Extracting Cybersecurity Related Linked Data from Text

Arnav Joshi, Ravendar Lal, Tim Finin and Anupam Joshi
Computer Science and Electrical Engineering
University of Maryland, Baltimore County, Baltimore, MD 21250 USA
{arnavj1, rlal1, finin, joshi}@umbc.edu

***Abstract*—The Web is typically our first source of information about new software vulnerabilities, exploits and cyber-attacks. Information is found in semi-structured vulnerability databases as well as in text from security bulletins, news reports, cybersecurity blogs and Internet chat rooms. It can be useful to cybersecurity systems if there is a way to recognize and extract relevant information and represent it as easily shared and integrated semantic data. We describe such an automatic framework that generates and publishes a RDF linked data representation of cybersecurity concepts and vulnerability descriptions extracted from the National Vulnerability Database and from text sources. A CRF-based system is used to identify cybersecurity-related entities, concepts and relations in text, which are then represented using custom ontologies for the cybersecurity domain and also mapped to objects in the DBpedia knowledge base. The resulting cybersecurity linked data collection can be used for many purposes, including automating early vulnerability identification, mitigation and prevention efforts.**

***Index Terms*—cybersecurity, linked data, information extraction, ontology**

I. Introduction

Cybersecurity is a critical concern as society has become highly interconnected and reliant on a global system of computers, communication networks and software systems. Cyber crime is more professional with the emergence of increasingly powerful methods of intrusion and exploits. For example, cyber criminals targeted users of Skype, Facebook and Windows using multiple blackhole exploits in late 2012 [1]. Many systems are under threat from vulnerabilities that are known and publicly documented. One reason for this is that these systems are not patched on a regular basis. While information about known vulnerabilities and patches for them is publicly available online, much of it is provided as text that is suitable for security experts, but not easily understood or directly usable by automated security systems.

One of the best public resources of security information is the National Vulnerability Database (NVD) and its associated components, including the Common Vulnerabilities and Exposures (CVE) and Common Weakness Enumeration (CWE), and Product Dictionary (CPE) datasets[1]. These resources list vulnerabilities and exposures, categorize them by type and severity, provide common names and identifiers, include links to patches and other information and have details as short text descriptions. Significant amounts of key information, however, even in such detailed descriptions, remain only in unstructured text, such as the systems that are likely to be affected, the operating systems environment for which the attack can occur, the versions of products affected, and the relationships between these entities. Vulnerabilities are also mentioned in various security bulletins and blogs, which typically are narrative descriptions that include the above mentioned relationships, though do not include any structured or semi-structured data. Collaborating and expressing these sources of information in a structured, semantic, machine-understandable format can help machines deal with possible “zero-day” attacks.

We describe an information extraction framework to extract cybersecurity-relevant entities, terms and concepts from the NVD and from unstructured text. These extracted concepts are then mapped and linked to related resources on the Web using an OWL ontology language [2] and represented as RDF linked open data [3]. Such a publicly available linked open data resource will help organizations uncover knowledge from multiple sources of cybersecurity-related data on the Web and support systems that automatically ingest, reason over and use the data to provide better cybersecurity.

II. Background and Previous Work

Our approach combines two aspects of the problem. The first is *extracting* relevant information about new security vulnerabilities, attacks and events from text. The second is *representing and integrating* this information along with data extracted from the the National Security Vulnerability Database as a linked data resource using custom ontologies in the Semantic Web languages RDF and OWL.

A. Information Extraction

Several repositories and security advisory sources address security changes and threat trends that might affect the overall security of a computer system. These sources can be used in a variety of ways to enhance the process of detection of an attack. NVD is a U.S. government repository of standards based vulnerability management data represented using the Security Content Automation Protocol (SCAP) [4]. Information sources such as the NVD and IBM XFORCE[2] provide XML feeds that report vulnerabilities with varying degrees of detail.

[1] See http://nvd.nist.gov/, http://cve.mitre.org/, http://cwe.mitre.org/ and http://nvd.nist.gov/cpe.cfm

[2] http://xforce.iss.net/

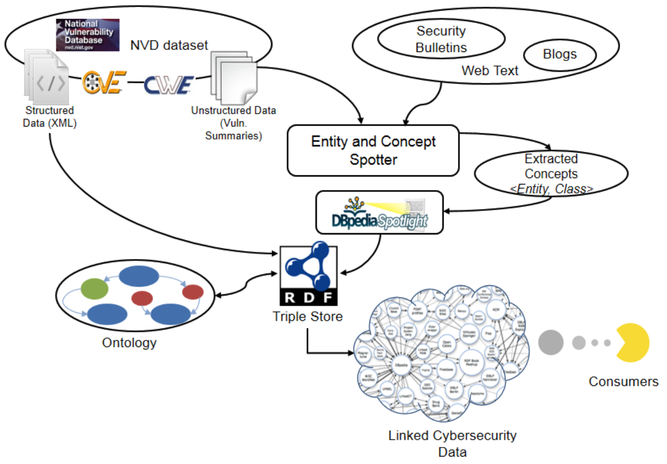
National Vulnerability Database
NVD dataset
Structured Data (XML)
CVE
CWE
Unstructured Data (Vuln. Summaries)
Security Bulletins
Blogs
Web Text
Entity and Concept Spotter
Extracted Concepts <Entity, Class>
DBpedia Spotlight
RDF
Triple Store
Ontology
Linked Cybersecurity Data
Consumers

Fig. 1. System architecture for extracting linked cybersecurity data from text

To the best of our knowledge, these repositories consolidate information present across multiple data sources, though are manually monitored.

These dictionaries not only contain redundant or overlapping information, but also miss out on important concepts such as the means and consequence associated with attacks and the versioning of a software product. Similar information is available in cybersecurity blogs such as *Krebs On Security*[3] and the *Metasploit Blog*[4], but their content is unstructured text, which can lead to an information overload, especially during threat analysis of a system. Furthermore, analyzing and integrating multiple textual resources can become a cumbersome task for system administrators. Extracting actionable content from these informal sources and representing it as a linked RDF data can enhance distribution of security information and the discoverability of security-related concepts.

More et al. [5], for example, demonstrate effective reasoning over such a semantically rich data for a situation aware intrusion detection system. Their framework requires a condensed source of web resources that provide meaningful information about the threat, and data sources that provide entities that map well into the ontology. Our approach provides automation to generate and update such a linked data resource that can be used to inform advanced intrusion detection and mitigation systems.

Mulwad et al. [6] describe a prototype system that analyzed relevant text snippets from the Web to generate assertions about vulnerabilities, attacks and threats. The system extracted concepts of interest using an SVM classifier and queried Wikitology [7] – a knowledge base of entities from Wikipedia, Yago [8] and Freebase [9]. The classification mechanism and the spotted concepts were limited to the identification of two classes: the *means* and the *consequence* of an attack. We adopted an approach that uses a Conditional Random Field (CRF) algorithm trained with ground truth annotations [10] to identify and classify mentions of entities and concepts that goes beyond their simple approach in terms of precision and recall.

The quality of the concepts extracted from free text largely depends on the method applied for concept spotting. More et al. [5] used OpenCalais [11], an information extraction system designed to recognize general entities such as people, places and organizations. Because of its orientation toward general coverage, it was unable to identify many of the entities and concepts important for cybersecurity. Similar experiments were run on the NERD information extraction framework [12], which failed to identify relevant technical jargon from the given piece of security-related text. These annotation tools are designed to capture information based on a custom ontology which models people, places and organizations. The Stanford Named Entity Recognition [13] also does not identify key cybersecurity concepts without proper feature filtering. Our approach introduces a cybersecurity entity and concept spotter that was primarily trained to identify entities (e.g., software products and operating systems) and concepts (e.g., denial of service and buffer overflow) which are related to computer security, threats and vulnerabilities in software products.

Khadilkar et al. [14] demonstrated the concept of using a semantic model to facilitate information representation and describe an ontology for the National Vulnerability Database. The ontology modeled information for software products and generic security concepts, though is unable to characterize and capture information from unstructured sources of information. Undercoffer et al. [15] specify an ontological model for categorizing computer attacks that used taxonomic characteristics of an intrusion to be limited to specific classes and attributes centered on the target of an attack. Our framework consolidates information across different knowledge bases and carries out concept-spotting for entities of interest, that can initiate characterization and understanding of the overall nature of the attack.

B. Linked Data

Linked data [16] enables publishing structured, machine-readable interpretation of heterogeneous sources of information. As defined by Bizer et al. [3], it is “a set of best practices for publishing and connecting structured data on the Web.” It focuses on interconnecting data and resources on the Web by defining relations between ontologies, schemas and/or directly linking the published data to other existing resource on the Web.

This approach can be leveraged to the cybersecurity domain by building an RDF data store for vulnerabilities, severity metrics, affected products and any remedial information. Relevant information about these concepts from other sources on the Web can be interlinked. With NVD data represented as RDF linked data, the task of finding all vulnerabilities pertaining to a single product version is reduced to the task of traversing the product-vulnerability dependency graph.

Additional contextual information obtained through establishing meaningful semantic links can help consolidate avail-

[3] http://krebsonsecurity.com/
[4] http://www.metasploit.com/

able information regarding a security threat. Moreover, the data representation for this interlinking will be in a structured, machine-readable format enabling faster, automated data consumption. The linked data resource can help improve the discoverability of data through the use of SPARQL [17] queries, SPARQL endpoints and resolvable URIs. It also helps in use cases such as distinguishing relevant vulnerabilities based on a product term or version. Such an interlinked corpus of data will enable stakeholders to share security-related information in a single resource, create business intelligence, support automated decision making systems and thereby speedup the exchange and digestion of information across different organizations.

III. System Architecture

Figure 1 shows the organization of our system, which is divided into three major components.

1) A CRF-based cybersecurity entity and concept spotter that identifies relevant concepts and entities from text
2) An ontology-based RDF triple generator that generates triples based on extracted information provided by the entity and concept spotter
3) A link generator that uses DBpedia Spotlight [18] to link extracted entities and concepts to DBpedia resources and aligns them with our cybersecurity-specific vocabulary.

In the following sections, these components will be described in detail.

A. Cybersecurity Entity and Concept Spotter

In order to extract relevant information from text, we developed a entity and concept spotter that identifies important entities and concepts in a given piece of text. This was done using general implementation of conditional random field (CRF) algorithm provided by Stanford named entity recognizer using a set of features for proper identification of concepts from the input text. We analyzed several cybersecurity-related blogs, security bulletins and CVE descriptions and identified a set of key classes that are relevant in terms of data representation of a vulnerability. We identified the following seven classes of relevance:

1) Software (e.g. Microsoft .NET Framework 3.5)
    a) Operating_System (e.g. Ubuntu 10.4)
2) Network_Terms (e.g. SSL, IP Address, HTTP)
3) Attack
    a) Means: Way to attack (e.g. Buffer overflow)
    b) Consequences: Final result of an attack (e.g. Denial of Service)
4) File_Name (e.g. index.php)
5) Hardware (e.g. IBM Mainframe B152)
6) NER_Modifier: This always follows Software or OS and helps in identifying software version information.
7) Other_Technical_Terms: Technical terms that cannot be classified in any of the above mentioned classes.

Each of these classes was chosen to represent key aspects in identification and characterization of the attack. The following described classes are most notable. *Network Terms* was identified as an important class since most of the attacks are using network technology these days. Thus it is important to extract relevant terms in text so that information regarding networks can be identified. The idea behind modeling the *Attack* class came from the work of Undercoffer et al. [15]. An Attack can be Further classified as a *Means*, which helps to identify a method of an attack, or as a *Consequence* that describes the final result of an attack. For example, "buffer overflow" is considered to be an instance of a *Means*, since it is not an attacker's final goal, but merely a step to achieve a desired consequence, such as a "denial of service."

Whether a phrase is considered to be an instance of a *Means* or *Consequence* is not always clear in a given text. We instructed annotators to use their discretion during annotation. When it was difficult to decide between them for a phrase, it was tagged as an *Attack Class*. In analyzing the gold standard annotation data we found that the inter-annotator agreement for these two subclasses was lower than all of the other classes. In this experiment, we took a random data sample from our corpus and asked two annotators to annotate the data for four classes (*Software Products*, *Operating System*, *Means* and *Consequences*). We found the agreement between the annotators to be over 90% for Software Products and Operating System. For Consequences, the agreement was 75%, while for Means it was 52%.

The *NER_Modifier* class also deserves some explanation. Understanding the version or versions of a software product being discussed is an important fact. In the text,

> "This vulnerability is present in Adobe Acrobat X *and earlier versions*..."

the phrase *"and earlier versions"* indicates that all Adobe Acrobat versions before version 10 are also vulnerable to the threat. These words hold key information about other versions that are vulnerable. The *NER_Modifier* class identifies these terms. It was observed that such terms were generally described immediately before or after a *Software* term or an *Operating_System* term. Identifying these pieces of text leverages the identification of product versions that may be susceptible to the vulnerability, though are not documented accordingly.

Based on these classes, our extraction framework was trained using the Stanford NER [19], a CRF based named entity recognition framework that is pre-trained to identify entities such as people, places and organizations. It includes a large feature set that can be customized to train a general implementation of a CRF model. We chose a training dataset consisting of over 30 security blogs, 240 CVE descriptions and 80 official security bulletins from Microsoft and Adobe. The data corpus [20] was manually annotated by twelve Computer Science graduate students, who had a fair understanding of cybersecurity related terms, concepts and technical jargon. We developed a custom application to simplify the annotation process using the BRAT rapid annotation framework [21], [22].

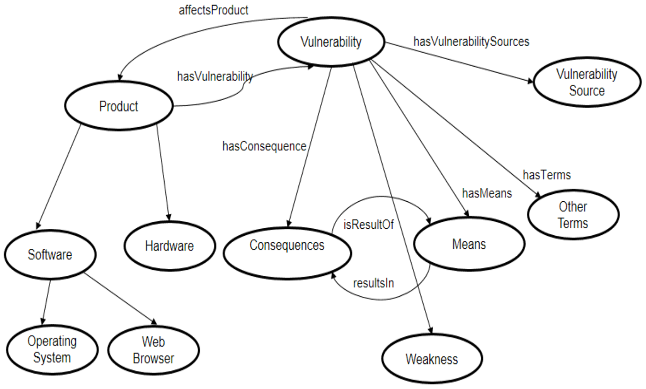

Fig. 2. A high level sketch of the IDS ontology

Feature Set Engineering: Feature set selection is a critical task in training a NER system. Though the Stanford NER provides an extensive selection of applicable features, filtering a subset that can capture all the relevant information pertaining to the cybersecurity domain is a tedious task. Feature selection is important, as applying all of the available features to the training and test data will not only slow down the annotation process, but also diminish the quality of results. Feature selection for our cybersecurity entities and concept spotter engine was carried out manually by analyzing the text and checking which features would be suitable. We selected a set of features that performed well for our analysis. The features that were used to train this system are: *useTaggySequences, useNGrams, usePrev, useNext, maxNGramLeng, useWordPairs* and *gazette*. A detailed discussion on our cybersecurity entity and concept spotter can be found in Lal et al. [23].

B. The IDS Ontology

We use the IDS ontology[5], partially depicted in Figure 2, to represent concepts and entities that are relevant to the cybersecurity domain. This vocabulary was originally developed by Undercoffer et al. [15], further enhanced by More et al. and this effort [5]. The ontology is expected to continue to evolve to cover additional concepts. We extended the ontology to provide model relations that capture the NVD schema structure and the security exploit concepts extracted by the NER. The new key classes defined in the ontology, specific to the entities which are part of the NVD dataset include *Vulnerability*, *Product* and *Weakness*.

Vulnerability: A vulnerability is an important class in the ontology, as each entry in the NVD is identified and documented based on a CVE number. The CVE number is a unique identifier for a vulnerability description provided by MITRE, on an incremental basis for each year. All information related to a particular identified vulnerability is associated with the CVE ID, including the list of affected products (identified based on their unique Common Platform Enumeration (CPE) name, the Web resources where it was first documented, and the severity metrics. The *Vulnerability* class hence is defined to have corresponding relationships with all other classes which are used to model entities that are part of an NVD entry.

Product: The *Product* class models the hardware and software products that are affected by a vulnerability. The *Software* subclass is further classified as an *Operating System* or *Web Browser* to correctly classify operating systems and Web browsers, apart from a generic "application" tag. The affected product information described in an NVD entry is limited to a list of product names described for a particular vulnerability. This information is incorporated using the CPE format, which includes version granularity. The *affectsProduct* relationship models a one-to-many mapping between the vulnerability identifier and the list of affected products. Additional information about the affected products in the NVD entry is extracted using our cybersecurity entity and concept spotter.

Weakness: An NVD entry contains a unique Common Weakness Enumeration (CWE) identifier that classifies a vulnerability based on a hierarchy of attack classes modeled to generalize different attack signatures. For example, *Cross-side scripting (XSS)* is a subclass of *Injection* which is a subclass of the *Invalid Input*. The severity score for a CVE ID is derived on parameters specified for the corresponding CWE ID. The *Weakness* class is thus included to extract more information regarding the metrics used to score the vulnerability's severity, by which the means for addressing a threat will be refined and enhanced. In addition, we define classes for each concept that is spotted by our classifier such as *Network Terms* (IP address, HTTP), *Means* (Buffer Overflow), and *Consequence* (Denial of Service).

All the concepts from the IDS vocabulary are aligned with concepts identified in the DBpedia ontology, by assigning a relevant DBpedia resource, thereby resolving the ambiguity of entities mapped in our ontology.

C. RDF Representation of NVD

Applying semantic web technologies to represent the data provided by the NVD dataset is useful for semantic analysis of vulnerabilities and exploits. However, correlating this data to the existing concepts on the Web and reasoning over such a corpus is a vital task to avail this information for different applications, front-end services and data consumers (e.g., security practitioners and system administrators). Semantics allow machine interpretation of links and relations between different properties of a vulnerability. Interlinking leads to an integrated and well-connected data corpus, available via an endpoint for advanced applications such as a semantic search and vulnerability statistics.

The NVD provides XML feeds for vulnerabilities that are published in a particular year. The NVD datasets are updated immediately with raw information whenever a new vulnerability is reported to the CVE repository, and iterated to a valid, confirmed source after analysis.

Our RDF-generation platform ingests XML feeds from the NVD dataset and generates RDF triples via an Extensible

[5] http://ebiquity.umbc.edu/IDSv2.0.1.owl

Stylesheet Language Transformation and the Jena RDF API [24]. The system includes primary attributes included directly in the NVD schema, as well as advanced properties fetched from the sources described in the former. For example, a NVD entry contains the CWE ID for the weakness class it belongs to. The CWE schema includes attributes such as the *Access Vector*, *Access Complexity* and *Authentication*. These attributes are used to calculate the severity score for a threat. It is observed that the vulnerabilities with the same combination of these features, take place under the same context or running environment.

D. Linked Cybersecurity Data

Establishing the relations between security exploit terms and identifiers that uniquely identify these concepts is essential to data integration. The objective to actually link instances and concepts with resources on the Web is a challenging aspect.

After RDF instances are generated from the properties provided by the NVD schema, the link generation component of our framework connects the security concepts extracted from the vulnerability descriptions to the existing deferenceable resources on the Web. Each NVD entry mentions a short summary of the vulnerability description, which is essentially unstructured text. This module annotates security-related terms from the vulnerability description and maps them to corresponding DBpedia resources using DBpedia Spotlight, an annotation tool for finding mentions of DBpedia resources in free text. DBpedia Spotlight provides flexibility to configure annotations to specific use cases, through quality metrics such as topical pertinence and disambiguation confidence. Binding the DBpedia references to the identified security concepts will enhance association of our linked data resource with other instances in the Linked Open Data cloud.

Entities with valid (contextual) resources in DBpedia are annotated based on adequate tuning of the confidence and support metrics. After experimentation with these parameters over our dataset, we selected a confidence of 0.3 and a support of 20 for generating DBpedia links for a specific vulnerability descriptions.

The annotations and subsequent linkages provided by DBpedia Spotlight are not final, or complete. For a given piece of text, the DBpedia Spotlight API returns the sets of annotated terms and corresponding DBpedia resources. However, the annotation does not provide the corresponding class from the DBpedia ontology that the resource belongs to. We employ our cybersecurity entity and concept spotter to map the security exploit concepts to appropriate classes from the IDS vocabulary. The NVD descriptions (vuln summary) are passed through the concept spotter, that identifies relevant terms, assigns a class label, and returns a set of `<Concept, Class>` tuples for the description. The *Concept* terms are then passed through DBpedia Spotlight. The annotated terms from DBpedia Spotlight were matched against the entities identified by our system using a string comparison. The corresponding DBpedia resource for the matched concept is assigned a class value, based on the *Concept, Class* pairs. These resources are then mapped with an appropriate object property from the IDS vocabulary. The choice for DBpedia Spotlight as a link generation tool, and the precision of the concept extraction and linking component are described in detail in Joshi et al. [25].

The IDS vocabulary models key aspects of a cyber attack which are not represented precisely in the DBpedia ontology. For example, the terms *Buffer Overflow* and *Denial of Service* are aptly represented as “Means” and “Consequence” respectively in the IDS vocabulary. These concepts are highly specific to a domain and hence not modeled in the DBpedia ontology.

Figure 3 shows a sample NVD entry which specifies the CVE identifier for the vulnerability description, together with the list of affected products with Common Platform Enumeration (CPE) names, the Weakness identifier, and the source where the vulnerability was documented. We extract information from this data to generate machine-understandable assertions in RDF, as shown in Figure 4. We use the IDS ontology to interpret key security concepts such as the vulnerability sources and severity metrics. Besides modeling semi-structured information, our framework extracts relevant DBpedia resources from the text description such as *Arbitrary_code_execution* and maps them to appropriate relationships (*hasConsequences*) from the IDS vocabulary.

Based on the relationships established with the linked concepts, we can retrieve vulnerabilities and attack descriptions pertaining to a specific product version, those affected by a specific means (*Buffer Overflow*), or those attacks that are carried out under the same operating environment. We can query over such a knowledge base via SPARQL queries to avail statistics on vulnerability trends, and can view the past history associated with a vulnerability or a particular software product. A triple store of such condensed information facilitates for a rich linked data resource, that can be used for semantic analysis of vulnerabilities.

The NVD datasets provide an RSS data feed on all recent CVE vulnerabilities. These immediate data sources can be represented as machine-understandable assertions as shown above. Such RDF assertions can be added to the triple store, and can help in applications such as a situation aware intrusion detection system that can consume linked data to generate rules and alerts on possible threats. In the future, we plan to extend the concept spotting system into an information extraction framework that is not limited to the NVD dataset and its auxiliaries. The proposed system will extract concepts from free text, find relationships between entities spotted in the text, make assertions about them based on a specific heuristic and publish it to the linked cybersecurity data resource.

IV. System Evaluation and Challenges

The focus in this paper has been on the problem of extracting cybersecurity concepts, entities and relations and generating linked data representations of them. In order to generate a quality linked data resource that captures all relevant security information from within a text description, the cybersecurity

```
<?xml version=“1.0” encoding=“UTF-8”?>
<nvd xmlns:vuln=“http://scap.nist.gov/schema/vulnerability/0.4”
xmlns:cvss=“http://scap.nist.gov/schema/cvss-v2/0.2”>
<entry id=“CVE-2012-0150”>
<vuln:vulnerable-software-list>
<vuln:product>cpe:/o:microsoft:windows_vista::sp2:x64
</vuln:product>
<vuln:product>cpe:/o:microsoft:windows_7:::x86
</vuln:product>
<vuln:product>cpe:/o:microsoft:windows_7::sp1:x86
</vuln:product>
<vuln:product>cpe:/o:microsoft:windows_vista::sp2
</vuln:product>
</vuln:vulnerable-software-list>
<vuln:cve-id>CVE-2012-0150</vuln:cve-id>
<vuln:cvss>
<cvss:base_metrics>
<cvss:score>9.3</cvss:score>
<cvss:access-vector>NETWORK</cvss:access-vector>
<cvss:access-complexity>MEDIUM</cvss:access-complexity>
<cvss:authentication>NONE</cvss:authentication>
</cvss:base_metrics>
</vuln:cvss>
<vuln:cwe id=“CWE-119” />
<vuln:references xml:lang=“en”
reference_type=”VENDOR_ADVISORY”>
<vuln:source>MS</vuln:source>
<vuln:reference
href=“http://technet.microsoft.com/security/bulletin/MS12-013”
xml:lang=“en”>MS12-013</vuln:reference>
</vuln:references>
<vuln:summary>Buffer overflow in msvcrt.dll in Microsoft
Windows Vista SP2, Windows Server 2008 SP2, R2, and R2 SP1,
and Windows 7 Gold and SP1 allows remote attackers to execute
arbitrary code via a crafted media file, aka “Msvcrt.dll
Buffer Overflow Vulnerability.”
</vuln:summary>
</entry>
</nvd>
```

Fig. 3. An excerpt of an NVD XML entry

entity and concept spotter was trained over a data corpus of unstructured texts from security blogs, CVE descriptions and security bulletins.

Our gold-standard dataset was created from human annotations of these unstructured pieces of text. The dataset was randomized and split into five equal chunks. The CRF-based classifier was trained over this dataset using the Stanford NER and appropriate feature selection, as mentioned previously. We evaluated the classifier using five-fold cross-validation, where four chunks of data were provided as training input to the classifier system and one chunk as a test set. The training set, on average, consisted of 3800 tagged entities and over 38000 tokens while the test set, on an average, consisted of over 9000 tokens and over 1200 entities. Figure 6 shows the results of each run in the five-fold cross validation experiment. On analysis, the trained model was observed to demonstrate promising results. Figure 5 shows a graph and a breakdown of the overall system performance on test data.

```
@prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ebqids:<http://ebiquity.umbc.edu/IDSv2.0.1.owl#> .
@prefix dbpedia:<http://dbpedia.org/resource/> .
<http://web.nvd.nist.gov/view/vuln/detail?vulnId=2012-0150>
ebqids:cveID “http://bit.ly/11A3wow”;
ebqids:cweID “http://cwe.mitre.org/data/definitions/119”;
ebqids:affectsProduct “dbpedia:Windows_Vista” ,
”dbpedia:Windows_7” ;
ebqids:summary “Buffer overflow in msvcrt.dll in Microsoft
Windows Vista SP2, Windows Server 2008 SP2, R2, and R2 SP1,
and Windows 7 Gold and SP1 allows remote attackers to execute
arbitrary code via a crafted media file, aka “Msvcrt.dll
Buffer Overflow Vulnerability.”” ;
ebqids:hasAccessComplexity “MEDIUM” ;
ebqids:hasAccessVector “NETWORK” ;
ebqids:hasAuthentication “NONE” ;
ebqids:hasSeverityScore “9.3” ;
ebqids:hasVulnerabilitySource
“http://technet.microsoft.com/security/bulletin/MS12-013” ;
ebqids:hasMeans “dbpedia:Buffer_overflow” ;
ebqids:hasConsequence “dbpedia:Arbitrary_code_execution” ;
ebqids:hasTerms “http://dbpedia.org/resource/Computer_file” ,
“http://dbpedia.org/resource/Dynamic-link_library” ,
“http://dbpedia.org/resource/Vulnerability_(computing)” .
```

Fig. 4. Turtle representation of extracted information

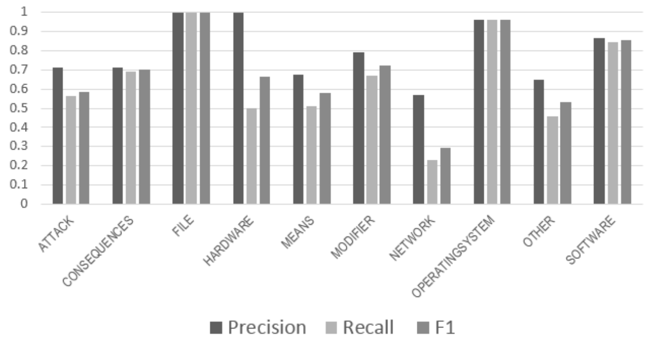

Fig. 5. Class-wise evaluation of the cybersecurity concept and entity spotter.

We used the precision, recall and F1 score measures to evaluate our CRF classifier. The entity and concept spotter generated consistent results after applying five-fold cross-validation, as shown in Figure 6. The weighted average of the precision value was calculated to be 0.83, the weighted average for recall was 0.76 and the weighted average F1 score was 0.80. This weighted average score was calculated from the values in Table I. We also noted that the *Gazetteers* feature from Stanford NER helped improve the score of *Software* and *Operating System* classes. There was notable inconsistency between the *Means* and *Consequences* classes. Their collective precision score was recorded as 0.75. A possible explanation would be that most of the false positives in both classes belonged to the opposite class. Moreover, it was observed that entities tagged as Means and Consequences were the most ambiguous terms encountered during the annotation process. Table I shows the statistics for the tested dataset in terms of true positives (TP), false positives (FP) and false negatives (FN). These statistics were calculated

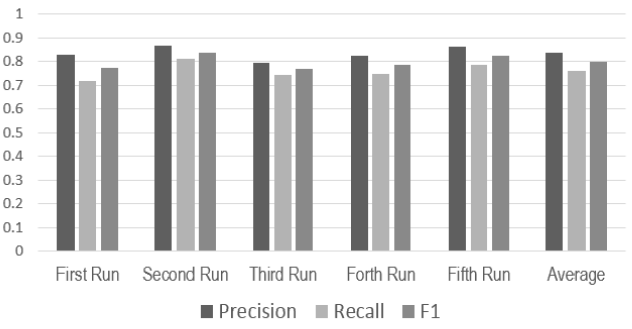

Fig. 6. Results of five-fold cross validation experiment

TABLE I
RESULTS OF CYBERSECURITY CONCEPT SPOTTER

CLASS	TP	FP	FN
ATTACK	30	14	27
CONSEQUENCES	299	123	135
FILE	52	0	0
HARDWARE	3	0	2
MEANS	185	94	177
MODIFIER	320	79	147
NETWORK	14	15	45
OPERATINGSYSTEM	920	34	36
OTHER	167	89	230
SOFTWARE	1449	224	268
TOTAL	**3439**	**672**	**1063**

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ebqids: <http://ebiquity.umbc.edu/IDSv2.0.1.owl#> .
@prefix dbpedia: <http://dbpedia.org/resource/> .
<http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2013-0610>
ebqids:cveID "http://bit.ly/11A3wow";
ebqids:cweID "http://cwe.mitre.org/data/definitions/119" ;
ebqids:summary "Stack-based buffer overflow in Adobe Reader
and Acrobat 9.x before 9.5.3, 10.x before 10.1.5, and 11.x
before 11.0.1, not different from CVE-2013-0626." ;
ebqids:hasAccessComplexity "LOW" ;
ebqids:hasAccessVector "NETWORK" ;
ebqids:hasAuthentication "NONE" ;
ebqids:hasSeverityScore "10.0" ;
ebqids:hasVulnerabilitySource
"http://rhn.redhat.com/errata/RHSA-2013-0150.html" ,
"http://adobe.com/support/security/bulletins/apsb13-02.html" ,
"http://opensuse.org/opensuse-updates/2013-01/msg00081.html" ,
"http://opensuse.org/opensuse-updates/2013-01/msg00028.html" ;
ebqids:hasMeans "dbpedia:Buffer_overflow" ;
ebqids:affectsProduct "dbpedia:Adobe_Acrobat" .
```

Fig. 7. An NVD entry excerpt which has an incomplete description, since it refers to another NVD CVE entry.

collectively for five-fold cross validation.

Our concept spotter system does face certain challenges when identifying entities for some specific NVD descriptions that refer to another NVD CVE description. There are certain sets of entries in the NVD repositories (mostly related to the same software product) that are observed to have the same summary description, with minor changes in the rest of the NVD (CVE, CVSS) properties. However, they provide references to other CVE IDs that might have the appropriate, more granular details regarding the attack.

Figure 7 shows an excerpt of NVD CVE-2013-0610 entry that describes a buffer overflow attack on Adobe Acrobat and Reader. Although the NVD summary describes the means of the attack (*Buffer Overflow*) and the affected product (*Adobe Acrobat*), it does not provide further information such as the consequences. Moreover, the severity score for the entry is 10 ("Critical"). Retrieving the text associated with the referenced NVD CVE entries might help gather more information about the nature of such a critical attack, not only for a single CVE but a group of CVEs that might be reported together. In the future, we plan to consolidate these missed sources to give richer context on such vulnerabilities.

Linked data supports data integration and interoperation by using the RDF representation, which has globally unique identifiers (URIs). Moreover, the linked data paradigm stipulates that these identifiers should be "resolvable", i.e., one can use an HTTP GET request on a URI and retrieve additional information about the thing it denotes. Integration can be further enhanced by linking a URI to another from a central knowledge base like DBpedia. These links assert the equivalence of objects the two URIs denote. Such a central resource serves as a common knowledge hub, allowing sets of URIs to be understood as equivalent if they link to the same source.

However, not all concepts and terms spotted in the vulnerability descriptions can be associated with a valid, available resource. This may be the case when there is no relevant DBpedia resource available for the concept. The terms extracted by the cybersecurity entity and concept spotter, though not instantiated to relevant URIs, are important for profiling an attack.

There is a considerable difference in the number of annotations picked by our cybersecurity classifier and the number of annotations (and thereby links) generated by DBpedia Spotlight. Figure 8 shows the comparison for the number of annotations extracted from a set of 300 NVD vulnerability descriptions by DBpedia spotlight and our cybersecurity classifier.

This not only demonstrates the performance of our classifier, but also indicates the absence of entities that describe security concepts in the DBpedia knowledge base. In order to represent these terms in useful RDF instances, we plan to resolve the unidentified concepts to external URIs that formally describe the security concept, and thereby reduce fact duplication and re-utilize existing URIs. Hence our prototype can support knowledge generation of terms relevant to cybersecurity that are not identifiable as relevant DBpedia resources.

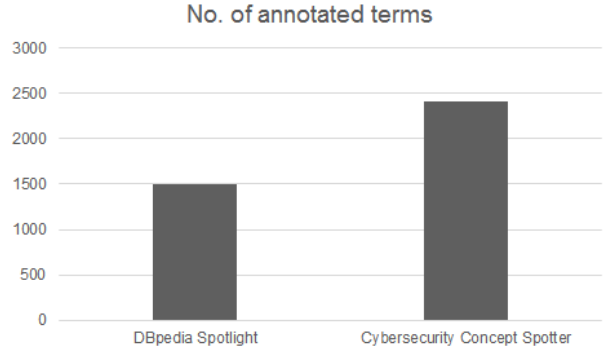

Fig. 8. Comparison of number of annotations

V. Conclusion and Future Work

We demonstrate a prototype for an entity and concept spotting framework that identifies cybersecurity-related concepts from heterogeneous data sources, aligns and links them to relevant resources on the Web using the IDS ontology, and generates an RDF linked data collection. We provide a semantic data representation for the concepts that are not limited to the NVD dataset. The linked data generation module leverages interoperability and reuse of URIs, thereby enhancing the binding with the Linked Open Data cloud.

Our evaluation showed promising results for the extraction framework. We plan to focus on further extracting previously unidentified security concepts from any given piece of text, identify properties and find relationships based on a heuristic. There are ongoing efforts to enhance the ontology to model detailed network-related terms and privacy concepts. We believe that expressing structured and unstructured cybersecurity-related text as linked data has potential to leverage automatic consumption and reasoning of security concepts, and can drive applications such as a situation aware intrusion detection system to detect and prevent potential "zero-day" attacks.

Acknowledgment

This research was supported by grants from AFOSR (FA9550-08-1-0265) and NSF (IIS-1250627).

References

[1] "Cyber criminals target Skype, Facebook and Windows users," http://bit.ly/cyberCriminals.

[2] D. McGuinness and F. e. a. Van Harmelen, "OWL web ontology language overview," World WIde Web Consortium, Tech. Rep., 2004.

[3] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data - the story so far," *Int. Journal on Semantic Web and Information Systems*, vol. 5, no. 3, pp. 1–22, 2009.

[4] S. D. Quinn, D. A. Waltermire, C. S. Johnson, K. A. Scarfone, and J. F. Banghart, "SP 800-126. The Technical Specification for the Security Content Automation Protocol (SCAP): SCAP Version 1.0," National Institute of Standards & Technology, Gaithersburg, MD, Tech. Rep., 2009.

[5] S. More, M. Matthews, A. Joshi, and T. Finin, "A Knowledge-Based Approach to Intrusion Detection Modeling," in *Security and Privacy Workshops (SPW), 2012 IEEE Symposium on*, 2012, pp. 75–81.

[6] V. Mulwad, W. Li, A. Joshi, T. Finin, and K. Viswanathan, "Extracting Information about Security Vulnerabilities from Web Text," in *IEEE/WIC/ACM Int. Conf. on Web Intelligence and Intelligent Agent Technology*, vol. 3, 2011, pp. 257–260.

[7] Z. Syed, "Wikitology: A Novel Hybrid Knowledge Base Derived from Wikipedia," Ph.D. dissertation, University of Maryland, Baltimore County, August 2010.

[8] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: A Core of Semantic Knowledge," in *16th Int. World Wide Web Conf.* New York: ACM Press, 2007.

[9] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proc. ACM Int. Conf. on Management of Data*. ACM, 2008, pp. 1247–1250.

[10] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in *18th Int. Conf. on Machine Learning*. Morgan Kaufmann, 2001, pp. 282–289.

[11] T. Reuters, "OpenCalais," 2009.

[12] G. Rizzo and R. Troncy, "NERD: a framework for unifying named entity recognition and disambiguation extraction tools," in *13th Conf. of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2012, pp. 73–76.

[13] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by Gibbs sampling," in *43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2005, pp. 363–370.

[14] V. Khadilkar, J. Rachapalli, and B. Thuraisingham, "Semantic Web Implementation Scheme for National Vulnerability Database (Common Platform Enumeration Data)," University of Texas at Dallas, Tech. Rep. UTDCS-01-10, 2010.

[15] J. Undercoffer, J. Pinkston, A. Joshi, and T. Finin, "A Target-Centric Ontology for Intrusion Detection," in *IJCAI-03 Workshop on Ontologies and Distributed Systems*. Morgan Kaufmann, 2004, pp. 47–58.

[16] "Linked Data," http://linkeddata.org/, 2007.

[17] E. PrudHommeaux, A. Seaborne *et al.*, "SPARQL query language for RDF," 2008, W3C Recommendation.

[18] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer, "DBpedia spotlight: shedding light on the web of documents," in *7th Int. Conf. on Semantic Systems*. ACM, 2011, pp. 1–8.

[19] "Stanford NER," http://nlp.stanford.edu/software/CRF-NER.shtml.

[20] R. Lal, "Annotations of cybersecurity blogs and articles," http://ebiquity.umbc.edu/r/355, June 2013.

[21] P. Stenetorp, S. Pyysalo, G. Topić, T. Ohta, S. Ananiadou, and J. Tsujii, "BRAT: a web-based tool for NLP-assisted text annotation," in *Demonstrations, 13th Conf. of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2012, pp. 102–107.

[22] "BRAT Annotation Tool," http://brat.nlplab.org/index.html.

[23] R. Lal, "Information Extraction of Security related entities and concepts from unstructured text," Master's thesis, University of Maryland Baltimore County, 2013.

[24] J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson, "The JENA Semantic Web platform: architecture and design," HP Laboratories, Tech. Rep. Technical Report HPL-2003-146, 2003.

[25] A. Joshi, "Linked Data for Software Security Concepts and Vulnerability Descriptions," Master's thesis, University of Maryland Baltimore County, 2013.