Developing an Ontology for Cyber Security Knowledge Graphs [*]

Michael Iannacone
iannaconemd@ornl.gov
Oak Ridge National Laboratory

Shawn Bohn
shawn.bohn@pnnl.gov
Pacific Northwest National Laboratory

Grant Nakamura
grant.nakamura@pnnl.gov
Pacific Northwest National Laboratory

John Gerth
gerth@graphics.stanford.edu
Stanford University

Kelly Huffer
testakm@ornl.gov
Oak Ridge National Laboratory

Robert Bridges
bridgesra@ornl.gov
Oak Ridge National Laboratory

Erik Ferragut
ferragutem@ornl.gov
Oak Ridge National Laboratory

John Goodall
jgoodall@ornl.gov
Oak Ridge National Laboratory

ABSTRACT

In this paper we describe an ontology developed for a cyber security knowledge graph database. This is intended to provide an organized schema that incorporates information from a large variety of structured and unstructured data sources, and includes all relevant concepts within the domain. We compare the resulting ontology with previous efforts, discuss its strengths and limitations, and describe areas for future work.

Categories and Subject Descriptors

H.3.3 [**Information systems**]: Search and Retrieval

Keywords

cyber security, information extraction, ontology architecture, security automation

1. INTRODUCTION

Cyber security professionals have a critical need for the most recent information to perform their duties. Moreover, as the field of cyber security has become more technically complex and more economically important, the amount of relevant information has been increasing rapidly, leading to difficulty in managing and using this information. There have been some notable successes in creating structured data sources of some domain entities (e.g. vulnerability databases,) however much domain information is only available in text sources. Where structured data sources are available, most use whatever representation is convenient, without any consensus on structure, contents, or names of entities. Greater effort is needed in the organization of this cyber security information, to aid both analysts and automated systems.

The data feeds provided by anti-virus (AV) vendors provide an important example of these difficulties; in some cases they will include DNS requests or other information about network traffic generated by the malware, at varying levels of detail, but in many cases this information is not provided. Likewise, some of these sources include lists of modified files, modified registry keys, and other information about modifications to the host environment, while other sources lack this information. Often, the same vendor will change what information is included, or how it is represented, without updating their previous entries. There is also the problem of grouping and naming the malware samples. Most AV vendors go through this organizing and naming process independently, so there is often no consensus, and any cross-referencing between these datasets is generally absent or sparse.

Another problem is the lack of cross-references between datasets of different types of entities. Continuing with the example above, it would be very useful for a malware database to reference IP and DNS registration information, and equally useful for IP and DNS blacklists to reference malware database entries wherever appropriate. Likewise, a vulnerability database could reference any malware samples which exploit that vulnerability, and vice-versa. This kind of association would be very valuable for a security practitioner, or even for automated tools such as vulnerability scanners or intrusion detection systems (IDS), however this type of

[*] This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan `http://energy.gov/downloads/doe-public-access-plan`.

CISR '15 April 07 - 09, 2015, Oak Ridge, TN, USA
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3345-0/15/04 ...$15.00.
http://dx.doi.org/10.1145/2746266.2746278

information is simply not available in most cases.

This ontology was developed to enable this kind of integrated data resource, which the STUCCO project [1] aims to provide. It combines a wide variety of publicly-available datasets, along with internal information such as netflows and IDS alerts, to build this information resource. We have developed this ontology as part of this effort, with the goal of organizing the information in the most useful way for both analysts and automated tools, given the constraints of the available datasets discussed above.

2. RELATED WORK

There has been significant previous work in this area, which we have incorporated as much as possible given our needs and the problems described above. First, there have been many efforts to represent knowledge of specific domain concepts, such as vulnerabilities, or attacks, or malware. Second, there have been previous attempts to create more general ontologies to combine these concepts. We provide an overview of both areas, and focus on some efforts which are of particular relevance to this work.

2.1 Modeling Security Concepts

The effort to create ontologies and taxonomies focused on specific security-related concepts has produced significant previous work. There are many useful surveys which give broad coverage of this area [9, 16] so in this paper we give only a brief overview.

The most mature of these frameworks are those describing vulnerabilities [2, 13, 21]. One of the main motivations of these efforts was to guide the development of security tools and practices [23]. These efforts ultimately lead to the development of widely-used vulnerability databases such as NVD[1] and OSVDB.[2]

Attack taxonomies progressed along a parallel path to vulnerability taxonomies, due to similar needs in developing tools and practices to cope with various attacks [7, 8, 12, 14]. This work has contributed to a common language and common understanding within the field, and lead to some useful resources, such as the OWASP Top 10[3]. However, there is no publicly available, structured database of attacks or incidents, in contrast with the vulnerability databases.

Efforts to categorize other topics within the security field, such as adversaries and malware, have proven more difficult. In these areas, the security landscape has been more fluid, and many past frameworks have not stayed relevant as the motivations and techniques of adversaries have evolved. There are still useful data resources available in these areas, such as IP and DNS blacklists, and AV vendor data feeds, but these sources differ greatly in what information is included, and in how it is represented.

2.2 Integrating Security Concepts

Our work follows previous efforts to combine these topics into an overall ontology representing the cyber-security domain. This previous work has been described thoroughly in some recent surveys [4], so here we focus on two ongoing efforts of particular note. Both of these efforts each share some common goals with the STUCCO project, and the ontologies they have developed share some similarities.

The first of these is a significant body of work by a group of authors from University of Maryland, Baltimore County (UMBC). Early work develops an ontology to model attacks and related entities, for use in an IDS [22]. More recent work uses a similar ontology to guide the extraction of entities and relations from unstructured text articles [11, 18]. These entities and relations are then also used in an IDS [17].

Secondly, MITRE has been investigating ways to develop an ontology for the cyber security domain [19, 20]. This has been of significant interest, in part due to their past successes in creating and maintaining several standards and datasets for specific topic areas within this domain. This effort has been developing rapidly, and is now beginning to attract early users. More complete documentation is available from the STIX[4] website, including a technical report [3]. This effort has involved combining and relating data resources and standards, similar to our current effort. It is interesting to note that, while both ontologies contain largely the same concepts, the STIX standard has in most cases opted to group them at a more general level than we have. For example, they include a "Tactics, Techniques and Procedures (TTP)" concept, which includes many components, such as malware, attack patterns, intended effects of an attack, etc.

3. DATA SOURCES

This ontology is intended to facilitate the integration of data from a variety of both structured and unstructured sources. Currently, data from 13 structured sources is included; this data is fed into a pipeline which collects the data, converts it to GraphSON format[5] and then loads it into the database, merging with existing records as needed. Thus, the ontology needs to provide entity types and properties which can represent all needed fields from all datasets, and we must develop some mapping between these before the data can be added to the knowledge graph.

There is also ongoing work to incorporate data from unstructured text sources, through a similar process [5, 15, 10]. This text processing also relies directly on the ontology to define the entities, relations, and properties that must be extracted from these texts. This presents significant difficulty because the language used in this domain ranges from extremely specific to extremely ambiguous. Furthermore, in many cases technical terms are simply used incorrectly — one glaring example is the many news headlines that contained the phrase "Heartbleed Virus." In general, the more specificity in the ontology definition, the more difficult it is to populate it from the available text sources.

The most problematic example of this constraint was differentiating between malware and exploits. A large amount of modern malware is relatively modular, largely due to in-

[1] https://nvd.nist.gov/
[2] https://www.osvdb.org/
[3] https://www.owasp.org/
[4] Structured Threat Information eXpression
[5] https://github.com/tinkerpop/blueprints/wiki/GraphSON-Reader-and-Writer-Library/

creasing specialization among its producers and users. Often, the malware payload is a separate component from the exploit used to deliver it, allowing both components to be re-used in various new combinations. Additionally, sometimes proof-of-concept exploits, with no malicious payload, are made available by researchers. This distinction would be very useful to a user of the knowledge graph. Unfortunately, information with this much detail is rarely available, either from AV vendors or from unstructured text sources like news articles. For this reason, we opted to include exploit code under the more general "malware" label.

4. USE CASES

Our anticipated use cases for the knowledge graph also had a large impact on the design of the ontology. Broadly, these can be grouped into human users, and automated users.

For a human user, we can take the example of a system administrator who is performing some incident response. This could often involve tasks such as:

- Searching through flow records and IDS records by address during some time window, and comparing remote addresses against blacklists or reputation systems
- Gathering information about the software packages on impacted hosts, and comparing with vulnerability databases and IDS alerts
- Attempting to identify malware based on system changes and network traffic logs

Not only should all of this information be readily available, but it should be organized in a way that would make intuitive sense to this kind of user; thus the ontology should match the users' existing mental model of the domain as much as possible. In future work, we hope to measure whether the ontology has accomplished this goal.

Another important use is in automated systems. The IDS described in [17, 22] provides a useful example. When this type of IDS notices, for example, a sudden spike in connection attempts on port 22, it should be able to discover if any ssh service is running on that machine, and if so, it should be able to find any known vulnerabilities in that version of the service. This kind of system could triage alerts much more effectively given this contextual information, and in some cases it could even respond automatically (e.g. by adjusting IDS or firewall rules.) For the knowledge graph to be useful to such a system, the information must be both correct and specific, which raises the same trade-offs discussed in Section 3.

5. ONTOLOGY DESIGN

The resulting ontology is summarized in figure 1. Among the 15 entity types, there are 115 properties in total, which are omitted in this figure for simplicity. Because this ontology aims to provide an intuitive model, we will discuss some items of note, instead of defining every entity comprehensively.

The first point, as mentioned in section 3, is that there is no explicit *Exploit* entity, it is instead grouped with the *Malware* entity, as described previously. Next, note that *flow* entities may have an edge referring back to the software process which produced them. Most data sources cannot provide this, because of how the data is collected on the network (e.g. from the border router). In contrast, host-based systems such as Hone [6] can provide this contextual information, due to their visibility into the host state. STUCCO makes use of both types of sources, but maintains this additional context wherever it is available. Finally, note that the *Address* node is broken up into more specific sub-components; in practice the address must always include an edge to at least one of these items. This structure, while slightly more complex, aids significantly in generating queries — for example the common *IP:Port* combinations would be more difficult to query without the aggregation this node provides.

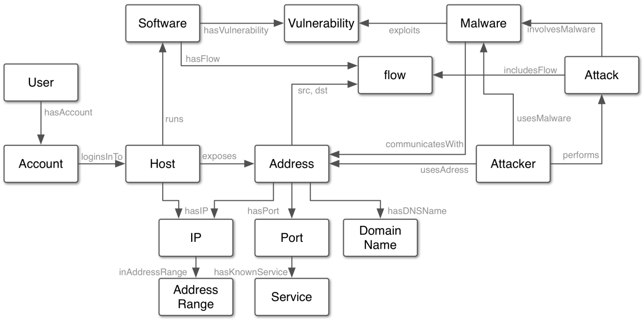

Figure 1: Entities and Relations in the STUCCO Ontology.

The full ontology definition, available on GitHub[6], includes text descriptions for all entity types, all relations, and all properties. Interested readers can refer to this repository directly, as this contains much more detail than we can provide here.

6. IMPLEMENTATION

The repository above specifies the ontology using JSON-Schema; the main benefit of this (e.g. compared to RDFS or OWL) is its compatibility with the GraphSON format that we use when loading and querying the graph database (currently Titan). This makes validating the incoming data very simple, and also defines the database schema during initialization.

Attributes of these entities and restrictions on these attributes are also specified as part of this JSONSchema definition. Currently, there are 115 properties in total, among the 15 entity types shown in Figure 1. These properties generally have restrictions of cardinality and type specified, and in some cases additional restrictions, such as allowable ranges, or a set of allowable values. Because JSONSchema is extensible, it also provides a convenient location to include additional metadata, which we use in upcoming work.

This choice also has some significant limitations. For example with OWL it is simple to perform automatic reasoning about transitive relationships, or to infer new relationships from known ones, based on first-order logic. However, these capabilities were not needed for our current use cases, so these limitations have caused little difficulty so far.

7. CONCLUSION AND FUTURE WORK

[6] https://github.com/stucco/ontology

This paper describes our efforts towards creating an ontology which can represent the cyber security domain, allowing information to be combined from as many sources as possible within this domain. STUCCO currently incorporates data from 13 structured sources with different formats, and as more are added, small additions and adjustments to the ontology will likely continue to occur. Likewise, as we develop more uses for the STUCCO knowledge graph, some changes could be needed to facilitate these new uses.

In future work, we plan to study how best to inter-operate with STIX and its related standards, since these are now beginning to gain acceptance among practitioners. As more data is provided in these formats, and as more tools can use data in these formats, interoperability will become increasingly important as this area develops.

8. ACKNOWLEDGEMENTS

This material is based on research sponsored by: the Department of Homeland Security (DHS) Science and Technology Directorate, Cyber Security Division (DHS S&T/CSD) via BAA 11-02; the Department of National Defence of Canada, Defence Research and Development Canada (DRDC); the Dutch Ministry of Security and Justice; and the Department of Energy (DOE). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of: the Department of Homeland Security; the Department of Energy; the U.S. Government; the Department of National Defence of Canada, Defence Research and Development Canada (DRDC); or the Dutch Ministry of Security and Justice.

9. REFERENCES

[1] Stucco: Situation and Threat Understanding by Correlating Contextual Observations. `https://stucco.github.io/`, 2015.

[2] T. Aslam, I. Krsul, and E. H. Spafford. Use of a taxonomy of security faults. 1996.

[3] S. Barnum. Standardizing cyber threat intelligence information with the structured threat information expression (stix). *MITRE Corporation*, page 11, 2014.

[4] C. Blanco, J. Lasheras, R. Valencia-García, E. Fernández-Medina, A. Toval, and M. Piattini. A systematic review and comparison of security ontologies. In *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on*, pages 813–820. IEEE, 2008.

[5] R. A. Bridges, C. L. Jones, M. D. Iannacone, K. M. Testa, and J. R. Goodall. Automatic labeling for entity extraction in cyber security. *arXiv preprint arXiv:1308.4941*, 2013.

[6] G. A. Fink, V. Duggirala, R. Correa, and C. North. Bridging the host-network divide: Survey, taxonomy, and solution. In *LISA*, pages 247–262, 2006.

[7] S. Hansman and R. Hunt. A taxonomy of network and computer attacks. *Computers & Security*, 24(1):31–43, 2005.

[8] J. D. Howard and T. A. Longstaff. A common language for computer security incidents. *Sandia National Laboratories*, 1998.

[9] V. Igure and R. Williams. Taxonomies of attacks and vulnerabilities in computer systems. *Communications Surveys & Tutorials, IEEE*, 10(1):6–19, 2008.

[10] C. L. Jones, R. A. Bridges, K. M. T. Huffer, and J. R. Goodall. Towards a relation extraction framework for cyber-security concepts. In *Proceedings of the CISRC-10, the tenth Cyber & Information Security Research Conference*. ACM, 2015.

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

[12] K. S. Killourhy, R. A. Maxion, and K. M. Tan. A defense-centric taxonomy based on attack manifestations. In *Dependable Systems and Networks, 2004 International Conference on*, pages 102–111. IEEE, 2004.

[13] C. E. Landwehr, A. R. Bull, J. P. McDermott, and W. S. Choi. A taxonomy of computer program security flaws. *ACM Computing Surveys (CSUR)*, 26(3):211–254, 1994.

[14] U. Lindqvist and E. Jonsson. How to systematically classify computer security intrusions. In *Security and Privacy, 1997. Proceedings., 1997 IEEE Symposium on*, pages 154–163. IEEE, 1997.

[15] N. McNeil, R. Bridges, M. Iannacone, B. Czejdo, N. Perez, and J. Goodall. Pace: Pattern accurate computationally efficient bootstrapping for timely discovery of cyber-security concepts. In *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*, volume 2, pages 60–65. Dec 2013.

[16] C. Meyers, S. Powers, and D. Faissol. Taxonomies of cyber adversaries and attacks: a survey of incidents and approaches. *Lawrence Livermore National Laboratory*, 7, 2009.

[17] 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*, pages 75–81. IEEE, 2012.

[18] V. Mulwad, W. Li, A. Joshi, T. Finin, and K. Viswanathan. 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, pages 257–260. IEEE, 2011.

[19] L. Obrst, P. Chase, and R. Markeloff. Developing an ontology of the cyber security domain. In *STIDS*, pages 49–56, 2012.

[20] M. C. Parmelee. Toward an ontology architecture for cyber-security standards. *STIDS*, 713:116–123, 2010.

[21] R. C. Seacord and A. D. Householder. A structured approach to classifying security vulnerabilities. Technical report, DTIC Document, 2005.

[22] J. Undercoffer, A. Joshi, and J. Pinkston. Modeling computer attacks: An ontology for intrusion detection. In *Recent Advances in Intrusion Detection*, pages 113–135. Springer, 2003.

[23] S. Weber, P. A. Karger, and A. Paradkar. A software flaw taxonomy: aiming tools at security. In *ACM SIGSOFT Software Engineering Notes*, volume 30, pages 1–7. ACM, 2005.