UNIVERSITÉ
Concordia
UNIVERSITY

Critical review for

“A Knowledge Graph to Represent Software Vulnerabilities”
&
"An ontology-based cybersecurity framework for AI-enabled systems and applications"

COMP 6591 - Summer 2024

Submitted By: -
Zheyi, Zheng
Id #40266266
Habeeb Dashti
Id #40261505
Chihoon Lee
Id #40292741

Submitted To: -
Dr. Nagi Basha

1 A KNOWLEDGE GRAPH TO REPRESENT SOFTWARE VULNERABILITIES

1.1 INTRODUCTION

Regarding “A Knowledge Graph to Represent Software Vulnerabilities” by Milad Taghavi, this research introduced the significance of detecting and fixing software vulnerabilities in the evolving world of software development and maintenance. As the complexity of software systems increases, the use of various artifacts such as sequence diagrams, source codes and requirement documentation become essential. Despite all the measures being taken for the security of the software, there still exists vulnerabilities that the hackers use for unauthorized access and potential breaches. These vulnerabilities do not always necessarily occur in the main software codebase, they can come from the indirect dependencies within third party systems which makes it difficult to detect proactively.

1.Problem Statement: Frequent software vulnerabilities due to indirect dependencies and trusted third-party systems complicate vulnerability assessments, taking 5 days to detect 50% of exploits.

2. Objective: Develop a framework for a unified knowledge graph (KG) to represent and interconnect vulnerability data with standardized metadata, enhancing automated browsing and task efficiency using Resource Description Framework (RDF)/XML [1].

3. Applications: Design ontology to identify tasks, artifacts, and known vulnerabilities, enabling innovative software analytics and improved security assessments.

1.2 BACKGROUND

Semantic Web: Semantic Web is an attempt to describe and link web content in a manner that's meaningful to machines, and it extends the original web. Semantic Web wants to transform the web from a "web of documents" into a "web of data" [2]. It aims at making it easier for computers to process the huge amount of information on the web, and indeed other large databases, by enabling them not only to read but also to understand the information [3]. The research mentioned Tim Berners-Lee as the inventor of the World Wide Web, who defined the Semantic Web as an extension; also mentioned features of it such as RDF as the standard knowledge representation, enables knowledge integration between separated graphs, data retrieval and manipulation through SPARQL as RDF query language and support of semantic reasoning that let’s new knowledge be asserted from the existing facts.

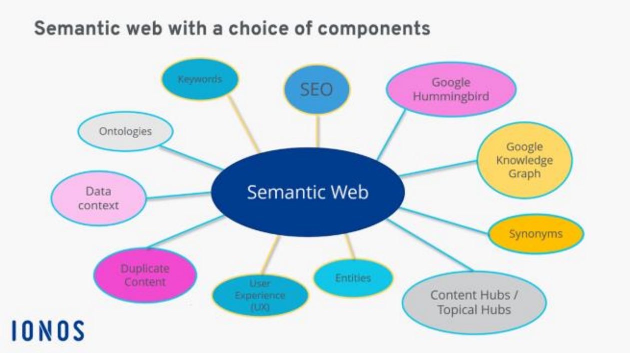
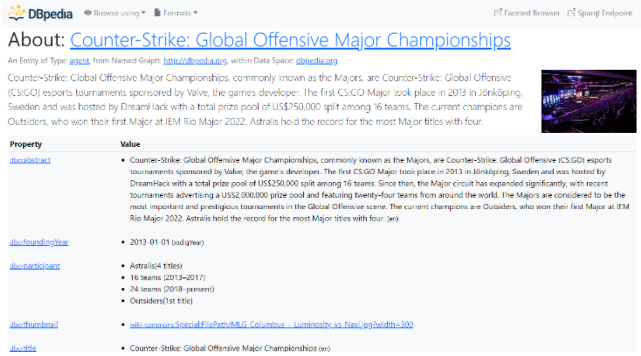


Fig. 1. Basis of the Sematic Web [4]

Knowledge Graph: The research explained Knowledge Graphs as a form of semantic networks that represent a network of real-world entities. It presented that the KG comprises of nodes, edges and labels that form a structured representation of interconnected data. For example, a node could be a client, like IBM, and an agency like Ogilvy. An edge would be categorizing the relationship as a customer relationship between IBM and Ogilvy [5]. Their role in the integration of heterogeneous knowledge resources was also displayed. A prominent example of DBpedia was cited by the researcher, which is a community-driven effort to extract structured content from Wikipedia and make it represent as a KG [6]. In Fig. 2, the DBpedia displays the data as RDF tuples for the exclusive e-sport organization.



About: Counter-Strike: Global Offensive Major Championships

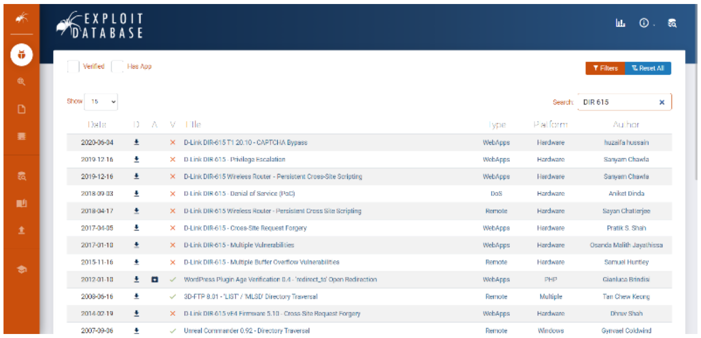
An Entity of Type: agent, from Named Graph: http://dbpedia.org, within Data Space: dbpedia.org

Counter-Strike: Global Offensive Major Championships, commonly known as the Majors, are Counter-Strike: Global Offensive (CS:GO) esports tournaments sponsored by Valve, the game's developer. The first CS:GO Major took place in 2013 in Jönköping, Sweden and was hosted by DreamHack with a total prize pool of US$250,000 split among 16 teams. The current champions are Outsiders, who won their first Major at IEM Rio Major 2022. Astralis hold the record for the most Major titles with four.

Property	Value
dbo:abstract	• Counter-Strike: Global Offensive Major Championships, commonly known as the Majors, are Counter-Strike: Global Offensive (CS:GO) esports tournaments sponsored by Valve, the game's developer. The first CS:GO Major took place in 2013 in Jönköping, Sweden and was hosted by DreamHack with a total prize pool of US$250,000 split among 16 teams. Since then, the Major circuit has expanded significantly, with recent tournaments advertising a US$2,000,000 prize pool and featuring twenty-four teams from around the world. The Majors are considered to be the most important and prestigious tournaments in the Global Offensive scene. The current champions are Outsiders, who won their first Major at IEM Rio Major 2022. Astralis hold the record for the most Major titles with four. (en)
dbo:foundingYear	• 2013-01-01 (xsd:gYear)
dbo:participant	• Astralis(4 titles) • 16 teams (2013–2017) • 24 teams (2018–present) • Outsiders(1st title)
dbo:thumbnail	• wiki-commons:Special:FilePath/MLG_Columbus_-_Luminosity_vs_Navi.jpg?width=300
dbo:title	• Counter-Strike: Global Offensive Major Championships (en)

Fig. 2. CSGO major e-sport tournament data [7]

Vulnerable Databases: The research covered Vulnerability Databases (VDB) as platforms aimed at collecting, maintaining, and distributing data about discovered security flaws in software systems [8]. National Vulnerability Database (NVD) was specifically mentioned as an exemplary VDB that offers comprehensive resources for researchers, developers, and cybersecurity, to address software vulnerabilities effectively. The research showed a unique identifier for each vulnerability and detailed information about vulnerabilities and mitigations as its features. In Fig. 3 is my personal favorite open VDB known as exploit DB that I used during my offensive cyber security practice.



□ Verified □ Has App

Show 15

Search: DIR 615

Date	D	A	V	Title	Type	Platform	Author
2020-06-04	⬇		✗	D-Link DIR-615 T1 20.10 - CAPTCHA Bypass	WebApps	Hardware	huzaifa hussain
2019-12-16	⬇		✗	D-Link DIR-615 - Privilege Escalation	WebApps	Hardware	Sanyam Chawla
2019-12-16	⬇		✗	D-Link DIR-615 Wireless Router - Persistent Cross-Site Scripting	WebApps	Hardware	Sanyam Chawla
2018-09-03	⬇		✗	D-Link DIR-615 - Denial of Service (PoC)	DoS	Hardware	Aniket Dinda
2018-04-17	⬇		✗	D-Link DIR-615 Wireless Router - Persistent Cross Site Scripting	Remote	Hardware	Sayan Chatterjee
2017-04-05	⬇		✗	D-Link DIR-615 - Cross-Site Request Forgery	WebApps	Hardware	Pratik S. Shah
2017-01-10	⬇		✗	D-Link DIR-615 - Multiple Vulnerabilities	WebApps	Hardware	Osanda Malith Jayathissa
2015-11-16	⬇		✗	D-Link DIR-615 - Multiple Buffer Overflow Vulnerabilities	Remote	Hardware	Samuel Huntley
2012-01-10	⬇	▣	✓	WordPress Plugin Age Verification 0.4 - 'redirect_to' Open Redirection	WebApps	PHP	Gianluca Brindisi
2008-06-16	⬇		✓	3D-FTP 8.01 - 'LIST' / 'MLSD' Directory Traversal	Remote	Multiple	Tan Chew Keong
2014-02-19	⬇		✗	D-Link DIR-615 vE4 Firmware 5.10 - Cross-Site Request Forgery	WebApps	Hardware	Dhruv Shah
2007-09-06	⬇		✓	Unreal Commander 0.92 - Directory Traversal	Remote	Windows	Gynvael Coldwind

Fig. 3. exploit-DB [9]

Bug Bounty: Bug bounty is a cybersecurity method that empowers organizations to minimize their threat exposure by leaning on the expertise of a community of ethical hackers. A monetary reward is given to ethical hackers for successfully discovering and reporting a vulnerability or bug to the application's developer. Bounty programs often complement regular Penetration Testing and provide a way for organizations to test their applications' security throughout their development life cycles [10]. HackerOne was cited as one of the leading cybersecurity-organizations.

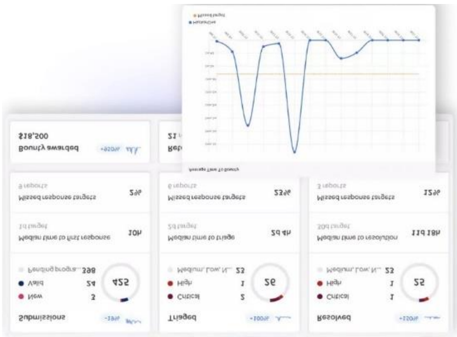


Fig. 4. HackerOne Bug Bounty [11]

1.3 SOFTWARE VULNERABILITIES ANALYSIS METHODS

Step 1 – Data Sources: In this section of the research, an awareness regarding the existence of various software artifact repositories like GitHub, Jira, Maven and exploit-DB was spread as they are potential data sources. However, the researcher guided to choose data sources that relate to software vulnerabilities and it is done so based on the sematic connections established between the available data sources. Thus, it assures an understandable and interconnected dataset.

Step 2 – Schema Extraction: It is explained that each data source is analyzed individually to understand its structure and content after relevant data sources are identified. This is the process of extraction of schemas from the interfaces.

Step 3 – Data Extraction: In this step, data extraction is different from one source to another. For this example, the researcher gave the HackerOne website to extract the data although it is not limited to any interface, it can be from Application Programming Interface (API)'s DB as far as there are any other format. In the paper, the data from the selected sources was extracted into JSON format and retrieval was facilitated through APIs or direct DB queries. It is to note that the format may not be in JSON format mandatorily, it can be in any other compatible format like XML and it acts as an intermediate representation for data processing.

Step 4 – Data Transformation to RDF: The information that is in JSON or XML format is processed and converted into RDF files. It includes data removal or converting the data for better data processing in the future. The resultant of all the steps is one-to-one mapping with the original data source in standardized representation of semantic web. In our observation, steps 2 to 4 focus on Knowledge Transformation that aims at amplifying data processing capabilities and support smooth integration into Knowledge Graphs.

Step 5 – Knowledge Graph Construction: The concepts of TBox and ABox in Knowledge Bases were introduced in this step as these terms display two different types of statements within a Knowledge Graph. The TBox statements are the "terminology component" and describe a domain of interest by defining classes and properties as a domain vocabulary. The ABox statements are the "assertion component", that are, facts associated with the TBox's conceptual model or ontologies. Together ABox and TBox statements make up a knowledge base or a knowledge graph [12]. The populating of a graph database effectively is crucial and therefore, it is made possible by the development of Knowledge Graph using these components. Until Step 4, the statements describing the ontology are available. These consist of designed schema and the statements that are instances of the data that follow the terminology. As mentioned in the research, Data integration and refinement is the source where a connection is established between ontologies and link data sources to each other, also, as this process is iterative, the design needs to be changed and updated at a continuous frequency to obtain an optimized design.

Step 6 – SPARQL Queries and Analytics: In the final step, the researcher mentioned about 'mikel/mail' bug bounty [13] that led to assigning a Common Vulnerabilities and Exposures (CVE) that is explained by NVD as its own graph. The connection junction established between the graphs obtained from step 5 is through the CVE ID is also mentioned in this step. Finally, to extract the data from the graph, SPARQL query was utilized to detect and understand software vulnerabilities in the cross-linked ontologies.

1.4 DESIGN OBJECTIVES FOR KNOWLEDGE GRAPHS

1. **Application To Software Vulnerability Domain:** It should be applied in the domain of known software vulnerabilities as it would strengthen the security of the systems.
2. **Reuse Of Existing Ontologies:** The re-usability of the ontologies should be carried out wherever and whenever applicable as it promotes consistency, compatibility and efficiency of the KG development.
3. **Support For Specific Use Cases:** It ensures that the KG meets the functional requirements and considers the challenges of the targeted domain.
4. **Utilization Of Semantic Reasoning Services:** It helps in inferring new knowledge from the existing facts.

1.5 DESIGN HIERARCHY IN KNOWLEDGE GRAPH CONSTRUCTION

System-Specific Ontologies: The research covered that these ontologies are unique to individual systems and they operate independent of each other while encapsulating knowledge resources. Examples of the unique resources that were selected into software security, such as HackerOne, NVD, Snyk, Common Weakness Enumeration (CWE), CPE and Libraries.io were given.

Domain-Specific Layer: It is mentioned that in this layer, certain ontologies are shared and used across system level resources within a particular domain. After analyzing each layer, the shared concepts are promoted to the upper layer, for example like in Object Oriented (OO) design. The researcher abstracted 2 domain ontologies such as Wiki and Bug Bounty, and modified SEVONT ontology to add additional connections to the newly created ones.

Domain-Spanning Layer: In this layer, it explained that it aggregates the knowledge that is inferred from 2 or more domains present in the lower layers. Therefore, it enables cross domain analysis and provides insights as it acts as an intermediate layer that captures diverse domain information. An activity was introduced with a potential reward for it and it is not limited to bug bounty as it could have any activity in the domain.

Integration With External Knowledge Graphs for Comprehensive Analysis: A NextCloud's profile was created in research work using DBpedia i.e., an external KG and VDB i.e., vulnerabilities from another graph, which, resulted in the created security profile containing 11 facts from DBpedia and 100 known vulnerabilities for a specific product by HackerOne general queried information.

General Layer: The research paper explained that this layer holds all the basic concepts such as resources and insights that are shared across all the domains and it holds the idea to link and re-use concepts available. In our opinion, this layer stands as a master knowledge layer. There were no modifications to this layer according to the paper and it is available only for re-usability of the existing concepts.

1.6 FOUR USE CASES IN SOFTWARE VULNERABILITY ANALYSIS

The paper showcased four distinct use cases that demonstrate the real-world applications and proficiency of the KG in detecting and analyzing vulnerabilities.

1. **Identification of Known Vulnerabilities in Projects and Dependencies:** According to the research work, the KG facilitated the detection of known software vulnerabilities inclusive of both direct and indirect dependencies. The libraries, vulnerabilities' reports, and distinct ontologies were selected to collect data about them using HackerOne and CVE. An example of the presence of 19 vulnerabilities in its dependencies and 1 known vulnerability from the researcher's mentioned 'mikel/mail' project was cited by him.
2. **Contextual Analysis of Dependency Relationships:** This part highlighted the importance of understanding dependency relationships with the help of an example where 408,000 projects use JUnit and 961 use Caffeine. Thus, the impact of Caffeine is significantly lower compared to JUnit, which clearly states that any vulnerability discovered in JUnit or Caffeine would impact the number of projects dependent on them. However, the paper pointed out that in the case of Maven, since the indirect dependencies are unknown, the researcher therefore constructed a bi-directional dependency link to assess vulnerability impacts. In the support of this use case, the researcher highlighted the bounty hunter, Rafal Janicki, who detected vulnerabilities in 6 components, that potentially affected 1893 other third-part components.

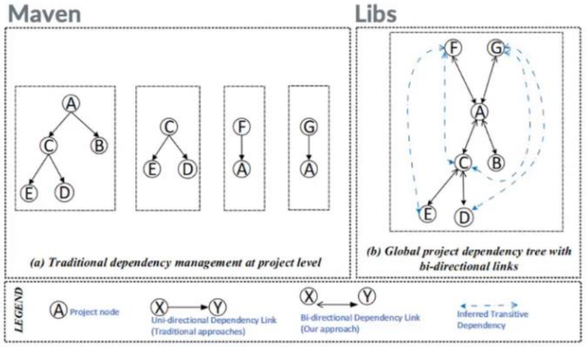

Maven
Libs
A
C
B
E
D
C
E
D
F
A
G
A
(a) Traditional dependency management at project level
F
G
A
C
B
E
D
(b) Global project dependency tree with bi-directional links
LEGEND
A Project node
X
Y
Uni-directional Dependency Link (Traditional approaches)
X
Y
Bi-directional Dependency Link (Our approach)
Inferred Transitive Dependency

Fig. 6. Bi-Directional KG [14]

3. **Detection of Specific Weakness Types:** It discussed the KG's proficiency to detect the types of weaknesses in the explored components by the expertise of a bounty hunter. The query would require HackerOne, CVE, NVD and CWE ontologies. An example of Rafal Janicki is shown in the paper where 15 path traversal, 12 Cross Site Scripting (XSS) and 1 Structured query language (SQL) injection were detected as his expertise.
4. **Integration With External Knowledge Graphs for Comprehensive Analysis:** The researcher created a NextCloud's profile using DBpedia i.e., an external KG and VDB i.e., vulnerabilities from another graph, which, resulted in the created security profile containing 11 facts from DBpedia and 100 known vulnerabilities for a specific product by HackerOne general queried information.

1.7 CONCLUSION & PAPER REVIEW

Conclusion: A robust methodology tailored for the development of a KG for resources associated with known software vulnerabilities was introduced. The modeling approach was validated using the four use cases that expressed versatility, relevance and adaptability in practical applications.

Review: Our team strongly agrees with this research paperwork as it showcased an aspiring and a promising comprehensive methodology for exploiting Knowledge Graphs to detect and analyze Software Vulnerabilities. It also implied the strength of applicability and flexibility due to its modeling approach. Lastly, our team mutually agreed that such ontology would enable us to integrate it into other technologies rather than Software Development alone because of the ability that KG(s) hold to capture any type of information.

2 AN ONTOLOGY-BASED CYBERSECURITY FRAMEWORK FOR AI-ENABLED SYSTEMS AND APPLICATIONS

2.1 INTRODUCTION

In this section, we will discuss the second paper we have selected, titled "An ontology-based cybersecurity framework for AI-enabled systems and applications" by Davy Entrepreneurs and Wouter Joosen [15]. Davy and Wouter's paper mainly focus on developing a cybersecurity framework for AI-enabled systems and applications. Their motivation is that current existing cybersecurity methods may not fully capture the unique vulnerabilities and threats posed by AI systems and they would like to create a cybersecurity framework that can systematically document AI threats and defenses. The following paragraphs will delve into their paper.

2.2 MOTIVATION

As we may all noticed, AI is getting more and more popular and has a huge influence on everyone’s daily life. However, this rise in AI usage has also highlighted the importance of addressing cybersecurity concerns related to AI-enabled systems. More specifically, AI-enabled systems are all trained by data, and hackers can aim at this process and poison the training data or manipulate inputs. Furthermore, AI is usually black boxes, which means users and creators don’t know the whole decision-making process. This issue makes it difficult to identify and understand potential issues within AI. Based on these reasons, Davy and Wouter believe a cybersecurity framework for AI-enabled systems is necessary. To build this framework, a knowledge base is necessary and Davy and Wouter picked ontology to be the foundation of their framework. An ontology can be used as a type of data model that holds a collection of concepts within a particular domain with relationships between these concepts. It can be used as a knowledge representation for knowledge bases.[16] Davy and Wouter believe ontologies can be a perfect way to represent the cybersecurity domain as ontologies provide structured and standardized representation for knowledge within a domain.[15] In addition, ontologies can semantically model, document, and reason about implicit relationships among diverse cybersecurity concepts and properties.[15]

2.3 METHODOLOGY

After deciding to use ontologies to build their cybersecurity framework, Davy and Wouter’s listed out their three-step approach for their project. Firstly, find and review existing taxonomies that fit with cybersecurity. Then transform those taxonomies along with new knowledge, relationships, attacks, and defense into semantically enhanced ontologies that can be used for future automated reasoning. Lastly, instantiate ontologies by integrating well-known machine learning threats and corresponding countermeasures.

For the first step, Davy and Wouter picked the MITRE ATLAS taxonomy as their ground. MITRE is a not for profit corporation that established to advance national security and serve the public interest as an independent adviser.[17] MITRE ATLAS is globally accessible knowledge base of adversary tactics and techniques based on real-world observations.[18] MITRE ATLAS is also continuously updating based on user feedback, which makes it always up to date. It is also widely used in cybersecurity research and education.[19] After picking the MITRE ATLAS taxonomy, Davy and Wouter retrieved MITRE ATLAS knowledge base in STIX 2.1 JSON collection format and then transformed it into a property graph by using python application they created. Till this step, the basic ontology is built and framework is able to make semantic reasoning.

2.4 LIMITATIONS

Even though MITRE ATLAS database is great foundation of such framework, it is still not enough for a cybersecurity framework for AI-enabled systems. Davy and Wouter argue there are two main limitations in MITRE ATLAS database. The first limitation is MITRE ATLAS database only record tactic and technique level information without procedure level information as the system is mainly designed for human analysts and researchers. Furthermore, information in database might contain URL, DOI or publications, which is not facilitate for machine. The second limitation is MITRE ATLAS database does not mark those mitigation even if it has been broken by new attacks. As introduce a new defense might will lead to a more sophisticated attack, and attack might will lead to a new defense, this arms race condition makes MITRE ATLAS database no guarantee of long term efficiency of existing information and hence the framework with only MITRE ATLAS won’t be able to correctly evaluate possible vulnerability.[15] Davy and Wouter’s solution to these two limitations is their step three, instantiate ontologies by integrating well-known machine learning threats and corresponding countermeasures.

To overcome limitations, Davy and Wouter first introduced a procedure level in their ontology to give a more detailed description of technique or mitigation. Then they organized related scientific literature mentioned in the MITRE ATLAS database into their ontologies and related those ontologies with existing ontology if there exists any connection. Moreover, Davy and Wouter instantiate their ontologies with another 60 scientific papers that were not included in original ontologies and these papers are all related to adversarial machine learning. As the goal of those ontologies is to provide knowledge about attacks and corresponding defense techniques, Davy and Wouter used the following steps to include those papers. First, semantically describe the ML pipeline, dataset artifacts used in the attack/defense experiments, threat model, and each related scientific publication on adversarial attacks and defenses using the SPAR ontologies. Then, link information retrieved from the last step with existing knowledge or mitigate existing knowledge that’s from MITRE ATLAS. Lastly, link how defenses have been broken with new attacks, or vice versa.[15] After instantiating ontologies by integrating well-known machine learning threats and corresponding countermeasures, Davy and Wouter believe the system is ready to be experimented.

2.5 REVIEW

After carefully reviewing this paper, we have uncovered several advantages and disadvantages that were not addressed. Firstly, MITRE ATLAS is a perfect choice of taxonomy. It provides both defense and attack side taxonomy and it encompasses a wide range of attack techniques and operating system, which makes it a great foundation for ontology. However, one of the limitations of MITRE ATLAS is also inherited by Davy and Wouter's framework, which is no guarantee of long-term efficiency of existing information. Even though they instantiated another 60 scientific papers into their system to mitigate expired techniques, this issue still exists in long team and require constant update of their ontologies. Keeping those ontologies up to date will be time consuming and needs human expertise to perform such update. The second point we would like to mention is the usage of this system. This ontology-based cybersecurity framework clearly provides a great knowledge base regarding cybersecurity, but Davy and Wouter only considered humans as a user. They believe AI system that use such knowledge base, able to process sensitive data and act without supervised by human might lead to privacy concerns.[15] We partially agree with their idea, but we still think AI that offer automated threat detection could be a wonderful research topic. The last point we noticed is Davy and Wouter did not talk about the scalability of this framework. As the AI system gets more and more complex and the framework will also expend its knowledge accordingly, then how significant will the size and complexity of the ontology impact the speed of OWL queries can be opportunity for another future research.

3 SUMMARY

In summary, both papers provided valuable contributions using knowledge base within cybersecurity domain. Paper 1 offered a promising way of using knowledge graphs to improve software vulnerability and paper 2 provided a framework for AI-enabled applications. By combining these approaches, we have the potential to create a future where AI-enabled software is not only powerful but also highly secure with constant iterative updates and automated processing.

REFERENCES

[1] Steven J. Baskauf & Campbell O. Webb. 2014. Bioimages graph model based on 2014 Darwin-SW (DSW) ontology (version 0.4). Vanderbilt University. Retrieved from: https://bioimages.vanderbilt.edu/pages/standards.htm

[2] Devopedia. 2022. "Semantic Web." Version 8, February 15. Accessed 2023-11-12. Retrieved from: https://devopedia.org/semantic-web

[3] Péter Szeredi, Gergely Lukácsy and Tamás Benkő. The Semantic Web Explained. 2014. Cambridge University. Retrieved from: https://assets.cambridge.org/97805217/00368/frontmatter/9780521700368_frontmatter.pdf

[4] Digital Guide IONOS. 2021. Semantic web. Retrieved from: https://www.ionos.ca/digitalguide/online-marketing/search-engine-marketing/semantic-web/

[5] IBM. 2024. What is a knowledge graph?. Retrieved from: https://www.ibm.com/topics/knowledge-graph

[6] DBpedia. 2024. About DBpedia. Retrieved from: https://www.dbpedia.org/about/

[7] DBpedia. 2024. Counter-Strike:_Global_Offensive_Major_Championship. Retrieved from: https://dbpedia.org/page/Counter-Strike:_Global_Offensive_Major_Championship

[8] Wikipedia.2024. Vulnerability database. Retrieved from: https://en.wikipedia.org/wiki/Vulnerability_database

[9] Exploit database. 2024. Exploit Database - Exploits for Penetration Testers, Researchers, and Ethical Hackers. Retrieved from: https://www.exploit-db.com/

[10] HackerOne. 2024. What are bug bounties and how do they work?. Retrieved from: https://www.hackerone.com/vulnerability-management/what-are-bug-bounties-how-do-they-work-examples

[11] HackerOne. 2024. Bug Bounty Program for Businesses | HackerOne. Retrieved from: https://www.hackerone.com/product/bug-bounty-platform

[12] Wikipedia. 2024. Abox - Wikipedia. Retrieved from: https://en.wikipedia.org/wiki/Abox#:~:text=TBox_statements_are_the_%22terminology,TBox's_conceptual_model_or_ontologies

[13] GitHub. 2024. mikel/mail. A Really Ruby Mail Library by Mikel Lindsaar lead Bug Bounty. Retrieved from: https://github.com/mikel/mail

[14] Milad Taghavi. 2023. A Knowledge Graph to Represent Software Vulnerabilities. Concordia University. Retrieved from: https://spectrum.library.concordia.ca/id/eprint/991794/1/Taghavi_MA_S2023.pdf

[15] Preuveneers D, Joosen W. 2024. An Ontology-Based Cybersecurity Framework for AI-Enabled Systems and Applications. Future Internet. 2024; 16(3):69. https://doi.org/10.3390/fi16030069

[16] Luyen, Le & Abel, Marie-Hélène & Gouspillou, Philippe. 2023. A Constraint-based Recommender System via RDF Knowledge Graphs. https://www.researchgate.net/publication/372487696_A_Constraint-based_Recommender_System_via_RDF_Knowledge_Graphs

[17] MITRE. (2024). Who we are. Retrieved from https://www.mitre.org/who-we-are

[18] MITRE ATT&CK® 2020. MITRE ATT&CK®: Design and Philosophy. https://attack.mitre.org/docs/ATTACK_Design_and_Philosophy_March_2020.pdf

[19] Roy, Shanto & Panaousis, Emmanouil & Noakes, Cameron & Laszka, Aron & Panda, Sakshyam & Loukas, George. 2023. SoK: The MITRE ATT&CK Framework in Research and Practice. https://www.researchgate.net/publication/370070213_SoK_The_MITRE_ATTCK_Framework_in_Research_and_Practice