Developing an Ontology of the Cyber Security Domain

Leo Obrst[a], Penny Chase[b], Richard Markeloff[a]

The MITRE Corporation

[a]McLean, VA

[b]Bedford, MA

{lobrst, pc, rmarkeloff}@mitre.org

***Abstract*— This paper reports on a trade study we performed to support the development of a Cyber ontology from an initial malware ontology. The goals of the Cyber ontology effort are first described, followed by a discussion of the ontology development methodology used. The main body of the paper then follows, which is a description of the potential ontologies and standards that could be utilized to extend the Cyber ontology from its initially constrained malware focus. These resources include, in particular, Cyber and malware standards, schemas, and terminologies that directly contributed to the initial malware ontology effort. Other resources are upper (sometimes called 'foundational') ontologies. Core concepts that any Cyber ontology will extend have already been identified and rigorously defined in these foundational ontologies. However, for lack of space, this section is profoundly reduced. In addition, utility ontologies that are focused on time, geospatial, person, events, and network operations are briefly described. These utility ontologies can be viewed as specialized super-domain or even mid-level ontologies, since they span many, if not most, ontologies -- including any Cyber ontology. An overall view of the ontological architecture used by the trade study is also given. The report on the trade study concludes with some proposed next steps in the iterative evolution of the Cyber ontology.**

***Index Terms*—ontology, malware, cyber, trade study.**

I. Introduction

This report is a trade study to support the development of a Cyber ontology. In this section we present the goals of both the Cyber ontology effort and this report. The following sections discuss the ontology development methodology and various ontologies and standards that could be utilized to extend the Cyber ontology. This report concludes with some proposed next steps in the iterative evolution of the Cyber ontology.

The ultimate goal of this effort is to develop an ontology of the cyber security domain, expressed in the OWL language, that will enable data integration across disparate data sources. Formally defined semantics will make it possible to execute precise searches and complex queries. Initially, this effort is focused on malware. Malware is one of the most prevalent threats to cyber security, and the MITRE team's work on the Malware Attribute Enumeration and Characterization (MAEC) language [1] provides a store of knowledge that can be readily leveraged.

As the scope of the ontology expands, the underlying conceptual framework will be provided by the Diamond Model of malicious activity [2], shown in Figure 1. The four corners of the diamond, Victim, Infrastructure, Capability, and Actor (the one threatening the victim), account for all the major dimensions of a malicious cyber threat.

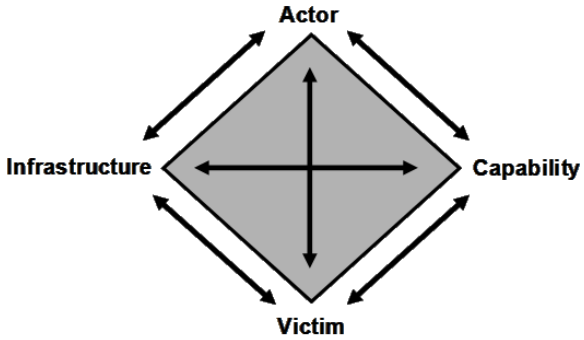

Fig. 1. The Diamond Model of malicious activity (from [2]).

The primary goals of this document are to explain the process followed in developing the Cyber ontology and catalog the sources upon which it is based. A secondary goal is to provide a compilation of resources useful for constructing semantic models in the cyber security domain.

II. Ontology Development Methodology

This section identifies the general methodology employed in the ontology development process, along with the specific methodology used to develop the Cyber ontology.

A. General Methodology

In general, the ontology development methodology employed here is called a "middle-out" approach. This means

that it contains aspects of top-down analysis and bottom-up analysis. Bottom-up analysis requires understanding the semantics of the underlying data sources which are to be integrated. Top-down analysis requires understanding the semantics of the end-users who will actually use the resulting ontology-informed, semantically integrated set of data sources, i.e., the kinds of questions those end-users want to ask or could ask, given the enhanced capabilities resulting from the semantic integration of those data sources (e.g., questions that require temporal integration or reasoning, as over integrated timelines of events). See references [3-8].

These kinds of analyses result in the development of competency questions [7, 8]. These are the questions that need to be asked of the ontology in order to provide the targeted value to the users. As such, these questions can be viewed as the queries that need to be executed. These queries, in turn, can be viewed as a test procedure that indicates when the ontology development is sufficiently complete for a given stage of development, i.e., when those queries return results that are accurate, sufficiently rich, and at the right level of granularity as judged by a subject matter expert (SME).

Capturing the right competency questions is part of the requirements analysis phase of ontology development. These help identify use cases and scenarios. Taken together, the competency questions, uses cases, and scenarios enable the requirements to be fleshed out.

The key to ontology development here is of course an understanding of the cyber domain, which drives the kinds of entities, properties, relationships, and potentially rules that will be needed in the ontology.

B. Specific Methodology

More specifically, the methodology used for the current ontology development is based on the following principles, focused on parsimony and reuse:

Reuse of existing ontologies: Existing ontologies are reused where possible. The methodology of reuse consists of the following steps:

A. Establish the base of possible existing ontologies in the domain areas of interest, including foundational, mid-level, utility, and reference ontologies.

B. When developing the current Cyber ontology, incorporate classes and properties (and definitions) that exist in the best of the ontologies of (A).

C. When the number of classes and properties incorporated from a given ontology of (A) into the Cyber ontology grows large, consider directly importing the given ontology into the Cyber ontology, and establishing equivalence relations between the classes of the (A) ontology and the classes of the Cyber ontology.

Harvesting of existing schemas, data dictionaries, glossaries, standards: Other structured and definitional resources are used when available, as a form of knowledge acquisition of the domain. These resources are analyzed for the kinds of entities, relationships, properties, attributes, and the range of values for those, expressed in the resource. Where it makes sense, and as correlated with other Cyber database schemas and expressed analyst questions and interests (and their decompositions), these entities, relationships, properties, and values are incorporated into the Cyber ontology, after refinement according to ontological engineering principles.

Keeping it simpler: Where possible, the simpler ontological approach is chosen. This can mean that, for example, where the choice is between a 4-D spacetime or a 3-D space and time conceptualization, the 3-D conceptualization is chosen because it is generally simpler for non-ontologists to understand.

C. Cyber Ontology Architecture

The final product of the ontology development methodology described above will be an ontology that consists of a number of modular sub-ontologies, rather than a single, monolithic ontology. Ontologies can be grouped into three broad categories of upper, mid-level and domain ontologies, according to their levels of abstraction [9]:

- Upper ontologies are high-level, domain-independent ontologies that provide common knowledge bases from which more domain-specific ontologies may be derived. Standard upper ontologies are also referred to as foundational or universal ontologies.
- Mid-level ontologies are less abstract and make assertions that span multiple domain ontologies. These ontologies may provide more concrete representations of abstract concepts found in the upper ontology. There is no clear demarcation point between upper and mid-level. Mid-level ontologies also encompass the set of ontologies that represent commonly used concepts, such as Time and Location. These commonly used ontologies are sometimes referred to as utility ontologies [10].
- Doman ontologies specify concepts particular to a domain of interest and represent those concepts and their relationships from a domain specific perspective. Domain ontologies may be composed by importing mid-level ontologies. They may also extend concepts defined in mid-level or upper ontologies.

These categories and their roles in ontology architecture are shown in Figure 2, reproduced from [9]. A further discussion can be found in [10].

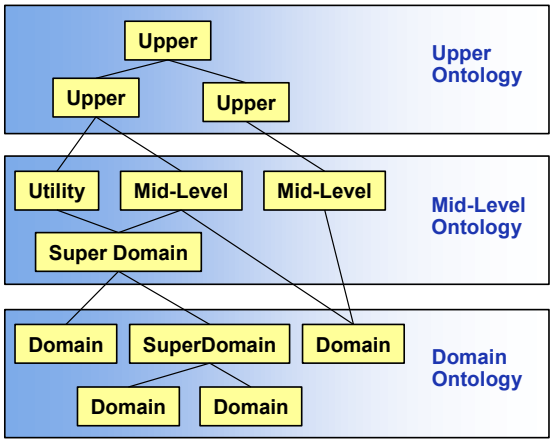

Fig. 2. Ontology architecture

Figure 3 depicts the expected architecture of the Cyber ontology. Each rounded box represents a major category of concepts. These concepts can be arranged along a level of abstraction continuum from broad and general to domain-specific. The larger bounding boxes represent separate ontologies that span multiple concept categories. The ontologies shown in Figure 3 and the sources they are based on are described in the following section.

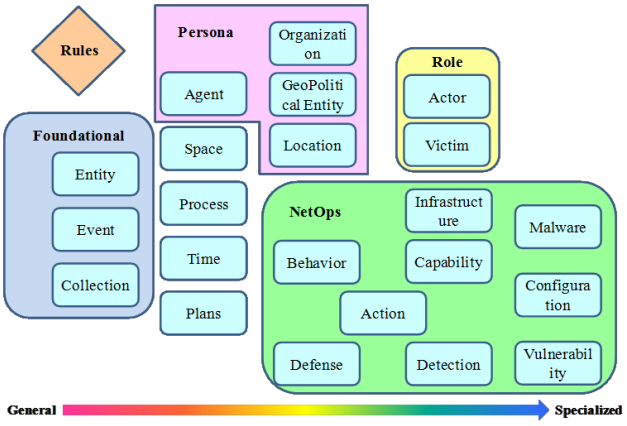

Fig 3. The Cyber ontology architecture

III. Resources for the Malware and Cyber Ontologies: Ontologies, Schemas, and Standards

There exist a variety of resources that can lay the groundwork for a Cyber ontology. This section presents a survey of those resources that we consider to be particularly applicable and important. These are not limited to ontologies, but also include taxonomies, lexica, and schemas.

A. Malware Resources

Published attempts to systematically categorize malware include one ontology [11] and three descriptive languages implemented in XML [1, 12, 13]. Also worthy of mention is an attempt at categorizing malware traits [14].

XML is a technology for defining text documents for information exchange, and the structure and content of a particular type of XML document is dictated by an XML schema. XML schemas offer enumerations of concepts and shared vocabularies for specific domains that can be useful as a basis for ontology development. However, XML schemas do not define formal semantics for the terms they contain, and are therefore not equivalent to ontologies.

1) Swimmer's Ontology of Malware Classes

A paper by Morton Swimmer [11] is the only non-trivial attempt to construct an ontological model of malware that we could identify. Swimmer's ontology is intended to enable data exchange between security software products. Swimmer's taxonomy of malware classes is shown in Figure 4.

Swimmer's malware class hierarchy is relatively simple. It organizes malware into well-known categories such as Trojan horse, virus, and worm. This may not be useful for malware instances that exhibit either behaviors from multiple classes or novel behaviors not associated with any recognized class.

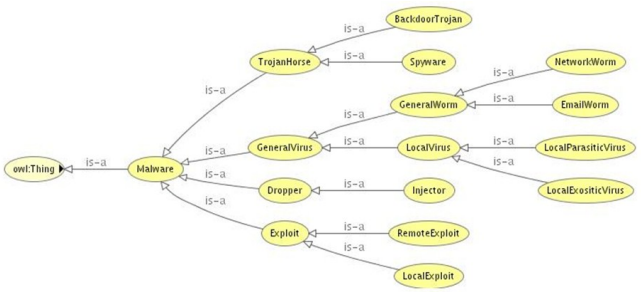

Fig. 4. Swimmer's malware class hierarchy (from [11]).

In Swimmer's taxonomy of malware characteristics, all malware characteristics belong to one of three high-level classes:

- Payload. This is assumed to be programmed with malicious intent.
- Vector. This defines how the malware is deployed or spread.
- Obfuscation. Characteristics for evading detection.

In describing vector characteristics, Swimmer coins the term "insituacy" to mean "the state the Malware strives to be in through its actions".

2) MAEC: Malware Attribute and Enumeration Characterization

MAEC is intended as a language for addressing all known types, variants, and manifestations of malware. Current signature-based malware detection techniques identify malware using a single metadata entity (e.g., a file hash), and MAEC's primary goal is to provide a more flexible method for characterizing malware based on patterns of attributes such as behaviors, artifacts, and attack patterns. This stands in contrast with Swimmer's work, which is focused on predefined malware families and discernible intent.

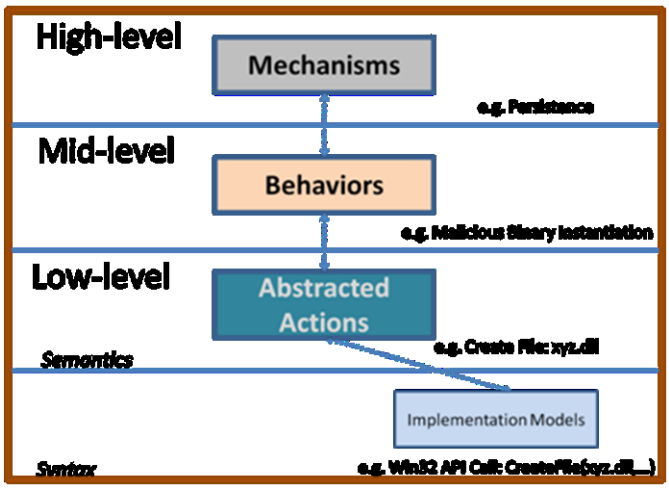

Fig. 5. The MAEC architecture

MAEC has a tiered architecture, as shown in Figure 5. At its lowest level, MAEC strives to portray what an instance of malware does by describing its actions, such as hardware accesses and system state changes. A distinction is drawn between semantics and syntactics by abstracting actions away from their implementations. This facilitates correlation between

malware instances that do similar things at a low-level but with different implementations (such as malware targeted at different platforms).

MAEC's middle level describes malware behaviors. Behaviors serve to organize and define the purpose behind low-level actions, whether in groups or as singletons. Behaviors can represent discrete components of malware functionality at a level that is useful for analysis, triage, detection, etc.

MAEC's top level summarizes malware in terms of its mechanisms. Mechanisms are organized groups of behaviors. Some examples would be propagation, insertion, and self-defense. Since there is likely a low upper bound on the number of possible mechanisms, they can be useful in understanding the composition of malware at a very high level.

There are other resources such as the Industry Connections Security Group (ICSG) Malware Metadata Exchange Format [12], and Zeltser's Categories of Common Malware Traits [14], which space limitations preclude us from elaborating.

B. *Languages for Cyber Security Incidents*

Howard and Longstaff's seminal work [15] represents an early attempt to establish a common language for describing computer and network security incidents. Since then, industry and standards organizations have promulgated several languages for describing computer and network security incidents. Some of the prominent ones are described below. These languages all share the goal of facilitating information sharing across the cyber security community.

OpenIOC is an XML format for sharing intelligence related to cyber security incidents. Intelligence is organized as Indicators of Compromise (IOCs), which represent patterns that suggest malicious activity. OpenIOC has been developed by MANDIANT [13] and offered as an open standard. MANDIANT's products are widely used by defense contractors, and consistency with OpenIOC facilitates processing information from the Defense Industrial Base (DIB). OpenIOC includes around 30 separate XML schemas that describe various classes of objects that can be used to detect suspicious activity, such as MD5 hashes, registry keys, IP addresses, etc. The OpenIOC schemas are probably the most comprehensive descriptions of these types of objects available. The MAEC team incorporated the OpenIOC objects into MAEC and subsequently the OpenIOC objects formed the starting point for CybOX objects (CybOX is discussed in Section III.H).

IODEF [16] is a specification, in the form of an XML schema, developed by the IETF Extended Incident Handling (INCH) Working Group of the Internet Engineering Task Force (IETF) [17]. IODEF is an information exchange format for Computer Security Incident Response Teams (CSIRTs). It also provides a basis for the development of interoperable tools and procedures for incident reporting.

The VERIS framework [18] is used by Verizon Business [19] to collect security incident data from anyone who volunteers to submit it. These data are collected using a Web application [20]. The goal is to collect data of sufficient quantity and quality to support statistical analyses. Verizon's data collection is based on what they refer to as the A4 Threat Model. In this model, security incidents are regarded as a series of events where an organization's information assets are adversely affected. These events have four descriptive dimensions:

- Agent: Whose actions affected the asset
- Action: What actions affected the asset
- Asset: Which assets were affected
- Attribute: How the asset was affected.

The details of the VERIS model are available online in a Wiki format [18].

C. *Attack Patterns and Process Models*

The literature offers a number of attempts to create taxonomies and conceptual models of cyber attacks and attack patterns. Howard and Longstaff's [15] attack model is shown in Figure 6. In their model, an attacker uses a tool to exploit a vulnerability. This produces an action on a target (which together comprises an event). The intention is to accomplish an unauthorized result.

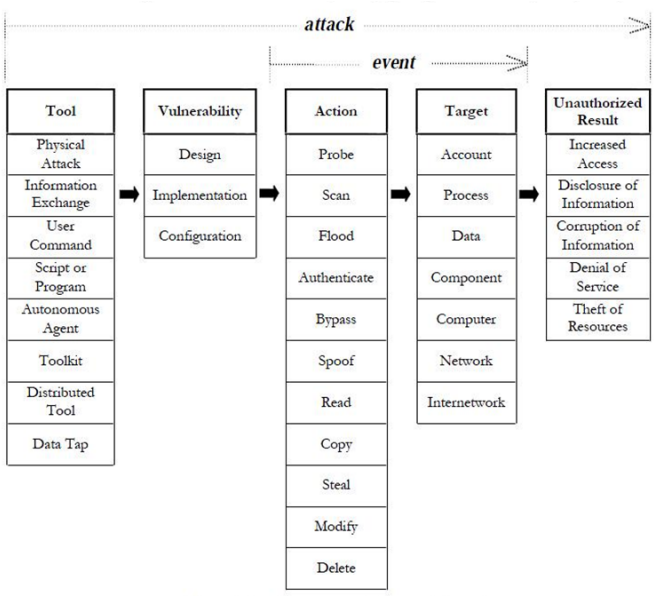

Fig. 6. Howard and Longstaff's model of computer and network attacks (from [15]).

A more recent work in a similar vein [21], presented at the 2007 IEEE International Symposium on Network Computing and Applications, delineates a model for the attack process that consists of the following phases:

- Reconnaissance. The search for information about potential victims.
- Gain Access. Gaining access, at the desired level, to a victim's system.
- Privilege Escalation. Escalate the initial privilege level, as necessary.
- Victim Exploration. Gaining knowledge of the victim's system, including browsing files, searching user accounts, identifying hardware, identifying installed program, and searching trusted hosts.

- Principal actions. Taking steps to accomplish the ultimate objective of the attack, such as installing malicious software or compromising data integrity.

This model is shown in flowchart form in Figure 7, reproduced from [21].

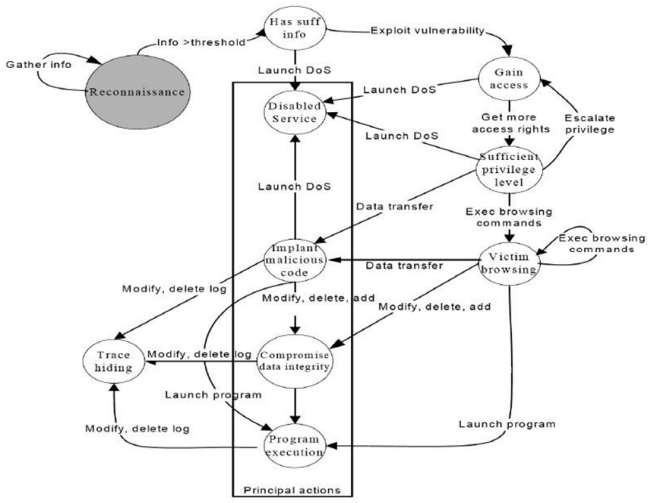

FIG. 7. A proposed attack process model (from [21]).

Relevant discussions of attack phases can also be found in blog postings by Bejtlich [22] and Cloppert [23].

The CAPEC catalog [24] defines a taxonomy of attack patterns. The CAPEC catalog currently contains 68 categories and 400 attack patterns. Attack patterns are modeled after object-oriented design patterns, and by design they exclude low-level implementation details. Categories are containers for related attack patterns. The patterns are more or less aligned with the top two MAEC layers, and categories roughly correspond to MAEC mechanisms.

The WASC Threat Classification [25] is similar to CAPEC.

D. Foundational Ontologies for the Cyber Ontology

Modeling choices are made in the development of foundational ontologies that have a downward impact on mid-level and domain ontologies. We cannot describe some of these ontological choices here, but invite the reader to see [9].

There are several foundational ontologies that could be considered for use in the Cyber ontology. These range from Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [26], Basic Formal Ontology (BFO) [27], Object-Centered High-Level REference ontology (OCHRE) [28], Generic Formal Ontology (GFO) [29], Suggested Upper Merged Ontology (SUMO) [30], Unified Foundational Ontology (UFO) [31, 32], and Cyc/OpenCyc [33-35].

E. Utility Ontologies

The Cyber ontology will necessarily include concepts from domains that transcend cyber security, such as notions concerning people, time, space, and events. Where possible, the Cyber ontology will import existing ontologies to provide descriptions of these concepts. In this section we very briefly catalog the utility ontologies that we would consider for inclusion in the Cyber ontology.

1) Persons

Modeling the Actor and Victim nodes in Figure 1-1 will entail an ontological description of persons, their social roles and relationships, and their relationships to things. Among the available ontologies that might address this need, we include Friend Of A Friend (FOAF) [36], DOLCE Social Objects [37] which includes social roles and organizations.

2) Time

The Cyber ontology will need to be able to express notions of time instances and intervals, as well as concepts related to clock and calendar time. Various theories of the structure of time have been proposed; see [38] for a survey. Of particular interest is Allen's Interval Algebra for temporal reasoning [39]. Allen's calculus defines 13 basic relations between two time intervals.

There are two W3C standard ontologies of temporal concepts, OWL-Time [40] and time-entry [41]. They both provide similar vocabularies for expressing facts about temporal intervals and instants, while time-entry also includes the concept of an event. Both ontologies contain object properties that implement the Allen relations. Also included in the ontologies are classes and relations for expressing intervals and instants in clock and calendar terms. Both ontologies include the concept of a time zone, and a separate global time zone ontology is available [42].

3) Geospatial

The Cyber ontology may require geospatial concepts to describe the physical locations of people or infrastructure. See [43] for a comprehensive survey of available geospatial ontologies. Another source of information about geospatial ontologies is the Spatial Ontology Community of Practice (SOCoP) [44]. SOCoP is chartered as a Community of Practice under the Best Practices Committee of the Federal CIO Council.

The two-dimensional analog to Allen's Interval Algebra for qualitative spatial representation is the Region Connection Calculus 8 (RCC-8) [45], so named because eight basic relations comprise the calculus. RCC theory can be extended to support reasoning about regions with indeterminate boundaries [46].

If it is the case that a significant portion of the geospatial information to be described by the Cyber ontology is in the form of text mentions of place names, then the GeoNames Ontology [47] may be suitable for inclusion in the ontology. Although GeoNames does not support RCC-8, it has relations such as locatedIn, nearby, and neighbor. It is accompanied by a knowledge base containing 140 million assertions about 7.5 million geographical objects that span the globe. A typical use for GeoNames is to infer what country a given town, city, or region is located in.

F. Events and Situations

Events are entities that describe the occurrences of actions and changes in the real world. Situations represent histories of action occurrences. In this context at least, situations are not equivalent to states. Events and situations are dynamic and challenging to model in knowledge representation systems.

As in the temporal and spatial domains, logic formalisms have been created for representing and reasoning about events and situations. These are the event calculus [48] and situation calculus [49]. Both calculi employ the notion of fluents. A fluent is a condition that can change over time. The main elements of the event calculus are fluents and actions, and for the situation calculus they are fluents, actions and situations.

Notions of events and situations are included in several of the ontologies previously described. DOLCE, GFO, Cyc, and time-entry all have Event classes. GFO has a class named History that corresponds to the concept of a situation, and Cyc has a Situation class. BFO's ProcessualEntity class has subclasses that correspond closely to events and situations.

Ontologies for events and situations include a DOLCE extension for descriptions and situations [50], a proposed upper event ontology [51], and an ontology for Linking Open Descriptions of Events (LODE) [52].

G. Network Operations

A network operations (NetOps) OWL ontology was developed in 2009 by MITRE as part of the data strategy effort supporting the NetOps Community of Interest (COI). The NetOps ontology includes entities and events, and represents mission threads of interest to US federal government network management.

H. Other Cyber Resources

There are a number of other resources that can be mined for concepts, abstractions, and relationships between entities that may be suitable for inclusion in a Cyber ontology.

Common Event Expression (CEE) [53] is intended to standardize the way computer events are described, logged, and exchanged. Some of these events would naturally correspond to malware actions and behaviors. The CEE components most relevant to cyber security ontology development are the Common Dictionary and Event Expression Taxonomy (CDET). The dictionary defines a collection of event fields and field value types that are used throughout CEE to specify the values of properties associated with specific events. The taxonomy specifies event types. Examples of event types are user login, service restart, network connection, privilege elevation, and account creation.

A recent foundational schema for the cyber domain is Cyber Observable Expression (CybOX) [54]. CybOX is designed for the specification, capture, characterization and communication of events or stateful properties observable in the cyber domain in support of a wide range of use cases. MAEC and CEE both leverage CybOX for describing cyber objects, actions, and events. An emerging schema is the Structured Threat Information Expression (STIX) [55], which provides an overarching framework for describing threat information, including adversaries, tactics, techniques and procedures (TTPs), incidents, indicators, vulnerabilities, and courses of actions. Malware is included under the heading of TTPs. STIX references other schemas and cyber information, including MAEC, CybOX, CVE, and CPE.

Security Content Automation Protocol (SCAP) [56] is a suite of specifications that standardize the format and nomenclature by which security software products communicate software flaw and security configuration information. In its current incarnation [57], SCAP is comprised of seven specifications:

- eXtensible Configuration Checklist Description Format (XCCDF) [58], a language for authoring security checklists/benchmarks and for reporting results of checklist evaluation.
- Open Vulnerability and Assessment Language (OVAL) [59], a language for representing system configuration information, assessing machine state, and reporting assessment results.
- Open Checklist Interactive Language (OCIL) [60], a framework for expressing a set of questions to be presented to a user and corresponding procedures for interpreting responses to these questions.
- Common Platform Enumeration (CPE) [61], a nomenclature and dictionary of hardware, operating systems, and applications.
- Common Configuration Enumeration (CCE) [62], a nomenclature and dictionary of security software configurations.
- Common Vulnerabilities and Exposures (CVE) [63], a nomenclature and dictionary of security-related software flaws.
- Common Vulnerability Scoring System (CVSS) [64], an open specification for measuring the relative severity of software flaw vulnerabilities

Of these standards, the ones most germane to developing a Cyber ontology would be OVAL, CPE, CCE and CVE. Parmelee [65] has outlined a semantic framework for these four standards built upon loosely-coupled modular ontologies. Parmelee's framework is intended to simplify data interoperability across automated security systems based on the OVAL, CPE, CCE and CVE standards.

IV. Cyber Ontology Development: Next Steps

The current Cyber ontology is focused primarily on malware and some preliminary aspects of the so-called 'diamond model', which includes actors, victims, infrastructure, and capabilities. Necessarily, more of the infrastructure and capabilities were developed first; however, even these are not yet developed to the level of detail that is warranted, i.e., expanding on behavioral aspects and events, in

particular that are the core of Cyber, would make it more useful. These are our next steps.

ACKNOWLEDGMENT

REFERENCES

[1] MAEC - Malware Attribute Enumeration and Characterization. [Online] http://maec.mitre.org/.

[2] Ingle, J. Organizing Intelligence to Respond to Network Intrusions and Attacks. *Briefing for the DoD Information Assurance Symposium.* Nashville, TN, 2010.

[3] Fernandéz, M., Gómez-Pérez, A. and and Juristo, N. METHONTOLOGY: From Ontological Art to Ontological Engineering. *AAAI97 Workshop on Ontological Engineering, Spring Symposium Series*. Stanford University, 1997. pp. 33-40.

[4] Fernández M. et al. Building a Chemical Ontology Using Methontology and the Ontology Design Environment. *IEEE Intelligent Systems.* January/February 1999. Vol. 14, 1. http://www.aifb.uni-karlsruhe.de/Lehrangebot/Sommer2001/SemanticWeb/papers/chemical_ontology.pdf.

[5] Fernández, M. Overview of Methodologies for Building Ontologies. Workshop on Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends. (IJCAI99). August 1996.

[6] Gómez-Pérez, A., Fernández, M. and de Vicente, A. Towards a Method to Conceptualize Domain Ontologies. *ECAI '96Workshop on Ontological Engineering.* Budapest, Hungary : s.n., 1996. pp. 41-52.

[7] Gruninger, M. and Fox, M. S. Methodology for the design and evaluation of ontologies. Montreal, 1995.

[8] Uschold, M. and Gruninger, M. Ontologies: Principles, Methods, and Applications. 1996. Vol. 11, 2, pp. 93-136.

[9] Obrst, L. Ontological Architectures. [ed.] Johanna Seibt, Achilles Kameas Roberto Poli. Chapter 2 in Part One: Ontology as Technology in the book: TAO – Theory and Applications of Ontology, Volume 2: Computer Applications. Springer, 2010.

[10] Semy, S., Pulvermacher, M. and Obrst, L. Toward the Use of an Upper Ontology for U.S. Government and U.S. Military Domains: An Evaluation. *MITRE Technical Report, MTR 04B0000063.* November 2005.

[11] Swimmer, M. Towards An Ontology of Malware Classes. [Online] January 27, 2008. http://www.scribd.com/doc/24058261/Towards-an-Ontology-of-Malware-Classes.

[12] IEEE-SA - Industry Connections. [Online] http://standards.ieee.org/develop/indconn/icsg/malware.html.

[13] MANDIANT: Intelligent Information Security. [Online] http://www.mandiant.com.

[14] Zeltser, L. Categories of Common Malware Traits. *Internet Storm Center Handler's Diairy.* [Online] Sept. 25, 2009. http://isc.sans.edu/diary.html?storyid=7186.

[15] Howard, J. D. and Longstaff, T. A Common Language for Computer Security Incidents. [Technical Report]. Sandia National Laboratories, 1998.

[16] Cover Pages Incident Object Description and Exchange Format (IODEF). [Online] http://xml.coverpages.org/iodef.html.

[17] Internet Engineering Task Force. [Online] http://www.ietf.org/.

[18] VERIS Framework. [Online] https://verisframework.wiki.zoho.com/.

[19] Verizon Business. [Online] http://www.verizonbusiness.com/.

[20] Verizon Incident Classification and Reporting. [Online] https://www2.icsalabs.com/veris/incidents/new#/welcome.

[21] Gadelrab, M., El Kala, A. and Deswarte, Y. Execution Patterns in Automatic Malware and Human-Centric Attacks. *IEEE International Symposium on Network Computing and Applications.* 2008.

[22] Bejtlich, R. TaoSecurity: Incident Phases of Compromise. [Online] June 6, 2009. http://taosecurity.blogspot.com/2009/06/incident-phases-of-compromise.html.

[23] Cloppert, M. [Online] Oct. 14, 2009. http://computer-forensics.sans.org/blog/2009/10/14/security-intelligence-attacking-the-kill-chain/.

[24] CAPEC - Common Attack Pattern Enumeration and Characterization. [Online] http://capec.mitre.org/.

[25] The Web Application Security Consortium/Threat Classification. [Online] http://projects.webappsec.org/w/page/13246978/Threat-Classification.

[26] Laboratory for Applied Ontology - DOLCE. [Online] http://www.loa-cnr.it/DOLCE.html.

[27] Basic Formal Ontology (BFO). [Online] http://www.ifomis.org/bfo.

[28] Schneider, L. How to Build a Foundational Ontology -- The Object-Centered High-level Reference Ontology OCHRE. *Proceedings OF THE 26TH Annual German Conference on AI, KI 2003: Advances In Artificial Intelligence* . 2003.

[29] General Formal Ontology (GFO). [Online] http://www.onto-med.de/ontologies/gfo/.

[30] Niles, I., and Pease, A. Towards a Standard Upper Ontology. [ed.] Chris Welty and Barry Smith. Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001). 2001.

[31] Guizzardi, G., Wagner, G. Some Applications of a Unified Foundational Ontology in Business. [ed.] Michael Rosemann and Peter Green. *Ontologies and Business Systems Analysis.* IDEA Publisher, 2005.

[32] Guizzardi, G., Wagner, G. Towards Ontological Foundations for Agent Modeling Concepts using UFO. Agent-Oriented Information Systems (AOIS), selected revised papers of the Sixth International Bi-Conference Workshop on Agent-Oriented Information Systems. Springer-Verlag, 2005.

[33] Cycorp, Inc. [Online] http://cyc.com/cyc/technology/whatiscyc_dir/whatsincyc.

[34] Cycorp, Inc. [Online] http://cyc.com/cyc.

[35] OpenCyc.org. [Online] http://www.opencyc.org/.

[36] The Friend of a Friend (FOAF) project. [Online] http://www.foaf-project.org/.

[37] Masolo, C. et al. Social Roles and their Descriptions. *Proceedings of KR'2004.* 2004. pp. 267-277.

[38] Hayes, P. A Catalog of Temporal Theories. *Technical Report UIUC-BI-AI-96-01.* s.l. : Univerisity of Illinois, 1996.

[39] Allen, J. F. Maintaining knowledge about temporal intervals. *Communications of the ACM.* 1983.

[40] Hobbs, J. R. and Pan, F. An Ontology of Time for the Semantic Web. CM Transactions on Asian Language Processing (TALIP): Special issue on Temporal Information Processing. 2004. Vol. 3, 1, pp. 66-85.

[41] Pan, F. and Hobbs, J. R. Time in OWL-S. *Proceedings of the AAAI Spring Symposium on Semantic Web Services.* s.l. : Stanford University, 2004. pp. 29-36.

[42] A Time Zone Resource in OWL. [Online] http://www.isi.edu/~hobbs/timezonehomepage.html.
[43] Ressler, J., Dean, M. and Kolas, D. Geospatial Ontology Trade Study. [ed.] Terry Janssen, Werner Ceuster Leo Obrst. *Ontologies and Semantic Technologies for Intelligence.* Amsterdam, Berlin, Tokyo, Washington D.C. : IOS Press, 2010, Chapter 11, pp. 179-212.
[44] Spatial Ontology Community of Practice (SOCoP). [Online] http://www.socop.org/.
[45] Randall, D., Cui, Z. and and Cohn, A. A spatial logic based on regions and connection. *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning.* Cambridge, MA, 1992. pp. 165-176.
[46] Gotts, A. Cohn and N. The 'Egg-Yolk' representation of regions with indeterminate boundaries. [ed.] P. Burrough and A. M. Frank. *Proceedings, GISDATA Specialist Meeting on Geographical Objects with Undetermined Boundaries.* Francis Taylor, 1996. pp. 171-187.
[47] GeoNames Ontology - Geo Semantic Web. [Online] http://www.geonames.org/ontology/documentation.html.
[48] Kowalski, R. and Sergot, M. A Logic-based Calculus of Events. *New Generation Computing* . 1986. Vol. 4, pp. 67–95.
[49] Reiter, R. The frame problem in the situation calculus: a simple solution (sometimes) and a completeness result for goal regression. [ed.] Vladimir Lifshitz. *Artificial intelligence and mathematical theory of computation: papers in honour of John McCarthy.* San Diego, CA : Academic Press Professional, Inc., 1991. pp. 359-380.
[50] Gangemi, A. and Mika, P. Understanding the Semantic Web through Descriptions and Situations. *Proceedings of CoopIS/DOA/ODBASE.* 2003. pp. 689-706.
[51] Kaneiwa1, K. Iwazume, M. and Fukuda, K. An upper ontology for event classifications and relations. *AI'07 Proceedings of the 20th Australian joint conference on Advances in artificial intelligence* . 2007.
[52] LODE: Linking Open Descriptions of Events. [Online] http://escholarship.org/uc/item/4pd6b5mh.
[53] Common Event Expression: CEE, A Standard Log Language for Event Interoperability in Electronic Systems. [Online] http://cee.mitre.org/.
[54] CybOX – Cyber Observable Expression. [Online] http://cybox.mitre.org/
[55] STIX-whitepaper. [Online] http://measurablesecurity.mitre.org/docs/STIX-Whitepaper.pdf
[56] The Security Content Automation Protocol (SCAP) - NIST. [Online] http://scap.nist.gov/.
[57] Quinn, Waltermire, Johnson, Scarfone, Banghart. The Technical Specification for the Security Content Automation Protocol (SCAP): SCAP Version 1.1 (DRAFT). Gaithersburg, MD : NIST, 2011. SP800-126.
[58] XCCDF - The eXtensible Configuration Checklist Description Format - The Security Content Automation Protocol (SCAP) - NIST. [Online] http://scap.nist.gov/specifications/xccdf/.
[59] OVAL - Open Vulnerability and Assessment Language. [Online] http://oval.mitre.org/.
[60] OCIL - The Open Checklist Interactive Language - The Security Content Automation Protocol (SCAP) - NIST. [Online] http://scap.nist.gov/specifications/ocil/.
[61] CPE - Common Platform Enumeration. [Online] http://cpe.mitre.org/.
[62] Common Configuration Enumeration (CCE): Unique Identifiers for Common System Configuration Issues. [Online] http://cce.mitre.org/.
[63] CVE - Common Vulnerabilities and Exposures. [Online] http://cve.mitre.org/.
[64] Common Vulnerability Scoring System (CVSS-SIG). [Online] http://www.first.org/cvss/.
[65] Parmelee, M. *Toward an Ontology Architecture for Cyber-Security Standards.* George Mason University, Fairfax, VA : Semantic Technologies for Intelligence, Defense, and Security (STIDS) 2010.