Toward a Knowledge Graph of Cybersecurity Countermeasures

Peter E. Kaloroumakis
The MITRE Corporation
Annapolis Junction, MD
pk@mitre.org

Michael J. Smith
The MITRE Corporation
Annapolis Junction, MD
smithmj@mitre.org

***Abstract*—This paper describes our research and development toward a precise, unambiguous, and information-dense knowledge graph of cybersecurity countermeasures. In project work for our sponsors we have repeatedly encountered the need for a model that can identify and precisely specify cybersecurity countermeasure components and capabilities. Furthermore, it is necessary that practitioners know not only what threats a capability claims to address, but, specifically how those threats are addressed from an engineering perspective, and under what circumstances the solution would work. This knowledge is essential to estimate operational applicability, vulnerabilities, and develop enterprise solutions comprising multiple capabilities. To address this recurring need in the near-term, we created D3FEND, a framework in which we encode a countermeasure knowledge base, but more specifically, a knowledge graph. The graph contains semantically rigorous types and relations that define both the key concepts in the cybersecurity countermeasure domain and the relations necessary to link those concepts to each other. We ground each of the concepts and relations to particular references in the cybersecurity literature. Numerous sources of research and development literature were analyzed, including a targeted sample of over 500 countermeasure patents drawn from the U.S. Patent Office corpus over the years 2001 to 2018. To demonstrate the value of this approach in practice, we describe how the graph supports queries that can inferentially map cybersecurity countermeasures to offensive TTPs. As part of a larger vision, we outline future D3FEND work to leverage the linked open data available on research literature and apply machine learning, in particular semi-supervised methods, to assist in maintaining the D3FEND knowledge graph over time. Finally, we welcome community feedback on D3FEND.**

***Index Terms*—countermeasures, cybersecurity, cyber defense, intrusion detection, knowledge acquisition, knowledge engineering, knowledge graph, linked data, network security, ontology, procedures, tactics, techniques, TTPs**

I. Introduction

The cybersecurity defense market comprises more than 5,000 companies [1]. More than 6,000 cybersecurity patent applications were filed in 2018 (Figure 1). Cyber defense teams also implement their own countermeasures to address what vendor products do not. These custom capabilities are often shared through open-source software communities. In a cycle of adaptation, countermeasures are rapidly developed in response to rapidly changing offensive techniques.

Funded by the United States Department of Defense. Approved for Public Release; Distribution Unlimited. Case 20-2034.

A *cybersecurity countermeasure* is any process or technology developed to negate or offset offensive cyber activities. It is not enough to understand what a countermeasure does—what it detects, what it prevents. We must understand how it does it. A security architect must understand their organization's countermeasures—precisely what they do, how they do it, and their limitations—if countermeasures are to be effectively employed. A red team conducting an exercise to identify security gaps must plan their engagement with expert knowledge of a countermeasure's functionality if they are to evade it. A venture capitalist considering a cybersecurity startup must understand what problem it is trying to solve, whether and how it has been solved before, and why the proposed solution is better or novel.

Existing cybersecurity knowledge bases do not explain with enough fidelity and structure what these countermeasures do to meet these needs, we reviewed prominent knowledge bases discussed in section II. Furthermore, no framework or model exists that has had its knowledge content sustained at the rate of change in the cybersecurity space. D3FEND™ establishes a fine-grained semantic model of countermeasures, their properties, relationships, and history of development. We have also defined a semantic model of a portion of MITRE's ATT&CK® framework [15] to represent the offensive TTPs with the same common, standardized semantic language (OWL DL). This enables us to incorporate ATT&CK by mapping its concepts directly to D3FEND's model of defensive techniques and artifacts. D3FEND provides a methodology for curating content into new knowledge and tying it to its source information in meaningful ways. Finally, this paper provides a research road map for harvesting and analyzing content at the industry's pace, using and extending the promising human language technologies and semi-supervised learning.

D3FEND's longer-term goals are to (1) create a sustainable knowledge framework for characterizing and relating cybersecurity countermeasure technology; and (2) accelerate knowledge discovery and acquisition efforts required to keep pace with technological changes in the cyber domain. The D3FEND knowledge graph we have constructed can be directly embedded within the much larger web of datasets available within the Linked Open Data Cloud [41]. These will be used to connect our knowledge to research literature, organizations, authors, inventors, and investors. We believe

the representation chosen also provides a strong foundation for research furthering automation, including machine-learning approaches.

This paper explains how we collected and analyzed data to produce the initial version of the model. In the following sections we discuss related work, our methodology, the resulting knowledge graph of countermeasure techniques, and finally our roadmap for future work.

II. Related Work

Related prior work includes early cybersecurity standards and formats, government and commercial cybersecurity threat frameworks and knowledge bases, commercial product taxonomies, and formal information modeling in cybersecurity and other domains.

A. MITRE-initiated Cybersecurity Standards and Formats

Over the past two decades, MITRE has developed standard languages and formats to capture cybersecurity information: Common Vulnerabilities and Exposures (CVE®) [2], Common Weakness Enumeration (CWE™) [3], Open Vulnerability and Assessment Language (OVAL®) [4], Common Platform Enumeration (CPE™) [5], Common Event Expression (CEE™) [6], Common Attack Pattern Enumeration and Classification (CAPEC™) [7], Malware Attribute Enumeration and Characterization (MAEC™) [8], and Cyber Observables (CybOX™) [9] languages. These shared vocabularies and disambiguating references are useful for cybersecurity practitioners to record and exchange cyber threat knowledge. CybOX, MAEC, and CAPEC introduced more taxonomic and relational information; their elements have been incorporated into the Structured Threat Information eXpression (STIX™) OASIS™ standard for Cyber Threat Intelligence (CTI). Together these provide references on which we build a detailed, explicit model of countermeasures.

B. Cybersecurity Threat Frameworks

The National Institute of Standards and Technology (NIST) created and updates the Cybersecurity Framework, which provides security guidance for organizations to align their cybersecurity activities to manage their cybersecurity risk around an Identify-Protect-Detect-Respond-Recover paradigm [10]. This paradigm is broader than the kill-chain models we discuss next, but it is focused on activity and organization rather than engineering or technology. NIST also maintains the National Vulnerability Database, the U.S. government repository of standards-based vulnerability management data for security controls enumerated in NIST 800-53 [11]. NIST maps these relevant security controls to the activities defined in the Cybersecurity Framework.

Kill-chain oriented and derived threat models have proved popular and effective. New models like D3FEND must relate to these to ensure rapid uptake by the cyber community and easy integration with existing resources. The U.S. Department of Defense Cyber Analysis and Review and U.S. Department of Homeland Security Cybersecurity Architecture Review (.govCAR) frameworks serve as threat-based tools for high-level characterization of cyber threats and mitigations, and a means to identify gaps in cyber portfolios and architectures [12]. The Office of the Director of National Intelligence has created the Cyber Threat Framework to serve as a shared lexicon to characterize and categorize cyber threat events [13]. The National Security Agency's Technical Cyber Threat Framework added extensive technical detail [14]. MITRE's ATT&CK has influenced and been influenced by these frameworks.

C. ATT&CK

In "Finding Cyber Threats with ATT&CK-Based Analytics," MITRE created an analytic development methodology focused on detecting post-compromise adversary behaviors [15]. This revolutionized the language security practitioners use to discuss their work. Vendors began to describe what specific adversary behaviors their products were able to detect, prevent, or monitor. ATT&CK primarily modeled adversary behavior via offensive techniques organized by the tactical objectives they support. ATT&CK has since amassed an online knowledge base of threat actor techniques [17].

ATT&CK™ knowledge (also encoded via STIX) is an especially critical counterpart to D3FEND. D3FEND focuses on countermeasures. The two can be related as explained in section IV-E.

The Cyber Analytics Repository (CAR) was a key product of MITRE's ATT&CK work. CAR identifies objects related to key operating system and processing events [18] with its CAR Data Model, it also catalogs MITRE developed analytics and maps them to specific ATT&CK techniques they were designed to detect. The initial D3FEND release incorporates the CAR analytics developed, but whereas CAR modeled data from endpoint telemetry software and the associated analytics from a Security Operations Center Operator's perspective, D3FEND models the countermeasure space from a hardware or software engineer's perspective. Therefore, D3FEND's scope is limited only by the scope of the cybersecurity technology space.

D. Commercial Product Taxonomies

Commercial product taxonomies tend to organize by form-factor or intended buyer rather than function. This is useful for understanding which vendors might be relevant but does not explain in enough detail how the products work or what they do. However, there are implicit sets of features or functions that each grouping represents. Cybersecurity marketing vocabulary changes at such a pace that analyst firms produce new taxonomies each year despite only modest changes in the underlying technical functionality [19].

Patent systems also provide taxonomies of technological innovation and assign one or more of these codes to individual patents. The International Patent Classification system of 70,000 codes, and its extension, the Cooperative Patent Classification system, are used to curate patents and are managed by the European and U.S. patent offices. In the cybersecurity

domain, the categories are often broad and only taxonomical; they do not describe key attributes of the technologies detailed in the patents.

E. Formal Modeling of Cybersecurity Domain Knowledge

There have been many efforts to model the cybersecurity domain formally and create knowledge representations for exchange and shared understanding. In 2007, Herzog, Shamehri, and Duma created a detailed model of information security, including threat, asset, and countermeasure classes and relationships among them [20]. Fenz, Pruckner, and Manutscheri built on this by creating guidelines for mapping information security knowledge from one standard to a more formal security model [21]. Wang and Guo created a formal knowledge model to capture key relationships between vulnerabilities, products, countermeasures, and actors—incorporating CVE, the Common Vulnerability Scoring System, CWE, CPE, and CAPEC—to describe patterns of threats and vulnerability management to make inferences and assist users in decision making [22].

In 2012, MITRE reported on a trade study performed with the goal of creating a general cyber knowledge model and method for extending its core over a series of iterative improvements [23]. This team focused their initial work on building a model from the MAEC language and prior malware conceptual models of Swimmer, which were developed for the purpose of data exchange between security software products. From this, Obrst et al provide a vision for a much broader cyber knowledge architecture and provide a useful survey of cyber-related knowledge representations and standards to contextualize their work and the architecture.

Oltramari et al. created a basic architecture with the key elements and interactions to support situational awareness in the cybersecurity domain and then provided a modeling example using their approach [24]. Salem et al. created the TAPIO tool, which extracts operational data and integrates data from numerous sources into a knowledge graph and enables real-time exploration of the causes and effects of cyber events [25].

Syed et al. created the Unified Cybersecurity Ontology (UCO) model by integrating several existing knowledge schemas and standards into a common model for the cybersecurity domain; like Oltramari et al, their model supports cyber situational-awareness scenarios [26]. In 2019, the Cyber-investigation Analysis Standard Expression (CASE) community was formed to coordinate a community-developed specification language to "serve the broadest range of interests in cyber-investigation domains including: digital forensic science, incident response, counter-terrorism, criminal justice, forensic intelligence, and situational awareness." CASE aligns with and extends the UCO.

F. Successes Using Knowledge Graphs and Linked Data

Recent advances in knowledge modeling and linked data technology have enabled its adoption in several domains. The technology has proven fundamental to the medical, biological sciences, and bioinformatics communities in cataloging and deeply understanding complex patterns in medical datasets and genomes. Scientists can now rapidly share knowledge and collaborate on complex chemical and protein interactions. The Unified Medical Language System [27] has integrated 213 medical vocabularies [28]. Bioinformatics researchers can leverage and tie together knowledge contained in The European Molecular Biology Laboratory and European Bioinformatics Institute's open data resources and The Universal Protein Resource, which contains protein sequence and annotation data.

The Schema.org vocabulary [29] is an open data community founded by Google, Microsoft, Yahoo, and Yandex. The schema has been developed through a community process. Schema.org builds on semantic web concepts and facilitates data exchanges and a shared understanding between sites choosing to share structured social data and aggregators of that social data. Google uses schema.org types for its Knowledge Graph Search API [30]. Thomson Reuters is investing in this technology as well, creating and spinning off a knowledge graph group for financial data [31] and news content.

Immediately relevant to the D3FEND knowledge graph and methodology are pre-existing open linked data resources. We see opportunities to rapidly integrate pertinent open linked data readily available, including patent data (and patent coding systems), authors, researchers, inventors, and organizations. We believe D3FEND may be enhanced by using and referencing these open linked datasets with standards-based linked data technologies.

III. Methodology

A. Frameworks, Models, and Knowledge Graphs

There are three key information organization approaches we use to create D3FEND. A conceptual framework which "explains, either graphically or in narrative form, the main things to be studied—the key factors, constructs, or variables—and the presumed relationships among them" [32]. A domain knowledge model which "is used to reduce conceptual and terminological confusion" and foster communication, reusability, and cooperation [33]. Finally, a knowledge graph which provides a flexible representation of knowledge and enables complex machine reasoning about the domain.

To support the recurring need for effective and clear countermeasure capability specifications, we aspire to (1) provide a conceptual framework that incorporates a domain knowledge model of the cyber countermeasure domain, (2) populate the framework and model to complete a knowledge graph, and 3) relate countermeasures to the ATT&CK framework's offensive counterparts and to the larger domain space of structured cyber knowledge.

D3FEND models core countermeasure functionality and the knowledge relationships necessary to effectively understand and contextualize that functionality. Countermeasure technologies perform many functions with their software or hardware components. Some functions directly counter adversary behavior. Others are more administrative in nature and supportive

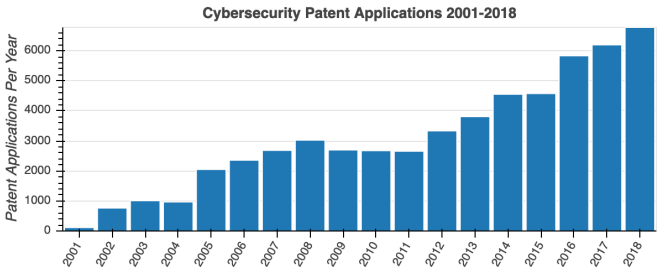

Fig. 1. Cybersecurity Patent Applications 2001-2018

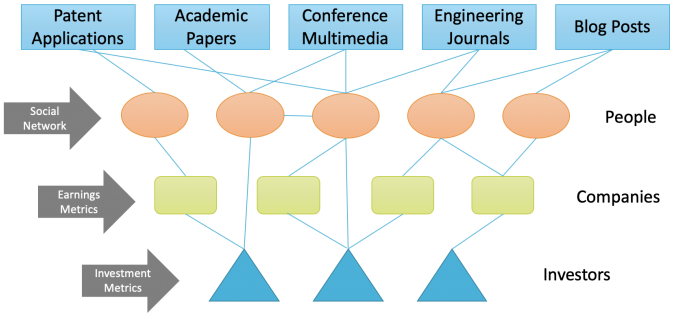

Fig. 2. Example Intellectual Property Development Network

of the core countermeasure functionality and are not the focus of D3FEND.

D3FEND is primarily concerned with abstract and generic semantics versus vendor specific terminology or technical jargon. However, analysis of technical jargon is usually required to select effective semantics—and in some cases create or define useful new semantics. For example, for D3FEND we first included an abstract definition of a *Kernel* instead of the Microsoft terminology: "Windows Kernel" or "Windows NT Kernel", thereby enabling our *Kernel* concept to apply to the "Linux Kernel" as well. When needed, D3FEND users will be able to extend this with more specific definitions of kernel types by subclassing specific OS kernels and also by compositions of their parts (e.g, "Kernel Module" elements).

B. Data Sources

The data used to build D3FEND underpins our new approach to systematically understanding cybersecurity countermeasures. The research team developed the model directly from the research claims in the literature in a bottom-up fashion, linking each countermeasure through specific citations to the literature and integrating those into higher-level abstractions. We discuss patents, existing knowledge bases, and other data sources.

1) Patents: Inventors file thousands of patent applications each year for technologies and methods for defensive cybersecurity techniques. We downloaded all U.S. Patent Office filings from 2001 to January 2019. A key phrase search against this corpus[1] shows an ever-increasing publishing rate on cybersecurity patents in Figure 1.

The patent corpus was our initial focus for multiple reasons. There is strong motivation for inventors, investors, and organizations to describe and distinguish how their cybersecurity technologies work in patents. This due to the various protections patents provide for intellectual property owners. It is also a highly curated corpus with category codes, citations, and an official legally-authoritative assessment of the novelty of their claims.In our experience, vendor white papers and marketing material do not sufficiently explain how the technologies work at an engineering level, nor do they do so in as uniform a manner as patents. To date, there appears to be no comprehensive public analysis of the cybersecurity patent corpus for the purpose of developing a knowledge graph of cyber countermeasures.

This corpus, while useful, has numerous issues that need to be understood when using it for our purpose. In some cases, the corpus is adversarial. For example, in academic papers, the citations tend to have high fidelity because researchers are incentivized to accurately represent prior scientific knowledge. Patents also have citations and prior art enumerations. However, these are often selected to bolster the case that the new patent is truly novel, useful, and non-obvious for business purposes. This is done without the peer review process used in academia.

Forty percent of U.S. patented inventions are not used. About half of these are patents meant to block competitors or to be used as bargaining chips in inter-firm negotiations [36]. Our study catalogs all cyber-defense approaches, whether yet practiced or not; use of the D3FEND knowledge graph may identify an unpracticed approach and encourage innovation and productization in that category. Furthermore, because our survey focuses on recent inventions in an emergent field, the "not used" distinction is ephemeral in many instances.

2) Existing Knowledge Bases: We analyzed the MITRE Cyber Analytic Repository [34] and mapped its analytics to the alpha version of D3FEND. The repository primarily contains detection analytics that consume endpoint telemetry. We also analyzed the ATT&CK knowledge base and developed a way to relate it to D3FEND, as discussed in Section IV-E.

3) Other Data Sources: We have also analyzed other data sources. Some of these sources include academic papers, technical specifications, and publicly available product technical documentation.

4) Key Findings: After reviewing these data sources, we determined that these available intellectual property documents could serve as the foundation for a cybersecurity countermeasure knowledge graph. We also were hopeful the resulting knowledge graph would be coherent and useful to cybersecurity architects. The D3FEND knowledge graph was built from the patent corpus primarily due to its scope, specificity, and

[1]Apache Solr search terms: "information assurance" "cyber security" "cybersecurity" "infosec" "information security" "network security" "computer security" "computer network defense" "network defense" "malware" "computer hacking" "computer virus" "data exfiltration" "cyber warfare" "information warfare" "intrusion detection" "intrusion prevention" "indicators of compromise" "security information events" "cryptographic" "cryptography"

availability. These data sources are published or made public in various formats and venues. The cybersecurity community has a diverse group of participants ranging from technicians to academics. Intellectual property is developed by the whole spectrum of participants. Example data sources are depicted in Figure 2. The figure also illustrates the intellectual property development network which produces cybersecurity countermeasure technology. We also determined these datasets are too large to analyze entirely manually. However, we started with a manual analysis process so that we can better develop automated means.

C. Countermeasure Analysis Process

Our initial approach included some preliminary efforts to use natural language processing techniques to organize, summarize, and classify the technologies claimed in the documents, primarily U.S. patent applications. We experimented with unsupervised topic modeling and text summarization algorithms. We determined these preliminary methods were not sufficient to create a semantic representation that would be useful to cybersecurity practitioners at the outset. Supervised and semi-supervised machine-learning approaches were not tried, as we did not yet have a countermeasure or artifact classification system or a set of fixed terms and thus were unable to provide labeled training data.

Our early efforts to use existing high-level models and work top down to specific countermeasures, or to propose our own high-level models absent the enumeration of countermeasure instances, did not prove to be an effective approach to create the desired framework of concepts and vocabulary. Our experience is that without anchoring the framework to specific instances from the outset, establishing team consensus for the vocabulary was difficult and was considered too subjective and too biased toward individual contributor's experience and background. Given this experience, we focused on describing specific technologies and built up a hierarchy of semantic abstractions which are directly linked to the original references to enhance the fidelity of the knowledge graph.

With a mindset towards automation, we began to manually analyze, summarize, and formulate semantics that describe the defensive techniques contained in the intellectual property documents. We then recorded the analyses in our database, creating a new labeled dataset. This resulted in a database with a vocabulary of countermeasure techniques, and references to source documents where the concepts are described. This process relied on our subject matter expertise and was labor intensive but necessary to develop an initial semantic model of the countermeasure space. In addition, we plan to use these analyses to research training algorithms to both refine the initial model and expedite the development and recognition of new countermeasure techniques.

The team reviewed over 500 cybersecurity patents selected based on multiple criteria, and analyzed those with substantial technical detail. We initially focused on "detection" oriented vendors because the team had familiarity with the domain. We chose vendors from IDC's Worldwide Cybersecurity Products Taxonomy, 2019 [19] and analyzed their patents. Some of these technologies do more than just detect unauthorized activity. We incorporated these additional techniques and categorized them in the D3FEND knowledge graph.

We approached the problem with a focus on countermeasure activity. We drew on two related systems engineering approaches to describing and understanding system activity: activity models [38] [39] and uses cases [40]. These approaches helped us frame two related sets of questions to capture the key aspects of the intellectual property when examining it in terms of being a defensive technique. The first set aimed to capture the core functional activity, and a second set captured key user interaction aspects. Both sets are shown in Table I. Given the volume of cybersecurity data and potential demands for users to stay in or on the decision loops (i.e., monitor and handle system alerts), we placed a particular emphasis on scaling concerns, both in implementing the basic capability and consequent user interactions.

TABLE I
TECHNIQUE ELICITATION QUESTIONS

How does the technology work?	**Activity model element**
What are the data inputs?	Activity Input
What are the data outputs?	Activity Output
When does the analysis occur?	Activity Control
What are the analytical algorithms?	Activity Mechanism
How does it work at scale?	Activity Mechanism
How can it be circumvented?	Activity Mechanism
How do humans interact with the technology?	**Activity model element**
What must the human user do?	User Responsibilities
How does this interaction scale?	User-System Relationship

The team quickly realized that this was too time consuming, and in many cases it was not possible to answer every question for each intellectual property document. Additionally, many technologies solved multiple problems, i.e., they contained multiple D3FEND techniques. We then improved our process with a new approach.

We noticed that data input types to the technologies were a key factor in understanding how the technology works and anchoring them to defensive techniques. Prior work by MITRE focused analytic development around an object enumeration, though the scope of the enumeration was focused on process objects rather than the entire countermeasure space [18]. This led us to create the D3FEND Digital Artifact Ontology to define these data input types with a higher degree of specificity. This concept is further discussed in section IV-E.

The knowledge and facts extracted during these analyses are recorded in the D3FEND knowledge graph; where possible we answered some of these initial questions for particular technologies or D3FEND techniques. The current knowledge graph is alpha level; we are adding features and information necessary to be useful to a public audience. Our road-map section explains our plan for developing a feature–complete beta release.

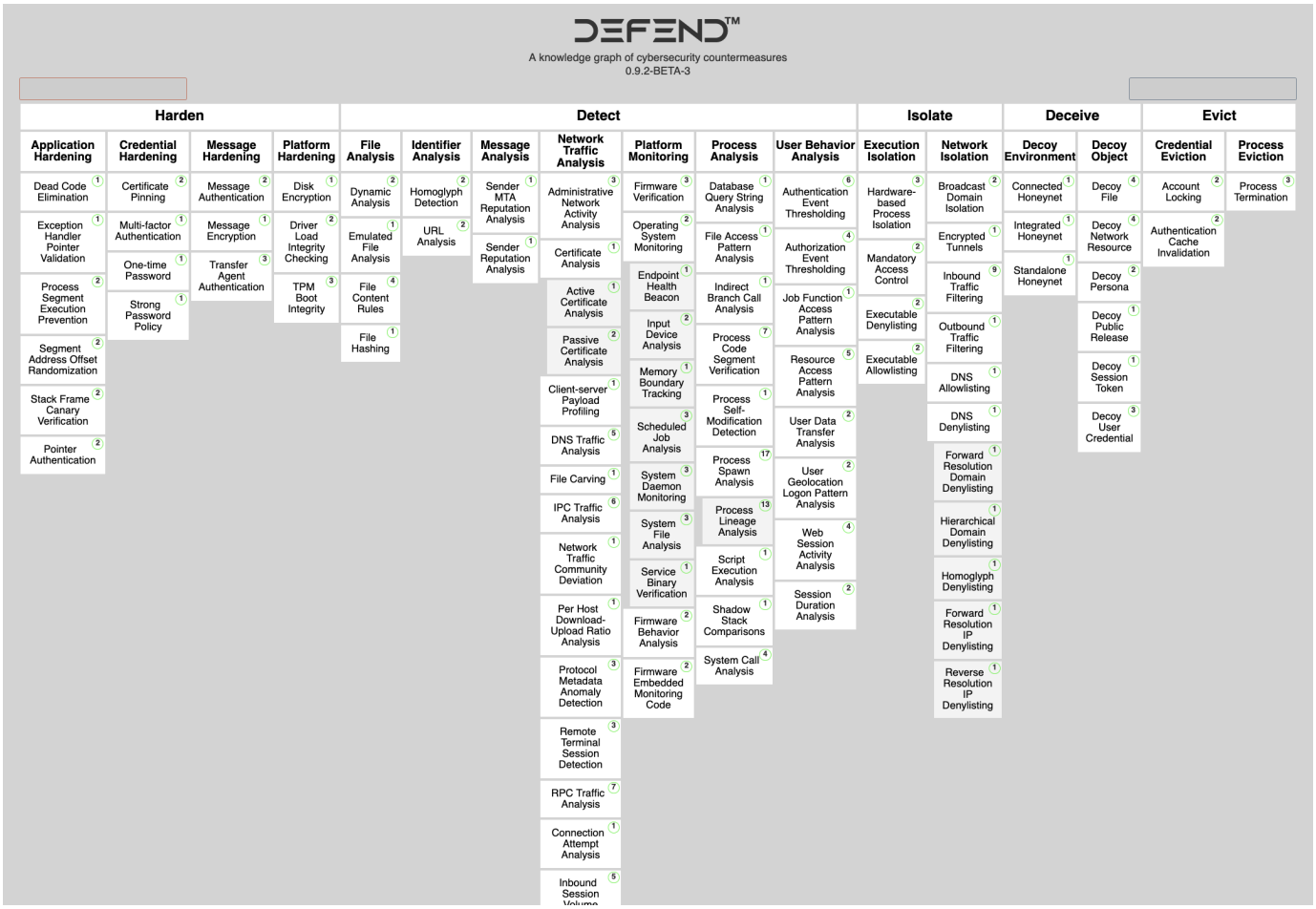

Fig. 3. D3FEND Knowledge Graph User Interface: Tactics and Techniques Overview

IV. The D3FEND Model

With our methodology we analyzed how defensive cybersecurity technologies work. Semantic patterns and structure began to emerge once a critical mass of technologies had been analyzed. We then organized and refined this structure as our knowledge increased. "D3FEND" refers to all of D3FEND's components: the knowledge graph, knowledge graph user interface, and the knowledge model.

A. Structural Overview

The D3FEND knowledge graph user interface, Figure 3, renders defensive tactics and techniques in a tabular view that also accounts for hierarchy. This view of the model is represented as a directed, acyclic graph. Each element links to more detailed information. The D3FEND knowledge model has a few key top-level concepts, shown in Figure 4. The hierarchy of classes is shown as gold arrows, while fundamental relationships between these core concepts are depicted as blue lines. This core is used to arrange the instances of concepts and organize the relational assertions that make up the D3FEND knowledge graph.

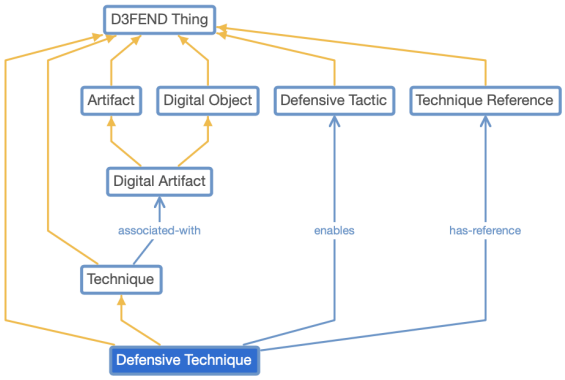

Fig. 4. D3FEND Core Knowledge Model

The D3FEND knowledge graph, currently being developed, is a particular type of knowledge base. It connects the conceptual model (i.e. knowledge model) to the particular facts. It is a graph structure that represents instances, their relationships, and their types.

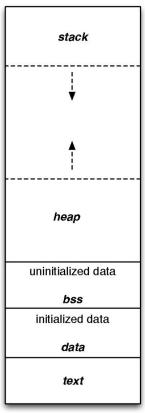

Fig. 5. Code (text) Segments and Process Memory Layout [37]

B. D3FEND Technique Semantics

A D3FEND technique is the central and most important concept in the D3FEND knowledge model. They are curated by D3FEND technique researchers. Cybersecurity technologies can be complex and may comprise multiple D3FEND techniques. Semantically, a D3FEND technique is represented as a concise phrase that captures significant information. This is difficult to do because it is a high-dimensional optimization problem.

Terms are selected for maximum information and minimal confusion. Cybersecurity is a multi-disciplinary field; this requires the technique researcher to be competent in not only the primary cybersecurity domain but also adjacent domains such as computer science and architecture, data analytics, social sciences, and information technology architecture. The D3FEND technique researcher must anticipate the potentially numerous interpretations people of different technical backgrounds may have, to select optimal, accurate, and precise terms. In general, the semantic context of computer science and computer engineering domains takes precedence over other domains.

C. Example Technique: Process Code Segment Verification

To better explain the core concepts used in D3FEND, we will discuss an example D3FEND technique—*process code segment verification*—that was discovered through our development process. Here we will explain why those words were chosen, what they mean, and what they and the resulting knowledge graph entries enable the user to do.

Using our methodology, we identified and analyzed hundreds of pieces of publicly disclosed intellectual property from various cybersecurity vendors to extract D3FEND techniques. We began to group and describe these techniques semantically.

In the case of *process code segment verification*, we determined several vendors were trying to solve the same technical problem in very different ways. These technologies were developed over two decades and deployed in radically different form-factors ranging from endpoint software agents to software compiler technologies. Additionally, their analytical approaches to verification were different. It was apparent there may be advantages and disadvantages to the different approaches. Despite the significant differences, these technologies were designed to solve the same problem: verify that the code segments (also known as text segments—Figure 5) within a running process were as expected.

To accurately make claims that a product implements a given technique, we must understand the technique's semantics. We define the term *process code segment* to mean the portions of memory assigned to a running process that contain machine code for execution. These code segments are usually loaded from disk when the application is executed to launch the process. By specifying *process code segment* we explain that this technique is not concerned with the application image on disk but its state once it is loaded into the memory of a launched process. *Verification* suggests that we are not only concerned with the integrity of the code segment but that a source of truth exists to verify against. This technique name is rooted in the computer science domain; the term *code segment* is used in many computer science textbooks to describe the machine code portions of an executable file. Additional research is required to objectively quantify the quality of technique names.

Now that we have vocabulary for understanding a specific D3FEND technique, we can ask the vendor key questions: Under what circumstances does your technology read and *verify* the *process code segments*? What is the source of truth used in the verification of the code segments; is it on the victim machine or a remote system? What happens if the code segments are determined to be invalid? The answers to these questions were interesting and illustrated the creativity and cleverness of the cybersecurity technology developers.

This knowledge, intellectual property references, and analysis were recorded in our D3FEND knowledge graph under the technique Process Code Segment Verification. With this information organized and the intellectual property catalogued, practitioners are now able not only to understand how the technique works but also to consider which approach is more suitable to their unique requirements.

Our example—*process code segment verification*—illustrates positioning within two key dimensions of semantic specificity, Figure 6. The level of specificity required for describing functionality is dependent on usage context. Our taxonomical approach accommodates various levels of specificity. This enables the model to be easily tailored to a particular use case. For example, an acquisition analyst may require more generalization, while an engineer may require more specificity.

D. D3FEND Tactics & Techniques

As we developed techniques in the D3FEND knowledge graph, we identified sets of similar techniques that had common relationships. For example, some techniques primarily analyze raw network traffic, while others focus purely on process analysis. These techniques then naturally grouped together further to organize more general types of techniques.

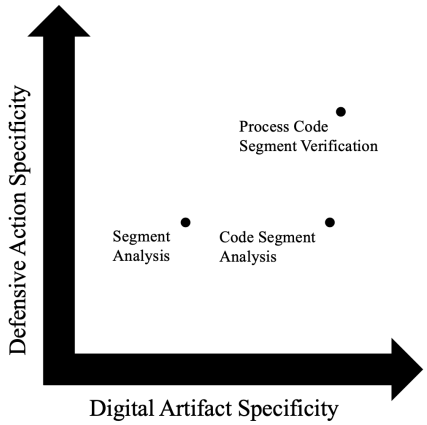

Fig. 6. Key Dimensions of Semantic Specificity

We distinguish the top-level techniques as *base techniques*, from which all the other techniques are derived. For example, the technique *process code segment verification* falls under the base technique "Process Analysis."

As we grouped the techniques, we identified another higher-level concept, the defensive tactic. A defensive tactic is the most general organizing class in the D3FEND knowledge graph; see Figure 4. A defensive tactic is a maneuver in response to some adversary action. These are action-oriented and carefully selected terms to generalize multiple techniques. Example defensive tactics we identified are *Detect, Harden, Deceive, Evict,* and *Isolate*.

The tactics are represented in the top row, and the base techniques are represented in the second row of Figure 3. More specific defensive techniques appear in the columns below the base techniques. Techniques belong to only one base technique; in general, techniques form a hierarchy from the most general to the most specific. For clarity, only two levels of the defensive technique hierarchy are depicted in Figure 3. Finally, the circled number in the individual techniques represents the number of source documents analyzed to develop the technique.

An implicit notion of state is expressed in terms chosen for tactics. A defender cannot *Evict* an adversary if he cannot *Detect* the adversary, and he cannot *Detect* the adversary if they are not there. Ideally, the defender would *Harden* his environment before the adversary penetrates it.

Tactics are the maneuvers defenders take against an adversary—"the what" of an action. The techniques are the methods used to employ those actions—"the how" of implementing the tactic. We say that these tactics are *enabled* by the techniques.

E. Digital Objects, Digital Artifacts, and Technique Mapping

A key construct in D3FEND is the *Digital Artifact Ontology (DAO)*. This ontology specifies the concepts necessary to classify and represent the *digital objects* of interest for cybersecurity analysis. In the D3FEND knowledge model, a

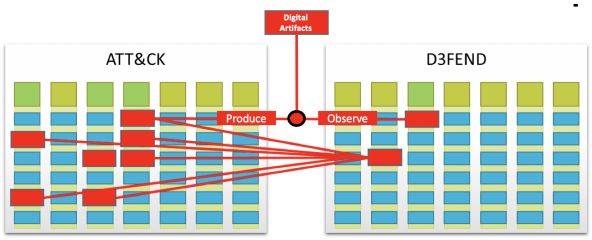

Fig. 7. Offensive and Defensive Techniques Mapping Via Digital Artifacts

digital object becomes a *digital artifact* when a cyber actor, either defensive or offensive, interacts with the object in any way. To ensure a reasonable modeling scope, the D3FEND knowledge model is only concerned with capturing knowledge about digital artifacts relevant to known cyber actors and known technologies—not all possible digital objects or their representations.

Related work in the field developed vocabularies to list and define common concepts used in cyber defense operations. Some limitations of these vocabularies motivated the construction of the D3FEND DAO, as these vocabularies were often syntactic rather than semantic, included vendor specific concepts, and were enumerative versus taxonomical. Furthermore, these vocabularies did not specify the relationships between concepts. Finally, their purview was security operations and incident response versus capability engineering. Therefore, it was necessary to develop the D3FEND DAO as more abstract semantic constructs, to unify the representation and enable vendor-agnostic and inferential reasoning.

The term artifact was chosen in the archaeological sense (e.g., *some human* put some artifact *somewhere*). If it does not exist in some digital form, on some computer, it is out of scope of the ontology. A digital artifact need not be observable or accessible, but it must possibly exist. An artifact may also comprise other artifacts, thus enabling the representation of compound artifacts; see Figure 8.

Digital artifacts also establish the conceptual scope of the D3FEND knowledge model. For example, a strong password policy is in scope because it directly affects an organization's technology configuration baseline, therefore it involves digital artifacts. As a counter-example, many organizations invest in employee cybersecurity awareness training programs. Training programs do not directly interact with digital artifacts, therefore they are not in scope.

When an attacker types on his keyboard and performs open-source internet research, he produces digital artifacts. When he develops software exploits, sends malicious phishing links, or operates a remotely controlled host in his target's environment, he creates digital artifacts on both his system, intermediate systems, and target systems. Whether the defensive actors can observe the attacker's digital artifacts depends on their vantage point and capabilities. Figure 7 illustrates these interactions between offensive and defensive techniques in a simplified way.

The cybersecurity analyst needs to know how cyber offense

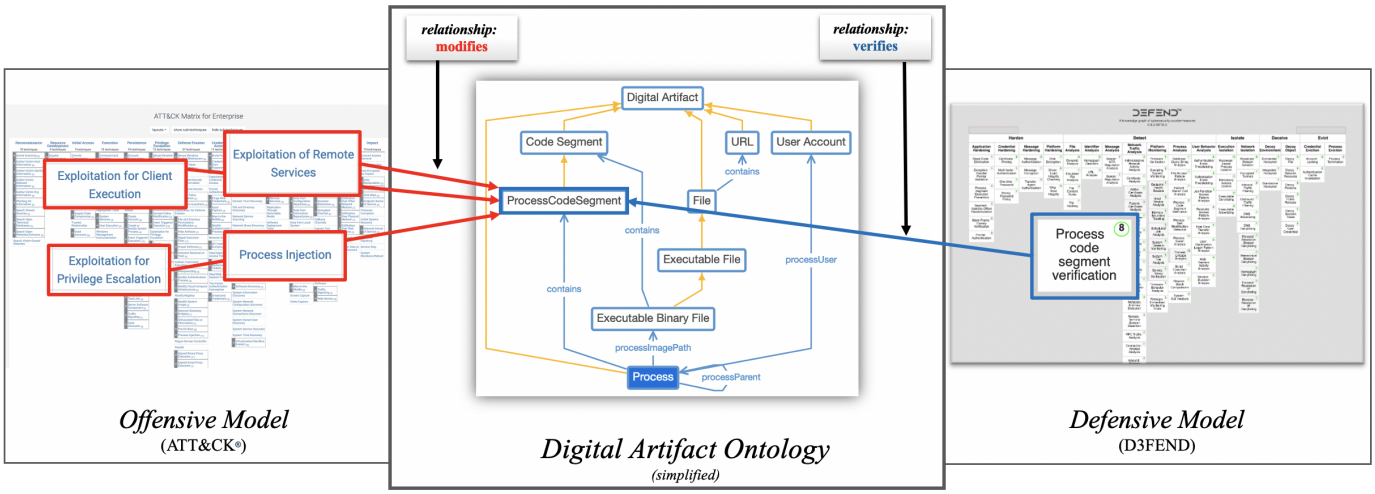

Fig. 8. Mapping via Inference Through the Digital Artifact Ontology

is covered by cyber defense and vice versa. Therefore, we need a reasonable mechanism to specify in detail the associations between these two. Our approach focuses on using the digital artifacts as the basis for conceptualizing and instantiating the relations. Both offensive and defensive techniques are associated with digital artifacts, where *associated with* is a generic relationship type for more specific relationships of *produces, executes, analyzes, accesses,* and *installs*. The D3FEND knowledge model supports more specific relationship types between offensive and defensive techniques as well, for example, *observes, detects*, and *counters*. The key benefit of this layered approach is that we can reason about the relationships between offensive and defensive techniques by analyzing how each technique is related to digital artifacts. This allows us to infer these specific types of relationships without needing to manually or directly relate offensive techniques to defensive techniques (Figure 8). Provided we accurately represent knowledge about offensive techniques, defensive techniques, and digital artifacts—each in isolation—we can accrete knowledge and derive additional knowledge and insights through inference that otherwise would require explicit enumeration.

Artifacts are defined in hierarchical specificity via classification, and their subsumption is represented by gold lines in Figure 8. For example, a Linux service daemon's configuration file is a service configuration file, which is a *System Configuration File*, which is a *Configuration File*, which is a *File—File* being the highest-order and least specific concept. This allows us to ascertain the artifacts with which a particular defensive technique is associated, with as much specificity as possible, while still maintaining the ability to ask general questions like, "which defensive techniques are associated with files?" or more specific questions like "which defensive techniques are associated with daemon configuration files?"

V. Road Map

There are three roadmap focus areas for the D3FEND project: improve and demonstrate the model utility for practitioners, deepen and widen the model via analytics, and update the model as the industry changes.

A. Model Utility

Semantic models are most useful when they are adopted; they create a common language. We will use the model in real-world use cases with cybersecurity architects and engineers. Success of the model is practitioner adoption and a positive consensus on its usefulness across multiple use cases. Garnering adoption will be an iterative process where we employ the model, qualitatively assess its usefulness, and make improvements.

One assessment will use the model to differentiate and analyze cybersecurity products. Large organizations receive many inbound requests from technology vendors. Our hope is that D3FEND clarifies the specific functionality a product offers and reduces the amount of time spent analyzing vendor marketing material. Another assessment will analyze a computer network's existing countermeasures and identify which defensive techniques are present. We will start by asking a cybersecurity architect what technologies they have related to the base techniques, as a breadth-first analysis. We hope this will identify any functionality gaps or overlap.

As we work through the use cases, we will track how much of the model was used and how many new techniques were added. Our hypothesis is that semantic reuse will increase over time, eventually reducing the frequency of additions to the model. We expect to follow-up this preliminary paper shortly with one that provides greater detail on the application of D3FEND to one or more specific capability analysis scenarios.

B. Model Depth, Breadth, and Technology Development

We must analyze more intellectual property to grow the knowledge graph. Due to the size of the ever-growing dataset,

we must automate portions of the analysis process. This requires a system of tools that can assist in extracting knowledge, finding new topics, and tracking old ones. We will explore using the system to enable machine-learning algorithms to process data in the knowledge graph and develop models to characterize the knowledge graph, and potentially new instances of knowledge. By automatically tagging new data we can expedite the discovery of new concepts and terminology in the countermeasures space. Underpinning this, we will develop a flexible knowledge graph architecture that ingests public data, captures new knowledge, and supports sophisticated storage and retrieval mechanisms.

1) Future Data Sources and the Intellectual Property Generation Network: Intellectual property can be collected from many sources. We plan to estimate the potential relevance or value of the intellectual property by creating a graph with vertices of actors (people, companies, investors), edges of relationships between them as seen in Figure 2, and weighting the edges with business or influence metrics. We anticipate applying the same methodology found in Section III that we used for analyzing patents to additional technical corpora, including academic papers, conference presentations, blog posts, and engineering journals. These data sources will be reduced to text-based representations, to enable accelerated analysis with natural language processing techniques.

C. Updating the Model

As illustrated by our analysis of the patent application submission rates, there is significant dynamism and activity in the cybersecurity market. As new sources of public intellectual property are collected, they must be archived, processed, and analyzed. Our knowledge graph tools will discover new digital artifacts and new defensive techniques. These discoveries will be added to the D3FEND knowledge graph.

VI. Conclusion

In D3FEND, we have created a precise semantic model of cybersecurity countermeasures that enables practitioners, for the first time, to assess their defenses and fill gaps with engineering-level knowledge of technical capabilities. The alpha version of D3FEND (Figure 3) received positive feedback from MITRE cybersecurity experts. Successful use in countermeasure requirements analysis is required to further validate it. We were satisfied with the semantic consistency of it because it was easy to add new defensive techniques in a repeatable fashion. These initial results show good promise, and we believe our research has demonstrated the feasibility of a countermeasure model built from real-world data sources. The next phase of work will focus on fully populating it with countermeasures and developing the tools necessary to ensure it stays up to date. Automation is required to deal with the rate of change in the domain and sustain the D3FEND countermeasure knowledge graph over time.

Acknowledgment

Thanks to fellow MITRE employees Greg Dunn, Chris Thorpe, Jay Vora, Maura Tennor, Joe Patrick, Drew Cannon, Chris Magrin, Matt Venhaus, Emily Hopkins, Zach Asher, Parker Garrison, Robert Heinemann, Dan Ellis, Steve Luke, and Bonnie Martin for their contributions to this work.

References

[1] Crunchbase Inc., "Companies — Crunchbase" 29 October 2019.
[2] The MITRE Corporation, "Common Vulnerabilities and Exposures," 1999.
[3] The MITRE Corporation, "Common Weakness Enumeration," 1999.
[4] The MITRE Corporation, "Open Vulnerability and Assessment Language," 2002.
[5] The MITRE Corporation, "Common Platform Enumeration," 2007.
[6] The MITRE Corporation, "Common Event Expression," 2007.
[7] The MITRE Corporation, "Common Attack Pattern Enumeration and Classification," 2007.
[8] The MITRE Corporation, "Malware Attribute Enumeration and Characterization," 2011.
[9] The MITRE Corporation, "Cyber Observable eXpression (CybOX™) Archive Website," 2012.
[10] NIST, "Framework for Improving Critical Infrastructure Cybersecurity," April 2018.
[11] NIST, "NIST Special Publication 800–53 Revision 4: Security and Privacy Controls for Federal Information Systems and Organizations," NIST, 2015.
[12] Department of Homeland Security, "DoDCAR/.govCAR," 2018.
[13] J. Richberg, "A Common Cyber Threat Framework: A Foundation for Communication," 2018.
[14] NSA Cybersecurity Operations, The Cybersecurity Products and Sharing Division, "NSA/CSS Technical Cyber Threat Framework v2," NSA/CSS, Fort Meade, MD, 2019.
[15] B. E. Strom, J. A. Battaglia, M. S. Kemmerer, W. Kupersanin, D. P. Miller, C. Wampler, S. M. Whitley, R. D. Wolf, "Finding Cyber Threats with ATT&CK™-Based Analytics," 2017.
[16] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "MITRE ATT&CK™ Design and Philosophy," 2018.
[17] The MITRE Corporation, "MITRE ATT&CK™," 30 October 2019.
[18] The MITRE Corporation, "CAR Data Model," 30 October 2019.
[19] International Data Corporation, "IDC's Worldwide Cybersecurity Products Taxonomy, 2019," 2019.
[20] A. Herzog, N. Shamehri, and C. Duma, "An Ontology of Information Security," in *International Journal of Information Security and Privacy*, 2007.
[21] S. Fenz, T. Pruckner, and A. Manutscheri, "Ontological Mapping of Information Security Best-Practice Guidelines," in Business Information Systems: 12th International Conference, Poznan, Poland, 2009.
[22] J. A. Wang and M. Guo, "OVM: An Ontology for Vulnerability Management," in CSIIRW, Oak Ridge, Tennessee, 2009.
[23] L. Obrst, P. Chase, and R. Markeloff, "Developing an Ontology of the Cyber Security Domain," in Semantic Technology for Intelligence, Defense, and Security (STIDS), Fairfax, Virginia, 2012.
[24] A. Oltramari, L. F. Cranor, R. Walls, and P. McDaniel, "Building an Ontology Of Cyber Security," in STIDS, Fairfax, Virginia, 2014.
[25] M. B. Salem and C. Wacek, "Enabling New Technologies for Cyber Security Defense with the ICAS Cyber Security Ontology," in STIDS, Fairfax, Virginia, 2015.
[26] Z. Syed, A. Padia, T. Finin, L. Mathews, and A. Joshi, "UCO: A Unified Cyber Ontology," in AAAI Workshop on Artificial Intelligence for Cybersecurity, Phoenix, Arizona, 2016.
[27] O. Bodenreider, "Unified Medical Language System (UMLS): integrating biomedical terminology," Nucleic Acids Res., vol. DB, no. 32, pp. 267–270, 23 May 2004.
[28] NIH, "UMLS Metathesaurus Vocabulary Documentation," NIH U.S. National Library of Medicine, 4 November 2018.
[29] Schema.org Community Group, "schema.org," Schema.org Community Group, 2015.
[30] Google, "Google Knowledge Graph Search API," Google, 2019.
[31] Refinitiv, "Knowledge Graph feed BETA," 2019.
[32] M. B. Miles and A. M. Huberman, "Qualitative Data Analysis: An Expanded Sourcebook," Thousand Oaks, CA: Sage Publications, 1994.
[33] M. Missikoff, P. Velardi, and P. Fabriani, "Text Mining Techniques to Automatically Enrich a Domain Ontology," Applied Intelligence, vol. 3, no. 18, pp. 323–350, 2003.

[34] The MITRE Corporation, “MITRE Cyber Analytics Repository”
[35] R. Bretnor, “Decisive Warfare: A Study in Military Theory,” in Decisive Warfare: A Study in Military Theory (New ed.)., Wildside Press LLC., February 1, 2001, p. 49–52.
[36] P. Zuniga, D. Guellec, et al. “OECD patent statistics manual,” Paris: OECD Publications, 2009
[37] Wikipedia contributors, “Organization of a program into segments,” Available: https://en.wikipedia.org/wiki/File:Program_memory_layout.pdf, November, 2019.
[38] D. Ross, “Structured Analysis: A Language for Communicating Ideas,” IEEE Transactions on Software Engineering 3(1), Special Issue on Requirements Analysis, January 1977, 16-34.
[39] Defense Acquisition University Press, ”System Engineering Fundamentals,” Fort Belvoir, VA, January, 2001.
[40] I. Jacobson, M. Christerson, P. Jonsson, and G. Overgaard, “Object-oriented software engineering - a use case driven approach,” Addison-Wesley 1992, ISBN 978-0-201-54435-0, pp. I-XX, 1-524.
[41] J. McCrae, “The Linked Open Data Cloud”, iod-cloud.net, https://iod-cloud.net, Accessed on: 2020-08-25.