Varish Mulwad, Wenjia Li, Anupam Joshi, Tim Finin, and Krishnamurthy Viswanathan, Extracting Information about Security Vulnerabilities
from Web Text, Proceedings of the Web Intelligence for Information Security Workshop, Aug. 2011, Lyon, FR, IEEE Computer Society Press.

Extracting Information about Security Vulnerabilities from Web Text

Varish Mulwad, Wenjia Li, Anupam Joshi, Tim Finin and Krishnamurthy Viswanathan
Computer Science and Electrical Engineering
University of Maryland, Baltimore County
Baltimore, MD, USA
{*varish1,wenjia1,joshi,finin,krishna3*}@*cs.umbc.edu*

***Abstract*—The Web is an important source of information about computer security threats, vulnerabilities and cyber-attacks. We present initial work on developing a framework to detect and extract information about vulnerabilities and attacks from Web text. Our prototype system uses Wikitology, a general purpose knowledge base derived from Wikipedia, to extract concepts that describe specific vulnerabilities and attacks, map them to related concepts from DBpedia and generate machine understandable assertions. Such a framework will be useful in adding structure to already existing vulnerability descriptions as well as detecting new ones. We evaluate our approach against vulnerability descriptions from the National Vulnerability Database. Our results suggest that it can be useful in monitoring streams of text from social media or chat rooms to identify potential new attacks and vulnerabilities or to collect data on the spread and volume of existing ones.**

***Keywords*-security, vulnerability, information extraction, entity linking**

I. Introduction

The Web has become a primary source of knowledge and information, largely replacing encyclopedias and reference books. This is especially true for dynamic topics, such as computer security threats, vulnerabilities and cyber-attacks. Detailed information on these topics are found in Web-accessible repositories of structured and semi-structured information, including the National Vulnerability Database (NVD) [1], IBM's XFORCE [2], the US-CERT Vulnerability Notes Database [3], and other security advisory sources. Various informal sources complement these curated repositories, such as computer help forums, hacker blogs and forums, chat rooms and social media streams. Even though these are noisy, redundant and often contain misinformation, they can be mined and aggregated to provide early warnings of new vulnerabilities and attacks, track the evolution of existing ones, produce evidence for attribution and estimate the prevalence and geographical distribution of known problems. The integration of knowledge and data from these two very different domains has great potential, but also offers significant challenges.

We present a framework that analyses text snippets found on the Web to identify and generate assertions about vulnerabilities, threats and attacks. Given a text description, we use the Wikitology [4] knowledge base along with a taxonomy of computer security exploits to decide if it is relevant to computer security and, if so, identify the set of security concepts it evokes. Once the concepts are identified and linked to related objects in our knowledge-base, we generate assertions about the text description.

II. Motivation and Background

A system to extract vulnerabilities, threats and attacks from unstructured text will be useful to many applications. Sources such as NVD provide XML and ATOM feeds of the latest vulnerability. Although this contains some structured information, such as vendor, software name, version, and severity, important information such as the exploit type (e.g., cross-site scripting) and attack mode (e.g., ping of death) are only mentioned, if at all, in the unstructured text. Using our techniques, we can detect concepts such as exploits and attacks from the free text and thus add more structure to the XML feed, adding to its value and utility.

We see our framework as part of a larger system which scans Web resources for descriptions of new vulnerabilities, threats and zero day attacks. Such a system can monitor and digest information from a set of sources, such as vulnerability description feeds and as well as hacker forums, chat rooms and hacker blogs. The system can process a stream of text to detect potential vulnerability descriptions and extract concepts and topics of interest as well as associated entities such as software products and notify security expert for further action about them.

Our framework will be also useful in creating a knowledge-base of vulnerabilities and threats. Using it, existing semi-structured and unstructured vulnerability descriptions can be transformed into machine understandable assertions in the RDF/OWL Semantic Web language [5] and linked to appropriate entities and concepts in the linked data cloud [6].

A key component of our framework is Wikitology, a general purpose, hybrid knowledge base containing both structured and unstructured information extracted from Wikipedia, DBpedia [7], Freebase [8], WordNet [9] and Yago [10]. Wikitology's interface is based on a specialized information retrieval index implemented using the Lucene information retrieval system that supports complex queries with structured and unstructured components and constraints. Wikitology provides various fields to query against

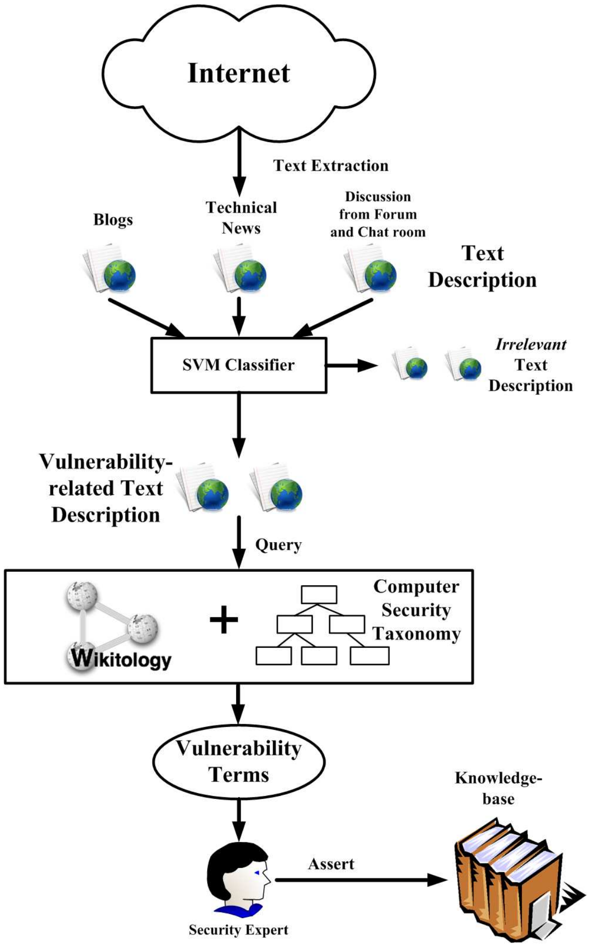

Figure 1. Our prototype system uses an SVM classifier to identify potential vulnerability descriptions, an information extraction system to identify relevant concepts, entities and events and link these to knowledge base objects, and custom procedures to generate assertions encoded in the Semantic Web language OWL.

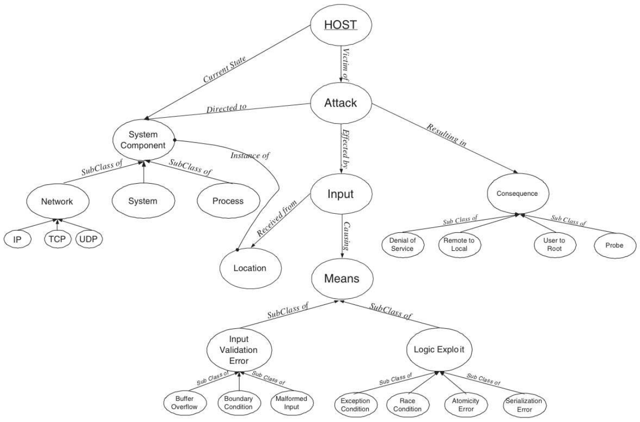

Figure 2. This is a high-level sketch of the IDS OWL ontology for describing computer attack based upon an analysis of over 4000 classes of computer intrusions and their corresponding attack strategies and is categorized according to system component targeted, means of attack, consequence of attack and location of attacker.

such as Wikipedia title, article categories, DBpedia types, Wikipedia article contents to name a few. For every query Wikitology returns a ranked and scored list of concepts that match the query.

III. System Architecture

Figure 1 shows a diagram of our framework, which comprises three components: (i) an SVM classifier to identify potential vulnerability descriptions; (ii) an information extraction system to identify relevant concepts, entities, relations and events from such descriptions using Wikitology and a computer security exploits ontology, and (iii) custom procedures to generate machine understandable assertions encoded in the Semantic Web language OWL.

The first step in the approach is identifying texts that describe vulnerabilities and threats. We trained an SVM classifier to recognize text segments likely to contain security-related information using the standard unigram bag of words vector model. The training set consisted of 75 positive and 80 negative examples. The positive examples were vulnerability text descriptions from NVD data feed and the negative examples were technical text descriptions sampled randomly from websites such as CNET [11]. Preliminary evaluations (using ten-fold cross validation) on this small dataset showed that classifier was able to correctly identify all vulnerability text descriptions.

Once a potential vulnerability description is found, we next extract entities and concepts of interests from that text. We can extract entities like organizations and software products using standard named entity recognition tools like OpenCalais [12]. The next step is to extract concepts related to computer security exploits. Our algorithm for concept extraction uses the knowledge from Wikitology along with a computer security exploit taxonomy extracted from Wikipedia to identify vulnerabilities, threats and attacks. The algorithm queries the text description against the *contents* field of Wikitology. The knowledge-base returns a ranked list of Wikipedia concepts matching the query. These results include software vendors, software products, organizations, people, places as well the concepts of interest to us like vulnerabilities, threats and attacks.

Since Wikitology returns entities and concepts of all possible types, we need to develop a mechanism to filter out the unwanted results. One possible approach is to restrict the results based on Wikipedia categories, DBpedia classes or Yago types. However, for computer security exploits none of these systems are well developed, complete or parallel standard classifications. Every article on Wikipedia is associated with a set of categories. However there are many articles in Wikipedia where the parent categories are not included in the article's set of categories. Thus, it is difficult to restrict results the articles returned from

Line : Buffer overflow in Fax4Decode in LibTIFF 3.9.4 and possibly other versions, as used in ImageIO in Apple iTunes before 10.2 on Windows and other products, allows remote attackers to execute arbitrary code or cause a denial of service (application crash) via a crafted TIFF Internet Fax image file that has been compressed using CCITT Group 4 encoding, related to the EXPAND2D macro in libtiff/tif_fax3.h.

```
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix ids: <http://ebiquity.org/ontologies/cybersecurity/
ids/v2.0/ids♯ > .
[a ids:Vulnerability;
ids:hasMeans dbpedia:Buffer_overflow;
ids:hasConsequence dbpedia:Denial-of-service_attack].
```

Figure 3. The top box is an example text description of a vulnerability in the NIST NVD data feed. The second shows extracted information encoded as OWL assertions serialized using N3.

Wikitology to some super categories from Wikipedia. To overcome these issues, we extracted the taxonomy under the Wikipedia category *Computer_security_exploits* [13]. This taxonomy allows us to filter and select concepts that belong to this category or fall under it. In our evaluation section we present results for concepts related to computer security exploits extracted from the top five and top ten Wikipedia concepts returned by Wikitology for every query.

Once the security exploit concepts are extracted from text snippets using the concept extraction algorithm, we generate machine-understandable assertions from them. We use the IDS OWL ontology [14] [15] [16] to represent and reason about intrusion detection concepts and events and to encode the resulting inferred facts. This ontology provides classes to describe different aspects of an attack. For example, the *Means* class is a super class of concepts representing methods to conduct the attack. The *Consequence* class subsumes classes representing attack outcomes and the *System* class covers classes for systems under various attacks. The ontology also has properties such as *hasMeans*, *hasConsequence* to describe the means and consequence for a given vulnerability. We use the knowledge from Wikitology to map the concepts extracted to the respective DBpedia concepts. Figure 3 shows a text description from the NIST NVD/CVE data feed and the assertions extracted from it encoded in the Semantic Web language OWL and serialized in N3.

Such assertions can be added to a knowledge-base of computer security exploits and can be used while reasoning and detecting new vulnerabilities and threats. In future we will also explore generating a more complex assertion which will capture software products under attack, vendors of the products etc.

IV. Discussion and Evaluation

Our work is broadly related to problems of concept spotting and named entity recognition. While named entity recognition is a well known and explored problem (see [17]), it has been focused on extracting people, places and organizations from free text. To the best of our knowledge, no effort has focused on extracting computer security exploits and associated entities, relations and events from free text. Portions of the NVD database has been mapped into RDF [18] using a schema-based approach [19] but much of the information remained in strings rather than RDF instances.

We evaluated our prototype system against a collection of vulnerability text descriptions from NVD. For every text description, the KB returned a ranked list of top N Wikipedia concepts associated with the text. The concept extraction algorithm then applies the filtering mechanism described above to produce a ranked list of computer security exploits from these.

In the first evaluation, we checked whether the top N concepts returned by the algorithm includes the correct exploit and at what rank is the correct exploit predicted. For each text description, one of the authors of the paper (who has a background in computer security and networks), went through the ranked list of computer security exploit concepts returned by the algorithm and identified whether a correct exploit was predicted and at what rank it was predicted.

Out of 107 vulnerability text descriptions with $N = 5$, the algorithm identified one or more security exploits for 76 text descriptions and it failed for 31. Out of the *76* descriptions in which an security exploit concept was detected, for *68* text descriptions (*89.47%*) our algorithm detected the correct concept. Out of the *68* text descriptions in which a correct concept was detected, *66* times the correct concept was at *rank 1*. In general the average rank for the correct concept was *1.0588*

Out of 107 vulnerability text descriptions when $N = 10$, the algorithm identified one or more security exploits for *80* text descriptions and it did not detect any security exploit concept for *27* text descriptions. Out of the *80* descriptions in which an security exploit entity was detected, for *72* text descriptions (*90%*) our algorithm detected the correct concept, which is slightly better than when $N = 5$. Out of the *72* text descriptions in which a correct concept was detected, *69* times the correct concept was at *rank 1*. In general the average rank for the correct concept was *1.125*. From the first graph in Figure 4, it is evident that the concept extraction algorithm yields a high accuracy for both $N = 5$ and $N = 10$.

Every concept on Wikipedia is associated with a set of multiple labels (or categories). There are a set of specific categories as well a set of general categories and labels associated with a given concept. Thus, to evaluate the ranked order of security exploit concepts generated by the algorithm, we perform a second evaluation. We compared the ranked list of computer security exploits generated by the algorithm against the ranked list of computer security exploits created by a human expert (the same author as before).

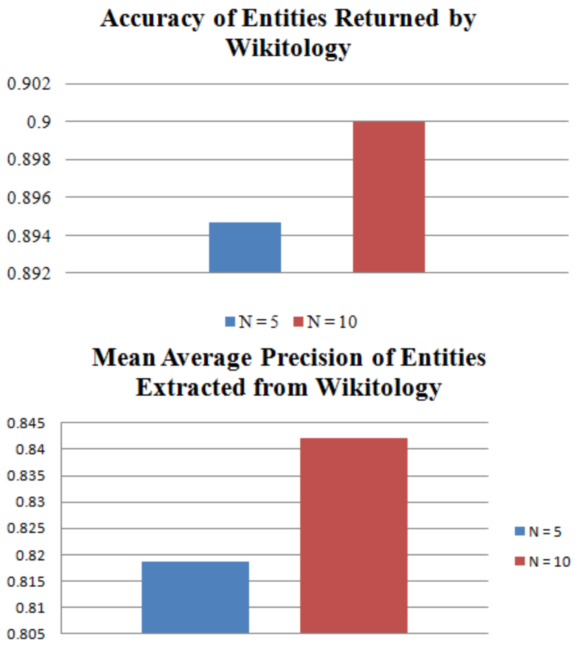

Figure 4. Figure show graphs for accuracies for concepts returned by Wikitology and the mean average precision between ranked list of concepts extracted by our algorithm against the list generated by our human expert for top five and top ten results returned by Wikitology.

We used the *average precision* [20] measure to compare two ranked list. Mean average precision (MAP) gives us the average precision over a set of queries. We calculated MAP for $N = 5$ over a smaller subset 17 queries (i.e. 17 text descriptions) and for $N = 10$ over subset of 19 queries. MAP for both $N = 5$ and $N = 10$ is greater than 0.8 (see Figure 4) which indicates that the concept extraction algorithm is not only producing a list of correct entities, but also producing it in the desired order.

V. Conclusion and Future Work

We described a prototype system to identify vulnerabilities, threats and attacks in Web text from which machine understandable OWL assertions can be generated. Evaluations showed promising results for our framework. We plan to focus on developing stronger reasoning algorithms and to develop a more principled security exploits ontology. Many difficult challenges will need to be addressed, including representing uncertainty, reasoning with both logical and probabilistic knowledge, and modeling and reasoning about the temporal aspects of the data.

Acknowledgment

This work was partially supported by an grant from the Air Force Office of Scientific Research (MURI FA9550-08-1-0265) and a gift from Northrop Grumman Corporation.

References

[1] “National vulnerability database,” http://nvd.nist.gov.

[2] “Internet security systems x-force security threats,” http://xforce.iss.net.

[3] US-CERT, “Vulnerability notes database,” http://www.kb.cert.org/vuls/.

[4] T. Finin and Z. Syed, “Creating and Exploiting a Web of Semantic Data,” in *Proc. 2nd Int. Conf. on Agents and Artificial Intelligence*. Springer, January 2010.

[5] O. Lassila and R. Swick, “Resource description framework (rdf): Model and syntax specification. recommendation,” W3C, Tech. Rep., 1999.

[6] C. Bizer, “The emerging web of linked data,” *IEEE Intelligent Systems*, vol. 24, no. 5, pp. 87–92, 2009.

[7] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, “Dbpedia - a crystallization point for the web of data,” *Journal of Web Semantics*, vol. 7, no. 3, pp. 154–165, 2009.

[8] 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*. New York, NY: ACM, 2008, pp. 1247–1250.

[9] G. A. Miller, “Wordnet: a lexical database for english,” *Commun. ACM*, vol. 38, pp. 39–41, November 1995.

[10] 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.

[11] CNET, http://www.cnet.com/.

[12] “Opencalais,” http://opencalais.com/.

[13] http://en.wikipedia.org/w/index.php?title=Special:CategoryTree.

[14] J. Undercoffer, A. Joshi, T. Finin, and J. Pinkston, “Using DAML+OIL to classify intrusive behaviours,” *The Knowledge Engineering Review*, vol. 18, pp. 221–241, 2003.

[15] J. Undercoffer, T. Finin, A. Joshi, and J. Pinkston, “A target-centric ontology for intrusion detection,” in *Proc. 18th Int. Joint Conf. on Artificial Intelligence*, 2004.

[16] http://ebiquity.umbc.edu/ontologies/cybersecurity/ids/.

[17] D. Nadeau and S. Sekine, “A survey of named entity recognition and classification,” *Linguisticae Investigationes*, vol. 30, no. 1, pp. 3–26, January 2007.

[18] V. Khadilkar, J. Rachapalli, and B. Thuraisingham, “Semantic web implementation scheme for national vulnerability database,” Univ. of Texas at Dallas, Tech. Rep. UTDCS-01-10, 2010.

[19] C. Bizer and A. Seaborne, “D2rq-treating non-rdf databases as virtual rdf graphs,” in *Proc. 3rd International Semantic Web Conference*, 2004.

[20] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, 1st ed. Cambridge University Press, July 2008.