future work directions.

Introduction

Cybersecurity data and information is usually generated by different tools, sensors and systems expressed using different standards and formats, published by different sources and is often scattered as isolated pieces of information. Furthermore, cybersecurity data is available in structured, semi-structured and unstructured forms from both, internal sources i.e. within the organization, and external sources i.e. outside the organization. Unifying such scattered information will provide better visibility and situational awareness to cybersecurity analysts. Also, such integration can support deep investigations and help transitioning from reactive approach to a more proactive and eventually a predictive approach.

Semantic Web technologies provide representation languages to build a common framework that allows data to be shared, integrated and reused across applications, enterprises as well as community boundaries. Languages, such as RDF and OWL, represent the semantics of an entity as a set of things or concepts rather than strings of words. They provide rich constructs to represent information that is not only machine readable, but also machine understandable, thus facilitating semantic integration and sharing of information from heterogeneous sources. Languages like OWL have well defined constructs to map classes and instances present in the internal knowledge base to corresponding classes and instances in external knowledge bases. This mapping exposes a larger pool of knowledge and helps in providing a more complete picture and situational awareness.

Figure 1: Things vs. Strings. "Strings" are ambiguous and can refer to different concepts in the real world. "Things" are precise and reference unique concepts using unique identifiers, such as Web URIs.

Semantic Web technologies represent real world entities as concepts rather than strings, as strings are lexical and ambiguous. Concepts are associated with a globally unique identifier called URI. For example, the string "Georgia" may refer to "Georgia state" in the United States or "Georgia country" (Figure 1). Moreover, concepts can be associated with attributes and can have relations with other concepts. These attributes and relations can be used to build up a context for the concept. An entity like "Georgia country" can have "longitude" and "latitude" as attributes, which provide

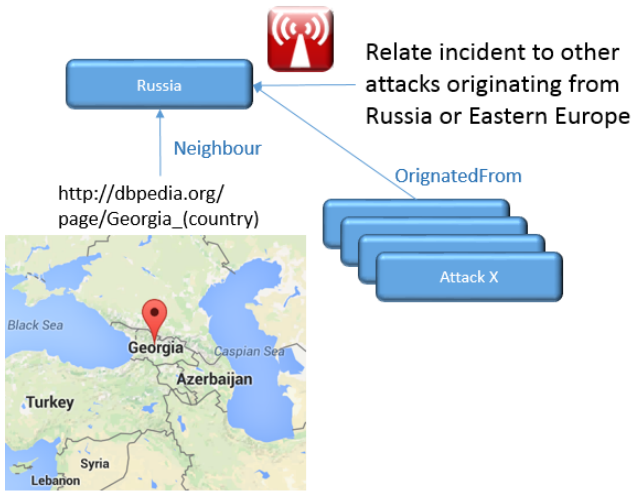

Figure 2: Semantic Relations enable supporting complex security use cases, for example, if "Georgia_(country)" has "neighbor" relation with "Russia" it may raise more alarms if several past incidents originated from Russia.

information about its location on the map and its neighboring countries. Moreover, such information can be used to derive inferences about possible source of attack. For example, if an incident originates from "Georgia country" and it's neighboring country is Russia, then it may raise more alarms if many cybersecurity attacks have originated from Russia in the past (Figure 2). Furthermore, these relations can help in connecting the dots and relating incidents with similar incidents to gain insight into the source and motivation of the attack.

Semantic technologies are used by big data companies like Google, Microsoft, Facebook and Apple (Domingue, Fensel, and Hendler 2011) for information sharing and interoperability and supporting high level functions like analyzing queries, providing semantic search and answering questions. In order to achieve situational awareness, cybersecurity systems need to transition to produce and consume semantic information about likely entities, relations, actions, events, intentions and plans.

We have developed Unified Cybersecurity Ontology (UCO) as an effort to help evolve the cybersecurity standards from a syntactic representation to a more semantic representation. We see several contributions that our work has to offer:

1. UCO ontology provides a common understanding of cybersecurity domain and unifies most commonly used cybersecurity standards.
2. Compared to existing cybersecurity ontologies which have been developed independently, UCO has been mapped to a number of existing publicly available cybersecurity ontologies to promote ontology sharing, integration and reuse. UCO serves as a backbone for linking cybersecurity ontologies.
3. UCO maps concepts to general world knowledge sources i.e. Linked Open Data cloud to support diverse use cases.
4. We describe important use cases that can be supported by unifying cybersecurity data with existing general world knowledge through the UCO ontology.
5. We have generated a catalog of cybersecurity standards that is available online[1].

This paper is organized as follows: In section 2 we briefly introduce RDF and a subset of OWL language, OWL DL. In section 3 we outline our approach for ontology construction and describe the UCO ontology along with other related ontologies. Section 4 presents the design and implementation of a demonstration system with real world cybersecurity data that uses the UCO ontology to support a number of use cases. We review related work in section 5 and conclude with a summary for future work in section 6.

Preliminaries

Resource Description Framework (RDF)

The Resource Description Framework[2] is a W3C standard to represent knowledge as a semantic graph in which the nodes represent entities, concepts or literal values and the arcs represent relations. Thus, we can think of a knowledge bases as a collection of triples with a subject, predicate and object. The subject is usually the entity that is being represented. The predicate represents an attribute or a relation of the subject and is used to associate with an object. The object can be a literal or a resource. Typically, each of the resource is identified with a URI. Example of an RDF triple can be $< John, studiesAt, School >$.

OWL DL

OWL DL[3], a sublanguage of OWL, which is based on Description Logics is a tractable fragment of First Order Logic and is used for knowledge representation. OWL DL is a W3C standard to represent knowledge and is more expressive compared to RDF. Formal definitions of some of the constructs used in DL are shown in Table 1. Further details on the constructs can be obtained from (Baader 2003).

Approach

Our approach to support cyber situational awareness has been through the development of a core cybersecurity ontology that facilitates data sharing across different formats and standards and allows reasoning to infer new information. We have surveyed, reviewed and cataloged existing cybersecurity standards and ontologies and selected the most common and widely used standards to incorporate in UCO ontology. In this section, we first briefly outline the advantages of using Semantic Web languages and describe the UCO ontology along with its design considerations. We describe the feasibility to support diverse and complex use cases by linking cybersecurity information to external knowledge sources in the next section.

[1] http://tinyurl.com/ptqkzpq
[2] http://www.w3.org/RDF/
[3] http://www.w3.org/TR/owl-guide/

Table 1: Syntax and Semantics of Description Logic constructors

Name	Syntax	Semantics	Symbol
Top	$\top$	$\Delta^{\mathcal{I}}$	$\mathcal{AL}$
Bottom	$\bot$	ϕ	$\mathcal{AL}$
Intersection	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$	$\mathcal{AL}$
Union	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$	$\mathcal{U}$
Negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus D^{\mathcal{I}}$	$\mathcal{C}$
Value restriction	$\forall R.C$	$\{a \in \Delta^{\mathcal{I}} \mid \forall b.\ (a,b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\}$	$\mathcal{AL}$
Existential quant.	$\exists R.C$	$\{a \in \Delta^{\mathcal{I}} \mid \forall b.\ (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}$	$\mathcal{E}$
Nominal	I	$I^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ with $\lvert I^{\mathcal{I}}\rvert = 1$	$\mathcal{O}$
Qualified Number restriction (less than)	$\leq nR.C$	$\{a \in \Delta^{\mathcal{I}} \mid \mid \{\ \forall b \in \Delta^{\mathcal{I}} \mid (a,b) \in \mathcal{R}^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\ \} \mid \leq n\ \}$	$\mathcal{Q}$
Qualified Number restriction (equal than)	$= nR.C$	$\{a \in \Delta^{\mathcal{I}} \mid \mid \{\ \forall b \in \Delta^{\mathcal{I}} \mid (a,b) \in \mathcal{R}^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\ \} \mid = n\ \}$	$\mathcal{Q}$
Qualified Number restriction (greater than)	$\geq nR.C$	$\{a \in \Delta^{\mathcal{I}} \mid \mid \{\ \forall b \in \Delta^{\mathcal{I}} \mid (a,b) \in \mathcal{R}^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\ \} \mid \geq n\ \}$	$\mathcal{Q}$
Role Hierarchy	$R_1 \sqsubseteq R_2$	$\{\ (a, b) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (a, b) \in R_1^{\mathcal{I}} \rightarrow (a, b) \in R_2^{\mathcal{I}}\ \}$	$\mathcal{H}$
Role Inverse	R^{-}	$\{\ (b, a) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (a, b) \in R^{\mathcal{I}}\ \}$	$\mathcal{I}$
Role Composition	$R_1 \circ R_2$	$\{\ (a, c) \mid \exists b.\ (a, b) \in R_1^{\mathcal{I}} \wedge (b, c) \in R_2^{\mathcal{I}}\ \}$	$\mathcal{R}$

Advantages of Semantic Web Languages

RDF is a directed graph and unambiguous compared to XML, which is tree based and has multiple representation for the same information. As RDF and OWL have formal semantics grounded in First Order Logic they are more preferable for dealing with security situations. RDF and OWL have a decentralized philosophy which allows incremental building of knowledge, and its sharing and reuse. For example, properties can be defined separately from classes (unlike Object Oriented Programming). OWL facilitates information integration by providing rich semantic constructs for schema mapping such as Sub Class, Sub Property, Equivalent Class, Equivalent Property, Same As, Union Of, Intersection Of etc. to represent complex facts (Table 1). Furthermore, OWL has powerful off-the-shelf reasoners, which enable detecting inconsistencies during data sharing. For example, if there is a constraint for two classes, "Malware" and "Virus", to be disjoint and the data sets imported from different sources mention the same software to be both a Malware and Virus, then in such cases the reasoner will infer an inconsistency. Semantic Web technologies are well established and there are powerful reasoners available both as Open Source Software and Commercial products.

Unified Cybersecurity Ontology (UCO)

The Unified Cybersecurity Ontology (UCO) is an extension to Intrusion Detection System ontology (IDS) (Undercoffer et al. 2004) developed earlier by our group to describe events related to cybersecurity. Our group has been working on a number of projects that focus on individual components of a unified cybersecurity framework to analyze different data streams and assert facts in a triple store (Undercoffer et al. 2004; More et al. 2012; Mulwad et al. 2011). The UCO ontology is essential for unifying information from heterogeneous sources and supporting reasoning and rule writing. The ontology supports reasoning and inferring new information from existing information. The ontology also supports capturing specialized knowledge of a cybersecurity analyst which can be expressed using ontology classes and terms as well as rules. Rules are used to infer new information which cannot be captured with an OWL reasoner. Figure 3 demonstrates a generic rule to infer an attack and alert the host. The rule uses terms from UCO ontology to connect information within the organization with external information available on the web. The rule states that if the web text description consists of some *vulnerability terms*, mentions some *security exploit*, has text mentioning a certain *product* (with some specific version) and some *process* which is being executed, which in turn is also logged by the scanner, and there is an opening up of an out-bound port; then there is a possibility of an attack on the host system with *Means* and *Consequences* mentioned in the ontology.

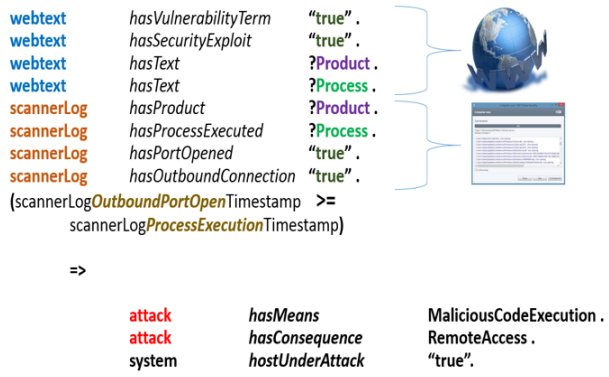

Figure 3: The UCO ontology facilitates writing generic rules and combining evidence from multiple sources.

UCO ontology provides a common understanding of cybersecurity domain. Among all cybersecurity standards and formats, STIX (Structured Threat Information eXpression) (Barnum 2012) is the most comprehensive effort to unify cybersecurity information sharing and enables extensions by incorporating vocabulary from several other standards. However, in STIX the information is represented in XML and therefore cannot support reasoning which is supported by UCO. We have created Unified Cybersecurity Ontol-

Table 2: Statistics for UCO and related ontologies

	CCE	CVE	CVSS	UCO	Total
Axiom	11	21	197	633	862
Class Count	1	3	35	106	145
Object Property Count	0	2	32	59	93
Data Property Count	5	6	3	45	59
Individual Count	0	0	23	7	30
Equivalent classes	0	0	3	16	19
DL Expressivity	AL	ALUHO(D)	ALUHOQ(D)	ALCROIQ(D)	

Table 3: Statistics for existing ontologies mapped to UCO

	CPC	CY	CYB_C	DM	KC	MC	STIX	STO
Axiom	7915	296	117	2	63	2	8808	60
Class Count	1219	21	10	1	12	1	1303	15
Object Property Count	6	22	5	0	5	0	114	21
Data Property Count	3	19	13	0	0	0	47	0
Individual Count	10	70	25	0	0	0	91	0
Equivalent classes	2	4	2	0	26	0	17	10
DL Expressivity	ALCHO(D)	ALUOQ(D)	ALUOQ(D)	AL	ALCIQ	AL	ALCHOIQ(D)	ALE

ogy as a semantic version of STIX. In addition to mapping to STIX, UCO has also been extended with a number of relevant cybersecurity standards, vocabularies and ontologies such as CVE[4], CCE[5], CVSS[6], CAPEC[7], CYBOX[8], KillChain[9] and STUCCO[10]. To support diverse use cases, UCO ontology has been mapped to general world knowledge available through Google's knowledge graph, DBpedia knowledge base (Auer et al. 2007), Yago knowledge base (Suchanek, Kasneci, and Weikum 2008) etc. Linking to these knowledge sources provides access to large number of datasets for different domains (e.g. geonames) as well as terms in different languages (e.g. Russian).

Below we describe the list of important classes present in UCO ontology:

1. **Means**: This class describes various methods of executing an attack and consists of sub-classes like *BufferOverFlow*, *SynFlood*, *LogicExploit*, *Tcp-PortScan* etc., which can further consist of their own sub-classes. The *Means* class maps to *TTP* field in STIX which characterizes specific details of observed or potential attacker Tactics, Techniques and Procedures.
2. **Consequences**: This class describes the possible outcomes of an attack. It consists of sub-classes like *DenialOfService*, *LossOfConfiguration*, *PrivilegeEscalation*, *UnauthUser*, etc. It maps to *Observables* in STIX.
3. **Attack**: This class characterizes a cyber threat attack and is mapped to *Incident* in STIX.
4. **Attacker**: This class represents identification or characterization of the adversary and is mapped to *ThreatActor* in STIX.
5. **AttackPattern**: Attack Patterns are descriptions of common methods for exploiting software providing the attackers perspective and guidance on ways to mitigate their effect. An example of attack pattern is *Phishing*.
6. **Exploit**: This class characterizes description of an individual exploit and maps to *ExploitType* in STIX schema.
7. **Exploit Target**: Exploit Targets are vulnerabilities or weaknesses in software, systems, networks or configurations that are targeted for exploitation by the *TTP* (cyber threat adversary Tactic, Technique or Procedure).

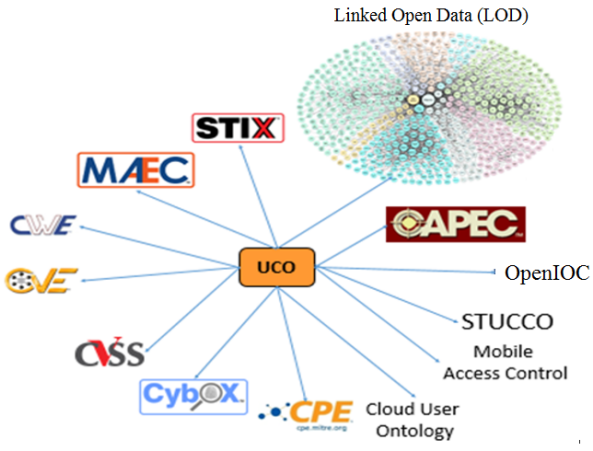

Figure 4: UCO ontology serves as the core for cybersecurity Linked Open Data (LOD) cloud

[4] https://cve.mitre.org/
[5] https://cce.mitre.org/
[6] https://www.first.org/cvss
[7] https://capec.mitre.org/
[8] https://cyboxproject.github.io/
[9] http://vistology.com/ont/STIX/killchain.owl
[10] https://github.com/stucco/ontology/blob/master/stucco_schema.json

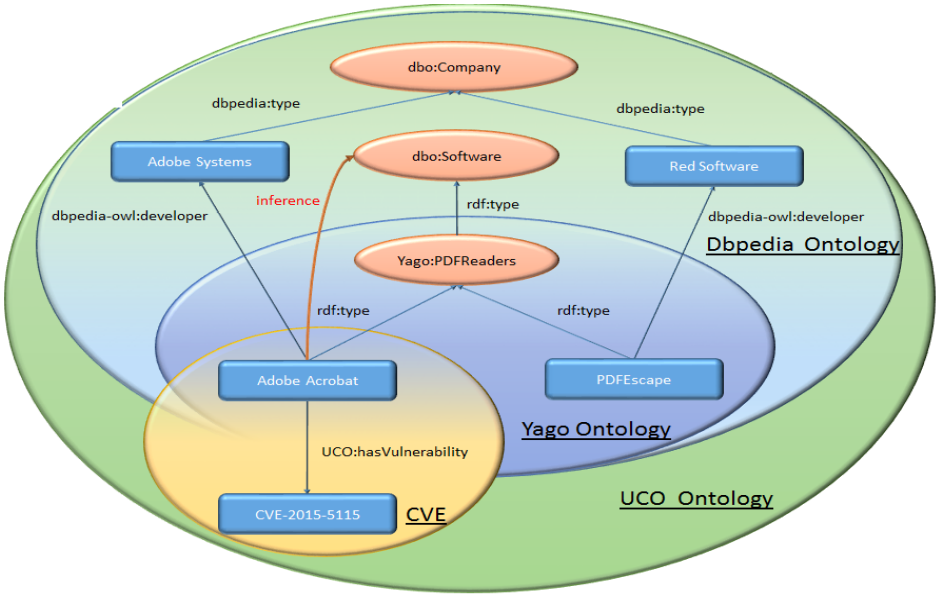

Figure 5: Advanced use cases can be supported using mappings between UCO and general world ontologies in linked open data cloud

8. **Indicator**: A cyber threat indicator is made up of a pattern identifying certain observable conditions as well as contextual information about the patterns meaning, how and when it should be acted on, etc. This class is mapped to *IndicatorType* in STIX schema and *Indicator* class in CAPEC ontology.

The ontology also has a number of classes to characterize Processes, Network State, Files, System, Hardware and Kill Chain along with Kill Chain Phases. The statistics for UCO and related ontologies are given in Table 2 & 3 where ontologies in Table 2 are developed by our group. These ontologies are independent and do not have any overlapping classes. However, to generate a connected graph, the UCO ontology has a few classes representing parent class of each of the other ontology. Such a design allows easy maintenance of the ontology as different ontologies are loosely coupled and hence each ontology can evolve independently. As shown in Table 2, CCE contains a class and 5 data type properties like description, references, platform etc. CVSS ontology contains 35 classes and includes classes like base group, environmental group and temporal group, which are represented as the combination of other classes. Similar with CVE which contains 3 classes and 8 properties available from the XML schema. As compared to other ontologies, we designed UCO to represent and considerably extend STIX framework with additional classes and defined relations among them. For example *IPAddress*, *Software*, *WebBrowser* are a few classes with *WebBrowser* being the subclass of *Software*. Moreover, there are 16 classes in the ontology which are defined as the combination of other classes. For example, *Product* is represented as the union of *Software* and *Hardware*. Figure 4 shows UCO ontology serving as the core ontology for linking with other cybersecurity ontologies and LOD cloud. To facilitate data integration from multiple freely available knowledge base, we mapped UCO to Linked Open Data (LOD) cloud. Such an extension allows an analysts to fetch data from multiple freely available data sources with different schema but represented using semantic web technologies. An example of a mapping from UCO to DBpedia is shown below:

$$<uco : acrobat\_reader\ owl : sameAs\ dbr : Adobe\_Acrobat>$$

Here, "uco:" is the namespace used for Unified Cybersecurity Ontology and the mapping asserts that the Adobe Acrobat from DBpedia (represented with "dbr" namespace) is same as the acrobat reader present in UCO ontology.

Prototype System Design and Use Cases

We have designed a system that uses STUCCO extractors[11] to extract entities from the National Vulnerability Database (NVD) XML file. We implemented code to generate $< subject, predicate, object >$ triples from the XML file. We defined mappings between entities obtained from NVD data to corresponding entities in DBpedia. The triples and the mappings were loaded on to the Fuseki server as it supports federated queries to integrate data from multiple sources and it supports reasoning. Any triple store with these capabilities can be used to host the triples and integrate data.

Ontology Use Cases

In Figure 5 we show several advanced use cases that can be supported using mappings between UCO and general world ontologies that cannot be supported by individual ontologies alone. Below we describe each use-case with corresponding SPARQL queries. We also show a snippet of results obtained by executing the same query over a sample NVD data set

[11] https://github.com/stucco/extractors

that was loaded into the Fuseki triple store. For each use cases "db" represents the namespace for DBpedia resources.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX yago: <http://dbpedia.org/class/yago/>

prefix uco: <http://ffrdc.ebiquity.umbc.edu/ns/ontology/>

select distinct  ?observable ?reader {
  SERVICE <http://dbpedia.org/sparql> {
    ?reader rdf:type yago:PDFReaders .
  }
  ?vulnerability uco:hasObservable ?observable .
  ?observable   uco:hasSoftware ?reader .
}
order by ?reader
```

observable	reader
uco:CVE-2002-2435	db:Adobe_Acrobat
uco:CVE-2002-2436	db:Adobe_Acrobat
uco:CVE-2002-2437	db:Adobe_Acrobat

Figure 6: SPARQL Query and results for Use Case 1

Use-Case #1: Vulnerabilities associated with PDF Readers: An organization or a security analyst may be interested in finding the kind of vulnerabilities associated with a specific type of software. The CVE entries only mention software, however one can also retrieve the associated type of software if these software are linked to external knowledge sources such as Google's knowledge graph or DBpedia. For example, from CVE we have the information that *Adobe Acrobat* has a certain vulnerability identified with CVE entry reference *CVE-2015-5115*. Mapping *Adobe Acrobat* instance to the corresponding instance in DBpedia and YAGO resources will provide additional information that it is a type of *Yago:PDFReaders*. This mapping with enable answering queries asking for vulnerabilities associated with a specific category of software, such as PDF readers. The SPARQL query shown in Figure 6 demonstrates this use-case.

Use-Case #2: Vulnerabilities associated with products from a given company: Another interesting use-case is to explore vulnerabilities associated with products from a given company. Again by mapping software instances to external knowledge sources, one can find the name of the company which developed the software. The following SPARQL query, shown in Figure 7, retrieves vulnerabilities for products along with information about the source company.

Use-Case #3: Suggest similar software to given software: After knowing information about a certain vulnerability a security analyst may be interested in finding alternate software that doesn't have the given vulnerability. For example in case of a PDF readers the SPARQL query, shown in Figure 8, retrieves software of the same type filtering out "acrobat" software.

Use-Case #4: Assess impact of changing vendors: In case an organization is interested in changing vendors, they

```
prefix dbp: <http://dbpedia.org/property/>
prefix uco:  <http://ffrdc.ebiquity.umbc.edu/ns/ontology/>

select distinct ?product ?company ?vulnerability
where {
    SERVICE <http://dbpedia.org/sparql> {
        ?product dbp:developer ?company
    }
    ?vulnerability uco:hasObservable ?observable .
    ?blankNode uco:hasSoftware ?product .
}
order by ?product
```

product	company	vulnerability
db:Adobe_Acrobat	db:Adobe_Systems	uco:CVE-2002-2435
db:Adobe_Acrobat	db:Adobe_Systems	uco:CVE-2002-2436
db:Adobe_Acrobat	db:Adobe_Systems	uco:CVE-2002-2437

Figure 7: SPARQL Query and results for Use Case 2

```
prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX yago: <http://dbpedia.org/class/yago/>
prefix uco:  <http://ffrdc.ebiquity.umbc.edu/ns/ontology/>

select ?similarSoftware ?product where {

    SERVICE <http://dbpedia.org/sparql> {
        ?similarSoftware rdf:type yago:PDFReaders .
    }

FILTER (!regex(str(?similarSoftware), 'acrobat', 'i'))
}
```

similarSoftware
db:STDU_Viewer
db:Preview_(Mac_OS)
db:Pdfescape
db:Adobe_Digital_Editions
db:Adobe_Creative_Suite

Figure 8: SPARQL Query and Results for Use Case 3

```
prefix dbp: <http://dbpedia.org/property/>
prefix uco:  <http://ffrdc.ebiquity.umbc.edu/ns/ontology/>

select ?company (count(?vulnerability) as ?vulnerability_count) where {
    SERVICE <http://dbpedia.org/sparql> {
        ?software dbp:developer ?company
    }

    ?vulnerability uco:hasObservable ?observable .
    ?observable uco:hasSoftware ?software .
}
group by ?company
```

company	vulnerability_count
db:Opera_Software	864
db:Adobe_Systems	74

Figure 9: SPARQL Query and Results for Use Case 4

can assess the impact of the vendor by using the SPARQL query shown in Figure 9 which creates a summary of vulnerability counts associated with the products from different vendors.

Related Work

STIX (Barnum 2012) is the most comprehensive standard to unify cybersecurity information sharing and enables extensions by incorporating vocabulary from several other standards. Most cybersecurity systems use STIX representation expressed in XML. A major limitation of XML is that it cannot support reasoning. One of the earliest efforts for developing ontologies to support reasoning for cybersecurity was by our group in 2003 (Pinkston et al. 2003; Undercoffer et al. 2004). Undercoffer et al. implemented a target centric ontology for the domain of intrusion detection composed of 23 classes and 190 properties. (More et al. 2012) from our group extended this IDS ontology to incorporate and integrate cybersecurity related information from heterogeneous sources. The UCO ontology is a further extension and enhancement of the IDS ontology. It represents and maps publicly available standards and ontologies in the cybersecurity domain. (Ulicny et al. 2014) created a STIX ontology based on the STIX schema along with a number of related ontologies. We have defined mappings between UCO and STIX ontology classes. (Iannacone et al. 2015) developed STUCCO ontology for integrating different structured and un-structured data sources along with data extractors. STUCCO ontology is composed of 15 entity types and 115 properties and is defined using JSON-schema. We translated STUCCO from JSON to OWL in order to map it to UCO ontology. (Vaibhav Khadilkar and Thuraisingham) developed CPE ontology for Common Platform Enumerations in National Vulnerability Database. The ontology is described in the report however it is not publicly available for download and hence we could not map it to UCO. A number of ontologies have been developed for specialized domains within cybersecurity such as ontology for insider threats in finance domain (Kul and Upadhyaya 2015), ontology for network security attack (Simmonds, Sandilands, and van Ekert 2004; Chan et al. 2015) and a cloud security ontology (Amit Hendre and Joshi 2014). UCO ontology can serve as a backbone to link these specialized ontologies.

Conclusion and Future Work Directions

The UCO ontology provides a common understanding of cybersecurity domain and unifies most commonly used cybersecurity standards. Unlike existing independent and isolated cybersecurity ontologies, UCO has been mapped to publicly available ontologies in the cybersecurity domain 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. To the best of our knowledge this is the first such effort in the area of cybersecurity ontologies to unify cybersecurity information with general world knowledge about entities and relations. We presented different use cases that demonstrate the utility and value of UCO ontology in supporting diverse security scenarios. We briefly discuss promising future work directions below.

Temporal Representation and Reasoning: Cybersecurity data and information may have a temporal component, for example timestamps associated with files, system logs and network events etc. The current version of UCO ontology uses a very basic representation of time where time is represented as a data property associated with classes that represent events. A number of frameworks and representations have been proposed in research such as OWL-Time (Hobbs and Pan 2004) and time-entry (Pan and Hobbs 2004) which provide vocabularies for stating facts about temporal instants and intervals. In the future, we plan to extend UCO ontology to represent time instances and intervals so that it can support temporal reasoning.

Modeling Uncertainty and Confidence: Ontology design requires crisp logic i.e. any sentences in these languages such as asserted facts, domain knowledge, or reasoning results must be either true or false and nothing in between. Real world domains contain uncertain knowledge because of incomplete or partial information that is true only to a certain degree. Probability theory is a natural choice for dealing with this kind of uncertainty. There is a large body of literature on fuzzy logic and probabilistic reasoning. More recent developments seek to combine First Order Logic with probabilistic models, such as the work on Markov Logic Networks (Richardson and Domingos 2006) and Bayesian Logic (Milch et al. 2007). Future work directions include reviewing different approaches, identifying and analyzing shortcomings and encountered challenges followed by the choice of suitable representation to extend UCO ontology.

Cybersecurity Information Extraction from Unstructured Data: Cybersecurity vulnerabilities are typically identified and published publicly but response has always been slow in covering up these vulnerabilities because there is no automatic mechanism to understand and process this unstructured text published on the web. There is a strong need for systems that can automatically analyze unstructured text and extract vulnerability entities and concepts from various non-traditional unstructured data sources such as cybersecurity blogs, security bulletins and hackers forums. This information extraction task will help expediting the process of understanding and realizing the vulnerabilities and thus making systems secure at faster rate. There is some initial work in (Mulwad et al. 2011; Joshi, Lal, and Finin 2013; Jones et al. 2015). The UCO ontology can benefit these approaches by guiding the extraction process and also in checking consistency of the extracted facts.

Acknowledgements

The research described in this paper was supported by a seed grant from MITRE.

References

Amit Hendre, T. F., and Joshi, K. P. 2014. Ontology describing Cloud data threats, security controls and compliance standards. http://ebiquity.umbc.edu/resource/html/id/361/

Cloud-Security-and-Compliance-Ontology. [Online; accessed 21-October-2015].

Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2007. *Dbpedia: A nucleus for a web of open data*. Springer.

Baader, F. 2003. *The description logic handbook: theory, implementation, and applications*. Cambridge university press.

Barnum, S. 2012. Standardizing cyber threat intelligence information with the structured threat information expression (stix). *MITRE Corporation* 11.

Berners-Lee, T.; Bizer, C.; and Heath, T. 2009. Linked data-the story so far. *International Journal on Semantic Web and Information Systems* 5(3):1–22.

Chan, P.; Theron, J.; van Heerden, R.; and Leenen, L. 2015. An ontological knowledge base for cyber network attack planning. In *Iccws 2015-The Proceedings of the 10th International Conference on Cyber Warfare and Security*, 69. Academic Conferences Limited.

Domingue, J.; Fensel, D.; and Hendler, J. A. 2011. Introduction to the semantic web technologies. *Handbook of Semantic Web Technologies* 1–41.

Hobbs, J. R., and Pan, F. 2004. An ontology of time for the semantic web. *ACM Transactions on Asian Language Information Processing (TALIP)* 3(1):66–85.

Iannacone, M.; Bohn, S.; Nakamura, G.; Gerth, J.; Huffer, K.; Bridges, R.; Ferragut, E.; and Goodall, J. 2015. Developing an ontology for cyber security knowledge graphs. In *Proceedings of the 10th Annual Cyber and Information Security Research Conference*, 12. ACM.

Jones, C. L.; Bridges, R. A.; Huffer, K.; and Goodall, J. 2015. Towards a relation extraction framework for cybersecurity concepts. *arXiv preprint arXiv:1504.04317*.

Joshi, A.; Lal, R.; and Finin, T. 2013. Extracting cybersecurity related linked data from text. In *Semantic Computing (ICSC), 2013 IEEE Seventh International Conference on*, 252–259. IEEE.

Kul, G., and Upadhyaya, S. 2015. A preliminary cyber ontology for insider threats in the financial sector. In *Proceedings of the 7th ACM CCS International Workshop on Managing Insider Security Threats*, 75–78. ACM.

Milch, B.; Marthi, B.; Russell, S.; Sontag, D.; Ong, D. L.; and Kolobov, A. 2007. 1 blog: Probabilistic models with unknown objects. *Statistical relational learning* 373.

More, S.; Matthews, M.; Joshi, A.; and Finin, T. 2012. A knowledge-based approach to intrusion detection modeling. In *IEEE Symposium on Security and Privacy Workshops (SPW)*, 75–81. IEEE.

Mulwad, V.; Li, W.; Joshi, A.; Finin, T.; and Viswanathan, K. 2011. Extracting information about security vulnerabilities from web text. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*, volume 3, 257–260. IEEE.

Pan, F., and Hobbs, J. R. 2004. Time in owl-s. In *Proceedings of the AAAI Spring Symposium on Semantic Web Services*, 29–36.

Pinkston, J.; Undercoffer, J.; Joshi, A.; and Finin, T. 2003. A target-centric ontology for intrusion detection. *University of Maryland, Baltimore County Department of Computer Science and Electrical Engineering*.

Richardson, M., and Domingos, P. 2006. Markov logic networks. *Machine learning* 62(1-2):107–136.

Simmonds, A.; Sandilands, P.; and van Ekert, L. 2004. An ontology for network security attacks. In *Applied Computing*. Springer. 317–323.

Suchanek, F. M.; Kasneci, G.; and Weikum, G. 2008. Yago: A large ontology from wikipedia and wordnet. *Web Semantics: Science, Services and Agents on the World Wide Web* 6(3):203–217.

Ulicny, B. E.; Moskal, J. J.; Kokar, M. M.; Abe, K.; and Smith, J. K. 2014. Inference and ontologies. In *Cyber Defense and Situational Awareness*. Springer. 167–199.

Undercoffer, J.; Pinkston, J.; Joshi, A.; and Finin, T. 2004. A target-centric ontology for intrusion detection. In *18th International Joint Conference on Artificial Intelligence*, 9–15.

Vaibhav Khadilkar, J. R., and Thuraisingham, B. Semantic web implementation scheme for national vulnerability database (common platform enumeration data). Technical report, The University of Texas at Dallas.