future internet

MDPI

Article

An Ontology-Based Cybersecurity Framework for AI-Enabled Systems and Applications

Davy Preuveneers * and Wouter Joosen

DistriNet, KU Leuven, Celestijnenlaan 200A, B-3001 Leuven, Belgium
* Correspondence: davy.preuveneers@kuleuven.be; Tel.: +32-16-327853

Abstract: Ontologies have the potential to play an important role in the cybersecurity landscape as they are able to provide a structured and standardized way to semantically represent and organize knowledge about a domain of interest. They help in unambiguously modeling the complex relationships between various cybersecurity concepts and properties. Leveraging this knowledge, they provide a foundation for designing more intelligent and adaptive cybersecurity systems. In this work, we propose an ontology-based cybersecurity framework that extends well-known cybersecurity ontologies to specifically model and manage threats imposed on applications, systems, and services that rely on artificial intelligence (AI). More specifically, our efforts focus on documenting prevalent machine learning (ML) threats and countermeasures, including the mechanisms by which emerging attacks circumvent existing defenses as well as the arms race between them. In the ever-expanding AI threat landscape, the goal of this work is to systematically formalize a body of knowledge intended to complement existing taxonomies and threat-modeling approaches of applications empowered by AI and to facilitate their automated assessment by leveraging enhanced reasoning capabilities.

Keywords: cybersecurity; artificial intelligence; ontology; attacks; defenses

check for updates

Citation: Preuveneers, D.; Joosen, W. An Ontology-Based Cybersecurity Framework for AI-Enabled Systems and Applications. *Future Internet* **2024**, *16*, 69. https://doi.org/10.3390/fi16030069

Academic Editor: Claude Chaudet

Received: 24 January 2024
Revised: 13 February 2024
Accepted: 20 February 2024
Published: 22 February 2024

1. Introduction

In the cybersecurity domain, ontologies can play a pivotal role by enabling a structured and standardized representation and organization of important or strategic knowledge within the domain. A key benefit of ontologies is that they are able to semantically model, document, and reason upon often implicit or obscure relationships among diverse concepts and properties that characterize the continuously evolving cybersecurity landscape. By leveraging ontologies, cybersecurity professionals can establish a common language and terminology to articulate the multifaceted perspectives of security, such as emerging threats and relevant mitigations. These structured representations and terminologies not only enhance communication among cybersecurity professionals, but they also enable comprehensive knowledge management via a machine-interpretable specification with support for automated reasoning. For example, during an incident response, ontologies can help in organizing and categorizing information related to security incidents when sharing this information with other stakeholders. This information includes details about the attack vectors, compromised systems, and the tactics, techniques, and procedures (TTPs) employed by threat actors. One of the first cybersecurity ontologies in this domain that integrates well-known security standards for information sharing and exchange, such as STIX (https://oasis-open.github.io/cti-documentation/stix/intro.html (accessed on 23 January 2024)), is the Unified Cybersecurity Ontology (UCO) [1].

In this research, we focus on a specific class of attacks, particularly security and privacy threats targeted at artificial intelligence (AI) components integrated into contemporary systems, services, and applications. The integration of machine learning (ML) and deep learning (DL) increases the attack surface of existing systems beyond those of traditional cyberattacks (e.g., brute-force password attacks, SQL injection, zero-day exploits). The goal of the adversary is to compromise the ML-based decision-making process embedded within the application or to disclose sensitive information pertaining to the data upon which the ML model was trained. A widely recognized classification system within the domain of adversarial ML is MITRE ATLAS (Adversarial Threat Landscape for Artificial-Intelligence Systems) [2]. The objective of this living knowledge base of adversary tactics and techniques is to increase awareness regarding known vulnerabilities and emerging threats in AI-enabled systems. The MITRE ATLAS taxonomy draws inspiration from and complements the tactics, techniques, and procedures (TTPs) from the MITRE ATT&CK framework [3,4].

The challenge that we address focuses on the limitations of current taxonomies and threat modeling methodologies for ML-based applications. Taxonomies offer a hierarchical structure to categorize concepts and knowledge about threats, adversarial tactics, and mitigations. These models do not have the expressiveness necessary for in-depth analysis and automated reasoning about both attacks and defenses, particularly in the context of the ongoing arms race between them [5,6]. This challenge becomes more pronounced as the body of knowledge, including the academic literature, on the AI threat landscape continues to grow at an unprecedented rate. Instead, our goal is to document important concepts in ontologies. Ontologies go beyond classification by offering a semantically richer machine-interpretable specification language with support for automated reasoning. Hence, the primary objective of this work is to systematically encode a comprehensive body of knowledge from the scientific literature to augment the current taxonomies, such as MITRE ATLAS, with the state of the art. By doing so, we aim to streamline the automated assessment of the attack surface of AI-enabled applications by leveraging enhanced reasoning capabilities, and to contribute to a more robust understanding of the challenges posed by emerging AI risks and threats to the development of enhanced defense mechanisms and an overall improvement of the security posture of these applications.

The remainder of this paper is structured as follows. In Section 2, we review relevant related work on the use of ontologies in the cybersecurity domain along with efforts to establish a framework for machine learning security. Section 3 discusses our methodology on how we extend existing ontologies to document attacks and defenses for ML-based applications. We evaluate and validate our approach in Section 4 from both a qualitative and quantitative perspective. We summarize our contributions and offer suggestions for further research in Section 5.

2. Related Work

In this section, we review relevant related works on cybersecurity taxonomies and ontologies, including those that focus on AI threats, and complementary threat modeling approaches. Additionally, we discuss how approaches have been used in key application domains.

2.1. Security Taxonomies and Ontologies

Syed et al. [1] propose the Unified Cybersecurity Ontology (UCO), which aims to enhance cyber situational awareness and to facilitate information sharing by incorporating data and knowledge schemas from various cybersecurity systems and widely adopted cybersecurity standards. UCO provides comprehensive coverage and has been systematically aligned with publicly available cybersecurity ontologies. To demonstrate the added value of the UCO ontology, the authors present and validate a prototype system complemented with concrete use cases.

Onwubiko presents an ontology called CoCoa [7], an acronym for “Cybersecurity Operations Centre Ontology for Analysis Process”. This process ontology is aligned with the NIST cybersecurity framework (see https://www.nist.gov/cyberframework (accessed on 23 January 2024)) with the objective to offer cybersecurity analysts in security operations centers (SOCs) enhanced operational situational awareness of monitored assets, potential threats, and vulnerabilities, the compromise path, and the attack surface. To realize this, the proposed process ontology goes beyond mere log collection and specifically focuses on the analysis of five information sources, namely: (1) events and logs, (2) network information, (3) structured digital feeds, (4) semi and unstructured feeds, and (5) threat intelligence.

Mozzaquatro et al. [8] present an ontology-based framework tailored to Internet of Things (IoT) cybersecurity with the objective to propose adequate security services for specific threats. The framework has a two-phase approach: (1) design time—constructing security services in light of existing enterprise processes; and (2) run time—real-time monitoring of the IoT environment for threats and vulnerabilities, as well as proactive countermeasures to adapt existing services. The authors define and instantiate the IoTSec ontology and evaluate its feasibility via an ontology assessment and a case study featuring an industrial implementation.

Martins et al. [9] take a different perspective and focus on the complexity of cybersecurity in large enterprises and the interdisciplinary nature of its management. Their research offers three key contributions. Firstly, it includes a literature review of cybersecurity ontologies. Secondly, it classifies these works based on key characteristics to facilitate a systematic comparison. Lastly, the paper analyzes the results, identifies gaps, and proposes good practices in ontology engineering for cybersecurity to achieve solutions that are better aligned with organizational needs.

2.2. Threat Modeling and Assessment of ML-Based Systems

In their report, ENISA [10] provides a taxonomy for ML algorithms, as well as a detailed analysis of threats and security controls in well-known standards. Their report covers key data types used by ML algorithms, the nature of training (supervised or unsupervised), and the importance of both accuracy and explainability for users. The report then conducts a comprehensive analysis of threats that target ML systems, including data poisoning, adversarial attacks, and data exfiltration. Additionally, it evaluates contemporary security controls from widely adopted standards, such as ISO 27001 [11] and the NIST Cybersecurity Framework, and maps them to the core functionalities of ML systems they aim to protect. Their analysis reveals that traditional security controls, while highly effective for information systems, require complementary security controls tailored to ML functionalities. Based on a systematic review of the relevant academic literature, the report presents a comprehensive list of security controls specifically for ML systems, such as the inclusion of adversarial examples in training datasets to make ML models more robust.

Introduced in June 2021, MITRE ATLAS [2] serves as a living knowledge base of adversarial ML tactics, techniques, and case studies. It targets cybersecurity experts, data scientists, and organizations with the objective to raise awareness about recent advancements in attacks and defenses related to adversarial ML. The information repository is structured within the framework known as Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) for AI. This framework establishes a standardized method for delineating and classifying adversarial techniques and for recognizing and mitigating vulnerabilities in AI systems.

Tidjon et al. [12] highlight that the ML embedded in critical systems is an attractive target for adversaries employing diverse tactics, techniques, and procedures (TTPs) to compromise the confidentiality, integrity, and availability of ML systems. Their empirical study of 89 real-world ML attack scenarios, 854 ML repositories from GitHub, and the Python Packaging Advisory database aims to create a better understanding of the nature of ML threats, with the objective to identify common mitigation strategies. Their results highlight convolutional neural networks (CNNs) as frequently targeted models, identify vulnerable ML repositories, and report on the most common vulnerabilities, targeted ML phases, models, and TTPs.

Mauri et al. [13] acknowledge the same concerns as those mentioned in the previous work and argue that, as ML-based systems become more prevalent, the need for a tailored threat modeling approach for the AI–ML pipeline is crucial. The paper introduces STRIDE-AI, an asset-centered methodology for assessing the security of AI–ML systems, and potential failure modes in assets across their life-cycle. The objective of STRIDE-AI is to help ML practitioners with selecting effective security controls to safeguard ML assets, which the authors illustrate with a real-world use case from the TOREADOR H2020 project.

2.3. Bridging the Gap

The challenge that we address in this work revolves around the limited expressiveness of current taxonomies and threat modeling methodologies for AI-enabled systems and applications to effectively analyze and reason in an automated manner about both offensive and defensive strategies, as well as the ongoing arms race between them. This challenge becomes increasingly prominent as the body of knowledge on adversarial ML continues to expand at an unprecedented pace. The objective of our research is to semantically structure this body of knowledge with an ontology-based cybersecurity framework that enhances rather than replaces existing taxonomies and threat modeling methodologies for AI-empowered applications and to facilitate their automated assessment by harnessing advanced reasoning capabilities, particularly in the context of an ever-expanding AI threat landscape.

3. Methodology

Our methodology is based on a three-step approach. Firstly, we review and analyze key taxonomies that align with our requirements and that can be augmented towards our needs. In the second step, we enrich these selected taxonomies by transforming them into semantically enhanced ontologies to facilitate automated reasoning. Subsequently, we extend these ontologies by incorporating specific concepts and relationships to unambiguously document the domain of ML attacks and defenses. Finally, in the third step, we instantiate the refined ontology by integrating well-known instances of ML threats and corresponding countermeasures, as well as the arms race between them.

3.1. Analysis of the Base Taxonomies and Ontologies

The MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) framework is a knowledge base that is used to describe the actions, behaviors, and intentions of cyber adversaries. It is a comprehensive collection of tactics, techniques, and procedures (TTPs) that adversaries use in the real world to achieve their objectives across different stages of the cyber attack life cycle.

- Tactics: High-level description that encompasses a series of behaviors and actions utilized by the adversary to accomplish a particular goal.
- Techniques: More detailed guidelines and intermediate methods outlining the implementation of a tactic.
- Procedures: Low-level, step-by-step description of the activities within the context of a technique to carry out an attack tactic, including the tools or methods used by the threat actor to effectively accomplish their goals.

The MITRE ATT&CK framework is organized into threat matrices, with each matrix focusing on a specific platform, such as Enterprise, Mobile, and Industrial Control Systems (ICSs). It is continuously updated to reflect the evolving landscape of cyber threats and adversary techniques.

Our ontological framework is grounded in the MITRE ATT&CK framework, chosen not only for its widespread adoption in the industry but also for its versatility in addressing adversarial ML threats via the complementary MITRE ATLAS taxonomy. The ATT&CK framework has been extended to encompass the unique challenges posed by adversarial ML. This alignment allows us to effectively capture and analyze the distinct tactics, techniques, and procedures employed in this evolving threat landscape.

3.1.1. MITRE ATT&CK in STIX 2.1 Format

The MITRE ATT&CK dataset is accessible in the Structured Threat Information Expression (STIX) 2.1 [14] format, a language and serialization format specifically designed for exchanging cyber threat intelligence. STIX serves as a machine-readable format that facilitates access to the detailed ATT&CK knowledge base. ATT&CK employs a combination of pre-defined and customized STIX objects to implement ATT&CK concepts, as documented at https://github.com/mitre-attack/attack-stix-data/blob/master/USAGE.md (accessed on 23 January 2024). In our research, we utilized version 14.1 released on 16 November 2023, which can be found at https://github.com/mitre-attack/attack-stix-data (accessed on 23 January 2024). Processing these JSON files is fairly trivial with the Python stix2 (version 3.0.1) package (see https://github.com/oasis-open/cti-python-stix2 (accessed on 23 January 2024)), as shown in Appendix A in Listing A1, and the Python mitreattack-python (version 3.0.2) package (see https://github.com/mitre-attack/mitreattack-python (accessed on 23 January 2024)), as shown in Listing A2. Both code snippets filter the first entry in the MITRE ATT&CK (version 14.1) Enterprise knowledge base that matches a set of two constraints.

3.1.2. From STIX 2.1 JSON Collections to Semantically Enriched Ontology Representation

The MITRE ATT&CK knowledge base for the Enterprise, Mobile, and ICS domains is represented in STIX 2.1 JSON collections. This particular JSON file format is easily machine-readable and queryable but lacks the expressiveness to represent complex relationships between ATT&CK concepts as well as the capacity for automated reasoning. To address this limitation, we transform the JSON dataset into a semantic representation. A similar effort was undertaken before as part of the UCO ontology [1], which provides an ontology specification for the STIX 2.0 standard (accessible at https://github.com/Ebiquity/Unified-Cybersecurity-Ontology/blob/master/stix/stix2.0/stix2.ttl (accessed on 23 January 2024)).

In our approach, we build upon the Web Ontology Language (OWL) representation of the STIX 2.1 standard produced by the OASIS Threat Actor Context (TAC) Technical Committee, accessible at https://github.com/oasis-tcs/tac-ontology (accessed on 23 January 2024). As documented by the OASIS TAC TC (see https://github.com/oasis-tcs/tac-ontology/blob/master/docs/gh-docs/stix-spec.md (accessed on 23 January 2024)), the content of the STIX 2.1 specification is more aptly suited for a property graph representation rather than a semantic graph representation due the use of relationship nodes between STIX objects. Consequently, the committee has undertaken various efforts to represent the STIX 2.1 specification in multiple forms, allowing for a comprehensive semantic graph representation. By relying on their work, our framework is now able to semantically reason with ATT&CK concepts in a more sophisticated manner.

3.2. Ontological Extensions for ML Attacks and Defenses

In order to attain both syntactic and semantic interoperability, we first reintroduce the mapping of MITRE ATT&CK concepts onto the STIX 2.1 semantic graph representation. This involves, among other tasks, the definition and alignment of new semantic concepts and relationships. We therefore enhance the STIX Semantic Extension Ontology (see https://github.com/oasis-tcs/tac-ontology/blob/master/docs/gh-docs/stix-semex.md (accessed on 23 January 2024)) so that the MITRE ATT&CK knowledge base can be examined by a description logic reasoner such as HermiT [15] or ELK [16]. A subset of our ATT&CK Semantic Extension Ontology in Turtle format is shown in Listing 1 and illustrates how we mapped ATT&CK concepts onto the STIX semantic graph representation.

The complete specification for our MITRE ATT&CK Semantic Extension Ontology can be accessed through the following link: https://people.cs.kuleuven.be/~davy.preuveneers/ns/cti/attack-semex.rdf (accessed on 23 January 2024). Currently, it encompasses and maps 4 object properties, 22 data properties, and 14 classes. Additionally, we have developed a Python application utilizing the mitreattack-python (version 3.0.2) package to transform the complete STIX-based ATT&CK (version 14.1) Enterprise knowledge base from JSON format into a semantic representation. The Turtle format representation of technique T1111 can be found in part in Listing 2.

Listing 1. Subset of our MITRE ATT&CK Semantic Extension Ontology in Turtle format.

```
@prefix : <https://people.cs.kuleuven.be/~davy.preuveneers/ns/cti/attack-semex#> .
@prefix cti: <http://docs.oasis-open.org/ns/cti#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix stix: <http://docs.oasis-open.org/ns/cti/stix#> .
@prefix stix-semex: <http://docs.oasis-open.org/ns/cti/stix-semex#> .
@prefix data-marking: <http://docs.oasis-open.org/ns/cti/data-marking#> .
@base <https://people.cs.kuleuven.be/~davy.preuveneers/ns/cti/attack-semex#> .

<https://people.cs.kuleuven.be/~davy.preuveneers/ns/cti/attack-semex> rdf:type owl:Ontology ;
     owl:imports <http://docs.oasis-open.org/ns/cti/stix-semex> .

##################################################################
# Object properties
##################################################################

:tactic_refs rdf:type owl:ObjectProperty ;
             rdfs:range :Tactic .

:x_mitre_modified_by_ref rdf:type owl:ObjectProperty ;
              rdfs:range cti:Identity .

##################################################################
# Data properties
##################################################################

:x_mitre_collection_layers rdf:type owl:DatatypeProperty ;
                           rdfs:range xsd:string .

:x_mitre_detection rdf:type owl:DatatypeProperty ,
                            owl:FunctionalProperty ;
                   rdfs:range xsd:string .

:x_mitre_is_subtechnique rdf:type owl:DatatypeProperty ;
                         rdfs:range xsd:boolean .

##################################################################
# Classes
##################################################################

:Matrix rdf:type owl:Class .

:Tactic rdf:type owl:Class .

:Technique rdf:type owl:Class .

:Mitigation rdf:type owl:Class .

stix:AttackPattern owl:equivalentClass :Technique .

stix:CourseOfAction owl:equivalentClass :Mitigation .

:SubTechnique rdf:type owl:Class ;
              rdfs:subClassOf :Technique ,
                               [ rdf:type owl:Restriction ;
                                 owl:onProperty :x_mitre_is_subtechnique ;
                                 owl:hasValue "true"^^xsd:boolean
                                ] .
```

Listing 2. Instantiating MITRE ATT&CK technique T1111 (see https://attack.mitre.org/techniques/T1111 (accessed on 23 January 2024)) with our ATT&CK Semantic Extension Ontology.

```
@prefix : <https://people.cs.kuleuven.be/~davy.preuveneers/ns/cti/attack-semex#> .
@prefix cti: <http://docs.oasis-open.org/ns/cti#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix stix: <http://docs.oasis-open.org/ns/cti/stix#> .
@prefix stix-semex: <http://docs.oasis-open.org/ns/cti/stix-semex#> .
@prefix data-marking: <http://docs.oasis-open.org/ns/cti/data-marking#> .
@base <https://people.cs.kuleuven.be/~davy.preuveneers/ns/cti/attack-semex#> .

<https://people.cs.kuleuven.be/~davy.preuveneers/ns/cti/attack-semex> rdf:type owl:Ontology ;
    owl:imports <http://docs.oasis-open.org/ns/cti/stix-semex> .

:attack-pattern--dd43c543-bb85-4a6f-aa6e-160d90d06a49 rdf:type owl:NamedIndividual ,
                                                               attack-semex:Technique ;
   cti:external_references :external_reference--46e7b238-dcea-4121-99ee-23d5c3827dae ,
                           :external_reference--67b54962-33e8-4dbf-ad13-f4d25941194a ,
                           :external_reference--9ed23d0d-b28c-40e1-8aa4-4dbb925fd43a ,
                           :external_reference--ca1662b9-fb12-4978-a814-bad02d51312b ;
   data-marking:object_marking_refs :marking-definition--fa42a846-8d90-4e51-bc29-71d5b4802168 ;
   stix:kill_chain_phases :kill_chain_phase--5e6568d4-5020-43bb-aedc-ba2a6bbf79c3 ;
   attack-semex:x_mitre_modified_by_ref :identity--c78cb6e5-0c4b-4611-8297-d1b8b55e40b5 ;
   cti:created "2017-05-31T21:31:23.195Z"^^xsd:dateTime ;
   cti:description "Adversaries may target multi-factor authentication (MFA) mechanisms, ..." ;
   cti:id "attack-pattern--dd43c543-bb85-4a6f-aa6e-160d90d06a49" ;
   cti:modified "2023-05-09T14:00:00.188Z"^^xsd:dateTime ;
   cti:name "Multi-Factor Authentication Interception" ;
   cti:spec_version "2.1" ;
   cti:type "attack-pattern" ;
   attack-semex:x_mitre_attack_spec_version "3.1.0" ;
   attack-semex:x_mitre_contributors "John Lambert, Microsoft Threat Intelligence Center" ;
   attack-semex:x_mitre_data_sources "Driver: Driver Load" ,
                                     "Process: OS API Execution" ,
                                     "Windows Registry: Windows Registry Key Modification" ;
   attack-semex:x_mitre_deprecated "false"^^xsd:boolean ;
   attack-semex:x_mitre_detection "Detecting use of proxied smart card connections by an ..." ;
   attack-semex:x_mitre_domains "enterprise-attack" ;
   attack-semex:x_mitre_is_subtechnique "false"^^xsd:boolean ;
   attack-semex:x_mitre_platforms "Linux" ,
                                  "Windows" ,
                                  "macOS" ;
   attack-semex:x_mitre_version "2.1" .


:external_reference--46e7b238-dcea-4121-99ee-23d5c3827dae rdf:type owl:NamedIndividual ,
                                                                   cti:ExternalReference ;
   cti:external_id "T1111" ;
   cti:source_name "mitre-attack" ;
   cti:url "https://attack.mitre.org/techniques/T1111"^^xsd:anyURI .

:identity--c78cb6e5-0c4b-4611-8297-d1b8b55e40b5 rdf:type owl:NamedIndividual ,
                                                         cti:Identity .

:kill_chain_phase--5e6568d4-5020-43bb-aedc-ba2a6bbf79c3 rdf:type owl:NamedIndividual ,
                                                                 stix:KillChainPhase ;
   stix:kill_chain_name "mitre-attack" ;
   stix:phase_name "credential-access" .
...
```

The MITRE ATT&CK (version 14.1) knowledge base, including its three threat matrices, has undergone conversion into the Turtle and Resource Description Framework (RDF) format, a W3C standard for describing web resources and data interchange. This facilitates straightforward reasoning and querying using the Protégé (version 5.6.3) tool [17], a tool primarily used for building and managing ontologies:

1. **Enterprise Threat Matrix**:
   - Turtle format: https://people.cs.kuleuven.be/~davy.preuveneers/ns/cti/attack-semex-enterprise.ttl (accessed on 23 January 2024)
   - RDF format: https://people.cs.kuleuven.be/~davy.preuveneers/ns/cti/attack-semex-enterprise.rdf (accessed on 23 January 2024)
2. **Industrial Control Systems (ICSs) Threat Matrix**:
   - Turtle format: https://people.cs.kuleuven.be/~davy.preuveneers/ns/cti/attack-semex-ics.ttl (accessed on 23 January 2024)
   - RDF format: https://people.cs.kuleuven.be/~davy.preuveneers/ns/cti/attack-semex-ics.rdf (accessed on 23 January 2024)
3. **Mobile Threat Matrix**:
   - Turtle format: https://people.cs.kuleuven.be/~davy.preuveneers/ns/cti/attack-semex-mobile.ttl (accessed on 23 January 2024)
   - RDF format: https://people.cs.kuleuven.be/~davy.preuveneers/ns/cti/attack-semex-mobile.rdf (accessed on 23 January 2024)

We implement the same transformation of the MITRE ATLAS Data v4.5.1 (see https://github.com/mitre-atlas/atlas-data (accessed on 23 January 2024)), which includes 14 tactics, 46 techniques, 36 sub-techniques, 20 mitigations, and 22 case studies. Despite adhering to the same design principles as the MITRE ATT&CK framework, this knowledge base is presented in YAML format [18], as illustrated in Appendix B in Listing A3 for the Reconnaissance (AML.TA0002) tactic.

Fortunately, the transformation of MITRE ATLAS into the STIX format has already been carried out by the ATLAS Navigator Data project, accessible at https://github.com/mitre-atlas/atlas-navigator-data (accessed on 23 January 2024), offering an ATLAS-only STIX representation and one that incorporates and references the MITRE ATT&CK Enterprise knowledge base. We used the latter to create a semantically enriched knowledge base in Turtle format and RDF format:

- Turtle format: https://people.cs.kuleuven.be/~davy.preuveneers/ns/cti/atlas-attack-semex-enterprise.ttl (accessed on 23 January 2024)
- RDF format: https://people.cs.kuleuven.be/~davy.preuveneers/ns/cti/atlas-attack-semex-enterprise.rdf (accessed on 23 January 2024)

3.3. Instantiating the Augmented Ontology for AI-Powered Applications

While MITRE ATLAS proves to be a valuable resource, it is not without practical limitations when serving as the cornerstone of an ontology-based cybersecurity framework that aims to automate the vulnerability assessment in a continuously evolving threat landscape of novel ML attacks and defenses. One notable limitation is that it lacks practical and detailed information about threats and mitigations, especially in a machine interpretable manner. To illustrate, the tactic 'Reconnaissance' (AML.TA0002) includes the technique 'Search for Victim's Publicly Available Research Materials' (AML.T0000) along with the sub-technique 'Journals and Conference Proceedings' (AML.T0000.000). The sub-technique is shown in Appendix B in Listing A4 and is available at https://atlas.mitre.org/techniques/AML.T0000.000 (accessed on 23 January 2024).

One of the shortcomings is that this body of knowledge remains too high-level to be practically useful for automated reasoning. Indeed, a tactic is the highest-level description of an adversary's behavior, while techniques give a more detailed description of behavior in the context of a tactic, and procedures an even lower-level, highly detailed description in the context of a technique [19]. The STIX representation of MITRE ATLAS is confined to tactics and techniques only, which proves inadequate for encapsulating the core nature of threats posed to ML-based systems and applications. It lacks the incorporation of detailed low-level procedures. In contrast, the YAML representation of ATLAS encompasses 22 case studies that document procedures delineating each step of a specific attack. An example is the 'Evasion of Deep Learning Detector for Malware C&C Traffic' (AML.CS0000) case study, available at https://atlas.mitre.org/studies/AML.CS0000/ (accessed on 23 January 2024), which provides associated tactics and techniques. It is noteworthy, however, that these textual descriptions are designed mainly for human analysts or security experts and do not facilitate machine interpretation. Furthermore, a given procedure may contain references to scientific papers or URLs pointing to the associated paper or relevant software code. Unfortunately, instead of citing a scientific paper by its distinctive digital object identifier (DOI), the publication is referenced in plain text, leaving it susceptible to potential textual mismatches.

Furthermore, the momentum of adversarial ML is propelled by a surge of conference and journal papers proposing novel attacks and adversarial techniques ranging from simple data manipulations to more sophisticated methods that leverage intricate knowledge of model architectures. Other references lead to blog posts, news articles, or project pages that might lack the essential details required for replicating an attack or devising a new mitigation strategy. As a result, deriving the interconnection between old and new attacks, as well as understanding how novel attacks compromise established defenses, based on just these references is a non-trivial task.

Simultaneously, the surge in proposed defense strategies reflects the urgency to mitigate the risks associated with adversarial attacks. Mitigations are also covered in MITRE ATLAS, as depicted in Appendix B in Listing A5 for the 'Model Hardening' (AML.M0003) mitigation. However, the landscape of adversarial ML is marked by an inherent cat-and-mouse game, where the introduction of new defenses often prompts the generation of more sophisticated attacks. This arms race raises questions about the long-term efficacy of existing defenses and emphasizes the need for a continuous reassessment of security measures.

Unfortunately, the MITRE ATLAS knowledge base does not indicate which mitigations have been broken by new attacks. Indeed, while the mitigation of 'Model Hardening' (AML.M0003) proves effective against ML attack techniques like 'Evade ML Model' (AML.T0015) and 'Erode ML Model Integrity' (AML.T0031), a more detailed depiction of these attacks and defenses, along with the ongoing arms race between them, is essential. Such a nuanced representation is crucial for evaluating the current and future vulnerability status of a given application.

As a result, we have implemented an ontological extension specifically designed to articulate individual attacks and defenses at the procedure level, providing a more detailed and structured description within the context of a technique or a mitigation. Additionally, it organizes relevant scientific literature in a structured semantic manner, facilitating the seamless reconstruction of timelines involving attacks, mitigations, and subsequent new attack developments. Where relevant, we reuse established ontologies. For example, to document the scientific literature, citations, and cross-references, we adopt the Semantic Publishing and Referencing (SPAR) ontologies [20]. The case study mentioned earlier (i.e., 'Evasion of Deep Learning Detector for Malware C&C Traffic' (AML.CS0000), see https://atlas.mitre.org/studies/AML.CS0000/ (accessed on 23 January 2024)), mentions the scientific reference [21] as follows:

> *Le, Hung, et al. "URLNet: Learning a URL representation with deep learning for malicious URL detection." arXiv preprint arXiv:1802.03162 (2018).*

The semantic equivalent documented with the SPAR ontologies is shown in Appendix C in Listing A6. These ontologies offer the benefit of reusable definitions of authors with multiple scientific references, or show how subsequent research builds upon earlier contributions. The latter is exemplified by the study on GramBeddings [22] citing URLNet [21] for comparison, showcasing the extension of prior work. Other methods in the Citation Typing Ontology (CiTO) to refer to previous research beyond *usesMethodIn* include *supports, updates, usesDataFrom, extends, disputes, . . .* (see https://sparontologies.github.io/cito/current/cito.html (accessed on 23 January 2024)).

With our approach to semantically modeling in more detail the MITRE ATLAS procedures, as shown in Figure 1, we can document how, for example, certain mitigations are confirmed to work on other datasets, how new attacks or mitigations improve on the same datasets, how attacks break existing mitigations, or vice versa. The additional benefit compared to the case studies in the YAML-based MITRE ATLAS knowledge base is that the insights into new attacks and defenses can be reasoned upon for other case studies. The specification is available in Turtle format and RDF format at:

- Turtle format: https://people.cs.kuleuven.be/~davy.preuveneers/ns/cti/atlas-semex-procedure.ttl (accessed on 23 January 2024)
- RDF format: https://people.cs.kuleuven.be/~davy.preuveneers/ns/cti/atlas-semex-procedure.rdf (accessed on 23 January 2024)

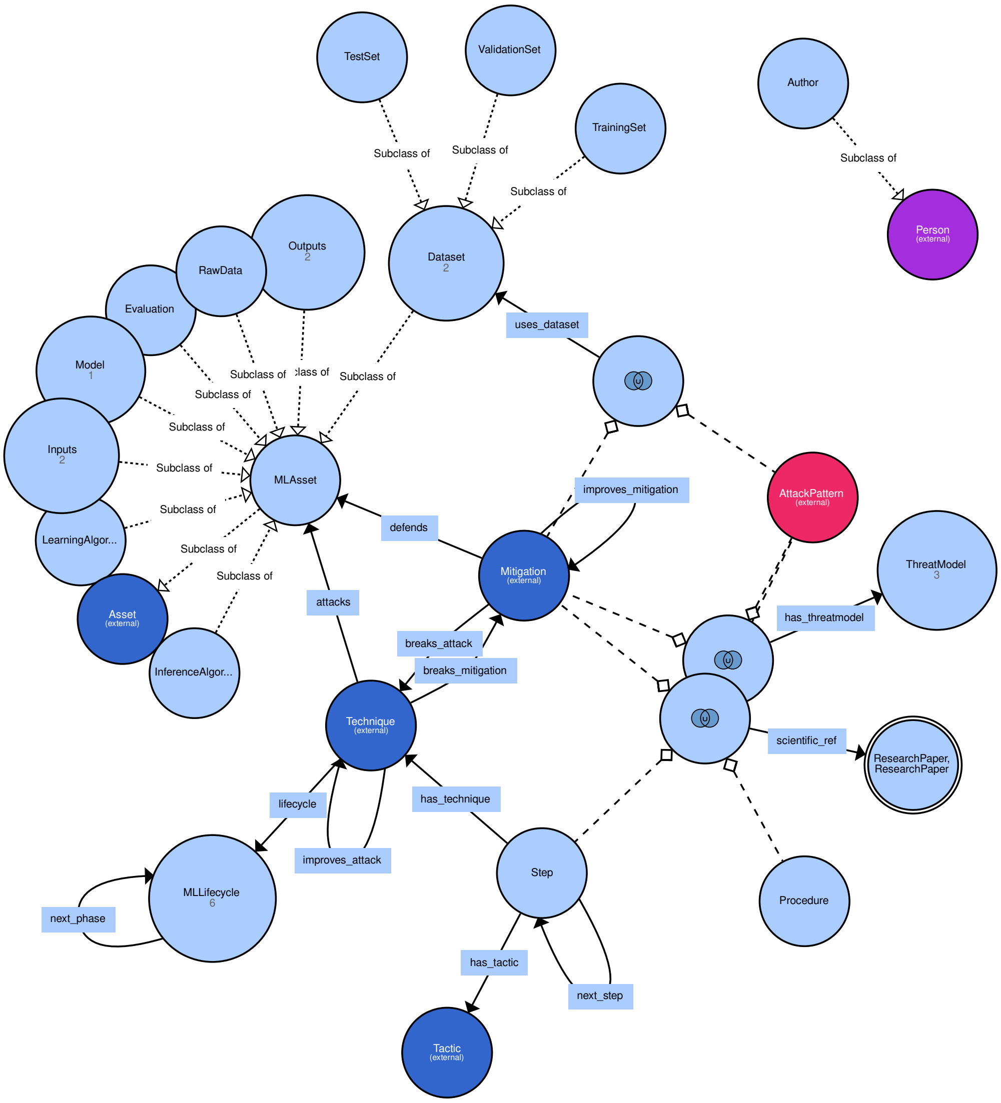


Figure 1. Semantically extending the MITRE ATLAS procedures to model the interplay between attacks and mitigations, as well as the associated scientific literature.

3.4. Implementation

The implementation of our ontology-based cybersecurity framework for AI-enabled systems and applications leverages the aforementioned STIX-based representation of the MITRE ATLAS Navigator Data v4.5.1 and the MITRE ATT&CK (version 14.1) Enterprise knowledge base. Furthermore, we instantiate the above ontologies with several scientific publications in the realm of adversarial ML, both offensive and defensive research, to analyze and reason upon concrete use cases. This adversarial ML knowledge base documents metadata and insights from about 60 scientific papers (see Listings 3 and 4). Obviously, the addition of new literature on adversarial ML is a continuous work-in-progress. However, the number of entries should be sufficient to ascertain the practical feasibility and computational impact of reasoning upon this knowledge base.

Listing 3. Semantic representation of an adversarial ML defense research paper (i.e., Li et al. [23]).

```
@prefix : <https://people.cs.kuleuven.be/~davy.preuveneers/ns/cti/atlas-semex-paper#> .
@prefix atlas-semex-procedure: <https://people.cs.kuleuven.be/~davy.preuveneers/ns/cti/atlas-semex-
   procedure#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
...

:xin_li rdf:type atlas-semex-procedure:Author ;
 foaf:name "Xin Li" ;
 foaf:givenName "Xin" ;
 foaf:familyName "Li" .

:fuxin_li rdf:type atlas-semex-procedure:Author ;
 foaf:name "Fuxin Li" ;
 foaf:givenName "Fuxin" ;
 foaf:familyName "Li" .

:arxiv rdf:type foaf:Organization ;
 foaf:name "arXiv" .

:1612.07767 rdf:type fabio:ResearchPaper ;
 dcterms:creator :xin_li, :fuxin_li ;
 fabio:hasURL "http://arxiv.org/abs/1612.07767"^^xsd:anyURI ;
 fabio:hasPublicationYear "2016"^^xsd:gYear ;
 dcterms:publisher :arxiv ;
 dcterms:title "Adversarial Examples Detection in Deep Networks with Convolutional Filter Statistics" .
```

Listing 4. Modeling ML attacks and defenses.

```
@prefix : <https://people.cs.kuleuven.be/~davy.preuveneers/ns/cti/experiment#> .
@prefix attack-semex: <https://people.cs.kuleuven.be/~davy.preuveneers/ns/cti/attack-semex#> .
@prefix atlas-semex-procedure: <https://people.cs.kuleuven.be/~davy.preuveneers/ns/cti/atlas-semex-
   procedure#> .
@prefix atlas-semex-paper: <https://people.cs.kuleuven.be/~davy.preuveneers/ns/cti/atlas-semex-paper#> .
...

:LiL16e rdf:type attack-semex:Mitigation ;
 # extends base MITRE ATLAS mitigation: 'Input Restoration' (AML.M0010)
 atlas-semex-procedure:extends atlas-semex:course-of-action--0df12e98-f47a-4126-8512-ff573cbfb6ea ;
 # extends base MITRE ATLAS mitigation: 'Adversarial Input Detection' (AML.M0015)
 atlas-semex-procedure:extends atlas-semex:course-of-action--37862c51-9708-45b5-b5a9-2ced6b96a68f ;
 # title: 'Adversarial Examples Detection in Deep Networks with Convolutional Filter Statistics'
 atlas-semex-procedure:scientific_ref atlas-semex-paper:1612.07767 .

:CarliniW17 rdf:type attack-semex:Technique ;
 # extends base MITRE ATLAS attack: 'Evade ML Model' (AML.T0015)
 atlas-semex-procedure:extends atlas-semex:attack-pattern--dbd25a74-6024-4e30-9ba2-20428a447b70 ;
 # title: 'Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods'
 atlas-semex-procedure:scientific_ref atlas-semex-paper:1705.07263 ;
 atlas-semex-procedure:uses_dataset atlas-semex-procedure:mnist ;
 atlas-semex-procedure:uses_dataset atlas-semex-procedure:cifar10 ;
 atlas-semex-procedure:has_threatmodel atlas-semex-procedure:perfect_knowledge_adversary ;
 atlas-semex-procedure:breaks_mitigation :LiL16e ;
 ...
```

We will illustrate the continuous arms race of adversarial attacks and defenses in an image classification context. One of the many mitigations against misclassification or evasion proposed in the ATLAS framework is 'Input Restoration' (AML.M0010). The objective is to preprocess the inference data, i.e., the input image, in order to eliminate or reverse any potential adversarial perturbations. The ATLAS knowledge base on this mitigation (see https://atlas.mitre.org/mitigations/AML.M0010 (accessed on 23 January 2024)) offers no suggestions on how exactly to accomplish this. Similar observations can be made for other mitigation strategies, such as 'Adversarial Input Detection' (AML.M0015). Our goal is to offer a knowledge base that is more detailed by complementing these mitigations with semantic references to scientific works, including the relationships between them.

For example, Li et al. [23] introduced a method involving a straightforward 3 × 3 average filter for image blurring prior to classification. The concept behind this simple defense is to mitigate adversarial examples created through the fast gradient sign attack [24]. Numerous other defenses have been proposed in the literature. Nonetheless, as underscored by Carlini and Wagner [25], the identification of adversarial examples proves to be a nontrivial task. They assessed 10 proposed defenses, revealing their susceptibility to white-box attacks. This was accomplished by formulating defense-specific loss functions, subsequently minimized using a strong iterative attack algorithm. Employing these methodologies on the CIFAR-10 dataset, they demonstrated an adversary's ability to generate imperceptible adversarial examples for every defense.

Whereas the previous works focused on evasion attacks and defenses, a similar arms race is happening for other types of ML attacks. As inspiration, we explore the poisoning attack survey by Cinà et al. [26]. In this extensive survey, the authors offer a thorough systematization of poisoning attacks and defenses in ML, scrutinizing over 100 papers published in the field over the past 15 years. Their approach begins by categorizing prevailing threat models and attacks, followed by the systematic organization of existing defenses.

Rather than explaining attacks and defenses in detail and how they were subsequently broken, our goal is to incorporate the knowledge about these attacks and defenses into our ontology-based cybersecurity framework with the following steps:

1. Semantically describe the ML pipeline, including the type of inputs and outputs.
2. Semantically describe the dataset artifacts used in the attack/defense experiments.
3. Semantically describe the threat model (e.g., black box vs. white box access to model).
4. Semantically describe each scientific publication on adversarial attacks and defenses using the SPAR ontologies, as illustrated earlier in Listings A6, 3, and 4.
5. Semantically link the publication with the ML pipeline, the threat model, as well as the corresponding techniques or mitigations of MITRE ATLAS.
6. Semantically link how defenses have been broken with new attacks, or vice versa, with our proposed MITRE ATLAS Semantic Extension Ontology, as depicted in Figure 1.

To evaluate, a variety of queries can be directed to the description logic reasoner to solicit information. Typically, these description logic queries are expressed in the Manchester OWL (Web Ontology Language) syntax [27], as illustrated in Figure 2. The queries may pertain to specific defenses that have been compromised, reported attacks for particular input data modalities (e.g., malware), operations of attacks within a black-box threat model context, identification of authors employing specific datasets, and similar topics of interest.

Query: Return attacks that broke any mitigation

Manchester OWL Syntax: (breaks_mitigation **some** Mitigation)

Query: Return attacks that broke a specific mitigation (i.e., Li et al. 2016)

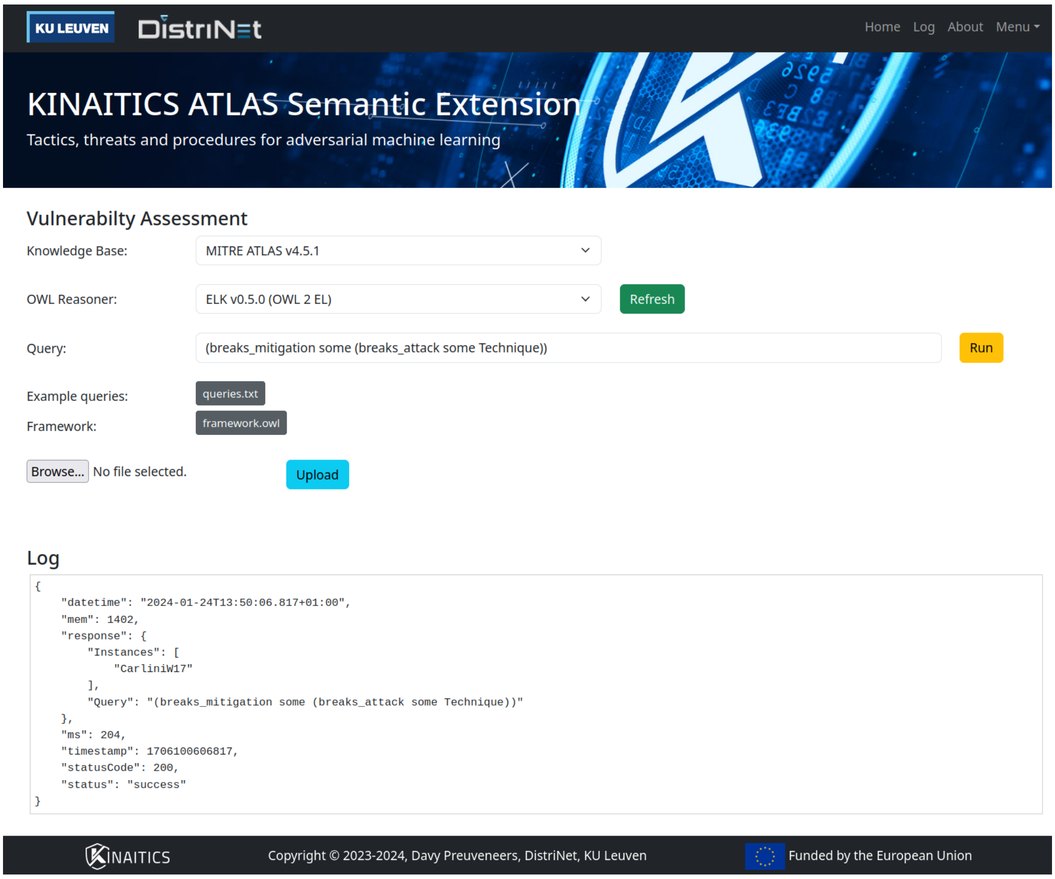
Manchester OWL Syntax: (breaks_mitigation **value** LiL16e)

Query: Return mitigations that extend the base MITRE ATLAS mitigation with name 'Adversarial Input Detection' (i.e., https://atlas.mitre.org/mitigations/AML.M0015)

Manchester OWL Syntax: (extends **some** (name **value** "Adversarial Input Detection"))

Figure 2. DL query examples in Manchester OWL syntax. The outputs are individuals that can be subsequently queried for more details [23].

In addition to analyzing the knowledge base with the Protégé tool [17], we have developed—as part of the Horizon Europe KINAITICS project (see https://kinaitics.eu (accessed on 23 January 2024))—a proof-of-concept software framework to raise the level of abstraction for end-users and facilitate the collaboration with other security tools by offering the knowledge base and reasoning capabilities through a RESTful interface. This framework comprises a Java-based Spring Boot (version 3.2.2) back-end application and an HTML5 user interface built with Bootstrap (version 5.3.2), as illustrated in Figure 3, and running on top of OpenJDK (version 17.0.9). The back end utilizes OWL reasoners and their descriptive logic (DL) capabilities to assess the vulnerability of a specific AI-enabled application through predefined DL queries declared in the Manchester OWL syntax. The aspiration is to further enhance the proof-of-concept implementation by incorporating a dashboard interface akin to the MITRE ATLAS Navigator (see https://atlas.mitre.org/navigator/ (accessed on 23 January 2024)).



KINAITICS ATLAS Semantic Extension

Tactics, threats and procedures for adversarial machine learning

Vulnerabilty Assessment

Knowledge Base: MITRE ATLAS v4.5.1

OWL Reasoner: ELK v0.5.0 (OWL 2 EL)

Query: (breaks_mitigation some (breaks_attack some Technique))

Example queries: queries.txt

Framework: framework.owl

No file selected.

Log

```
{
    "datetime": "2024-01-24T13:50:06.817+01:00",
    "mem": 1402,
    "response": {
        "Instances": [
            "CarliniW17"
        ],
        "Query": "(breaks_mitigation some (breaks_attack some Technique))"
    },
    "ms": 204,
    "timestamp": 1706100606817,
    "statusCode": 200,
    "status": "success"
}
```

Copyright © 2023-2024, Davy Preuveneers, DistriNet, KU Leuven

Funded by the European Union

Figure 3. Spring Boot-based cybersecurity framework for vulnerability assessment with MITRE ATLAS semantic extensions.

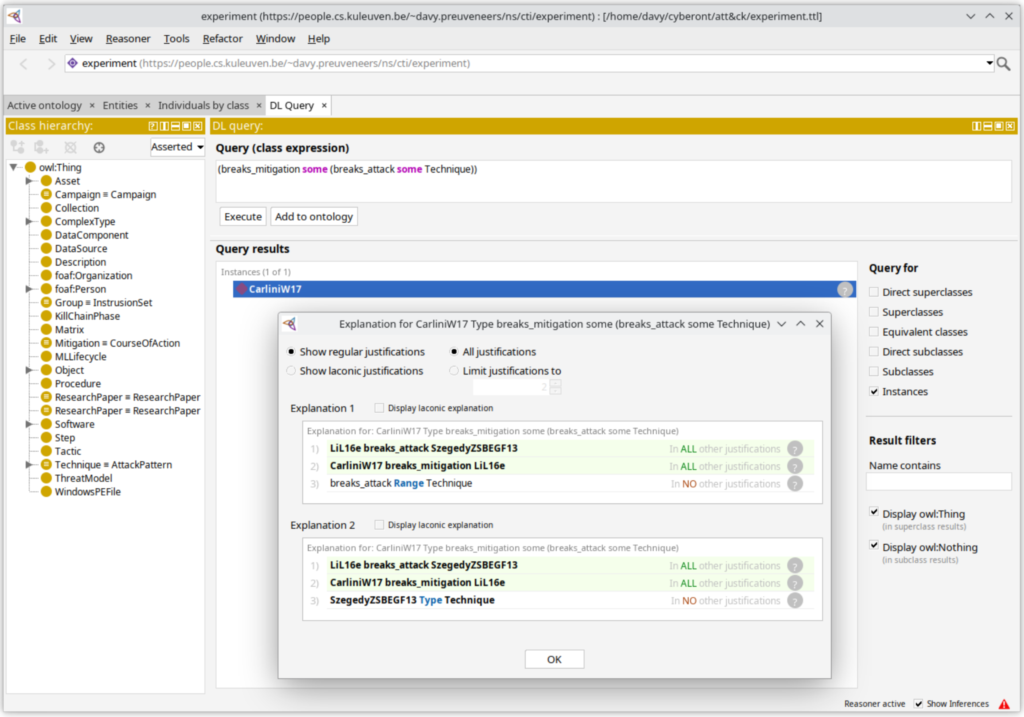
4. Evaluation

In this section, we validate our cybersecurity framework by leveraging the knowledge bases mentioned earlier and illustrate the main benefits as well as the practical feasibility of our framework to reason upon a large body of knowledge. Our framework and the benchmark application are executed in Ubuntu (version 23.10), operating on an HP ZBook Power 15.6 inch G8 Mobile Workstation PC featuring an 11th Gen Intel Core i7-11800H running at 2.30 GHz and 32 GB RAM.

4.1. Qualitative Evaluation: Adaptability of the Cybersecurity Framework

Modifying, adding, or removing an attack or defense is facilitated through instantiating ontology classes and properties. The ontology-based approach of our cybersecurity framework enables the incorporation of new attacks and defenses without necessitating changes to the software (see Figure 3). The rationale behind embedding a description logic-based ontology reasoner into our software, as opposed to implementing rigid if-then-else rules, is two-fold. Firstly, the reasoner infers implicit interdependencies between attacks and defenses by leveraging semantic relationships, a task not easily achieved with hard-coded if-then-else rules. Secondly, the ontology readily accommodates the addition of new knowledge, thereby enhancing the adaptability of our cybersecurity framework to new attacks and defenses, albeit at the expense of an ontology reasoner being more computational intensive than a lightweight rule engine.

Additionally, the ontology reasoner can also be used to explain certain inferences, such as depicted in Figure 4 with the Protégé tool for the Manchester OWL query in Figure 5.



Query (class expression)

(breaks_mitigation some (breaks_attack some Technique))

Query results

Instances (1 of 1)

CarliniW17

Explanation for CarliniW17 Type breaks_mitigation some (breaks_attack some Technique)

● Show regular justifications ● All justifications
○ Show laconic justifications ○ Limit justifications to 2

Explanation 1 □ Display laconic explanation

Explanation for: CarliniW17 Type breaks_mitigation some (breaks_attack some Technique)

1) LiL16e breaks_attack SzegedyZSBEGF13 In ALL other justifications
2) CarliniW17 breaks_mitigation LiL16e In ALL other justifications
3) breaks_attack Range Technique In NO other justifications

Explanation 2 □ Display laconic explanation

Explanation for: CarliniW17 Type breaks_mitigation some (breaks_attack some Technique)

1) LiL16e breaks_attack SzegedyZSBEGF13 In ALL other justifications
2) CarliniW17 breaks_mitigation LiL16e In ALL other justifications
3) SzegedyZSBEGF13 Type Technique In NO other justifications

Figure 4. Explaining the results of Manchester OWL queries.

Query: The arms race of mitigations against attacks that are subsequently broken again

Manchester OWL Syntax: (breaks_mitigation **some** (breaks_attack **some** Technique))

Figure 5. DL query examples in Manchester OWL syntax. The outputs are individuals that can be subsequently queried for more details.

The ability to explain inferences offers transparency about how query results are derived from the underlying knowledge base. Security analysts can gain a deeper understanding of the relationships and dependencies among different entities and concepts within the adversarial ML domain. By explaining inferences, they can gain insights into the hidden or inferred relationships, helping them discover new knowledge without explicitly encoding it in the ontology.

Also, when inconsistencies or unexpected inferences arise, the analysts can use the ontology reasoner to diagnose and trace the root cause of the issues. This helps in identifying and rectifying errors in the ontology or refining the knowledge representation.

4.2. Quantitative Evaluation: Performance

The backbone of our ontology-based cybersecurity framework is powered by an OWL 2 reasoner. Therefore, the careful selection of an optimal implementation of such a reasoner is paramount. We initially selected the HermiT ontology reasoner because of its strong compliance with the OWL 2 DL (description logic) specification and its efficient reasoning capabilities [28]. Additionally, the Protégé tool comes with HermiT pre-installed.

Our objective is to achieve a query response time below 3 s, deemed reasonable from a usability perspective. We then carried out performance assessments of our REST-based framework, deploying all components on a single system to minimize the impact of network latencies that would otherwise arise when submitting queries and retrieving the results.

4.2.1. Experimental Setup

With a basic shell script that invokes the *curl* command line utility, we orchestrated the execution of 100 REST requests within distinct query categories against our framework, subsequently assessing the corresponding response times. The queries were systematically categorized into three complexity tiers, determined by query length and depth. Presented below are examples representing each complexity category of queries:

- **Simple queries**: class instances or property restrictions
  - ResearchPaper
  - (breaks_mitigation **value** LiL16e)
  - (breaks_mitigation **some** Mitigation)
- **Normal queries**: nested property restrictions
  - (extends **some** (name **value** “Adversarial Input Detection”))
  - (breaks_mitigation **some** (breaks_attack **some** Technique))
  - (breaks_attack **some** (breaks_mitigation **value** LiL16e))
- **Complex queries**: multi-nested property restrictions and conjunctions
  - ((extends **some** (external_references **some** (external_id **value** “AML.T0015”))) **and** (breaks_mitigation **value** LiL16e))
  - (breaks_mitigation **some** (breaks_attack **some** (breaks_mitigation **some** (breaks_attack **some** Technique))))
  - ((extends **some** (name **value** “Evade ML Model”)) **and** (has_threatmodel **value** perfect_knowledge_adversary) **and** (uses_dataset **value** cifar10))

We will carry out two types of experiments, the first one without including the extensive MITRE ATT&CK (version 14.1) Enterprise knowledge base. As a result, if a query filters instances of the class *Technique*, it will not process those covered by the ATT&CK Enterprise knowledge base, only those covered by the ATLAS knowledge base. In the second experiment, we will include the ATT&CK Enterprise data, expecting that this will increase memory usage, computational complexity, as well the query response times.

4.2.2. HermiT Reasoner: Query Response Times

Figure 6 depicts the response times in milliseconds for our knowledge base, without loading MITRE ATT&CK (version 14.1) Enterprise. The box plot lines depict the median (50%), upper quartile (75%), and lower quartile (25%), while the average is represented by the green diamond, and outliers (if any) are denoted by white circles. The average response times fall below 3 s. The maximum heap memory usage by our Spring Boot-based framework is about 255 MB.

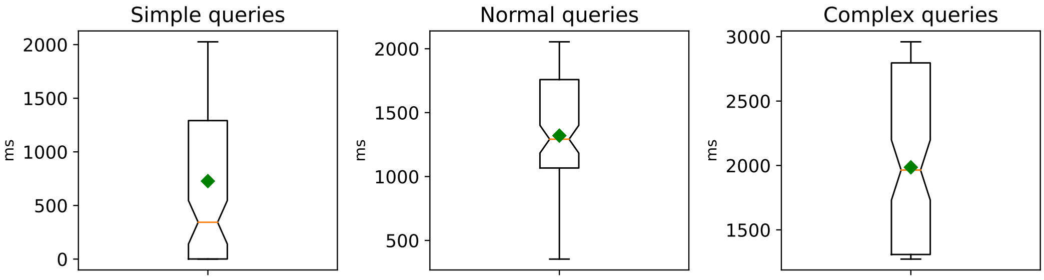


Figure 6. *HermiT* query response times *without* the MITRE ATT&CK Enterprise knowledge base.

Upon incorporating MITRE ATT&CK (version 14.1) Enterprise into our knowledge base and maintaining the same set of queries, there is a substantial impact on resource utilization. Note how the overall heap memory usage increases from 255 MB to 3490 MB. Due to response times exceeding several hours, even for a single straightforward query, we were unable to finish the 100 queries for each complexity category. It became evident that the efficiency of the HermiT reasoner did not meet our expectations. However, if the security analyst's needs are limited to the MITRE ATLAS knowledge base enriched with our ontologies and semantic references to the scientific literature, HermiT remains a sufficiently capable reasoner.

4.2.3. Openllet Reasoner: Query Response Times

After replacing the HermiT reasoner in our cybersecurity framework with the Openllet OWL 2 DL reasoner (version 2.6.5) (see https://github.com/Galigator/openllet (accessed on 23 January 2024)), we observed a significant enhancement in query response times, improving by at least an order of magnitude. We conducted the same experiments as before. Figure 7 illustrates the query response times without the ATT&CK Enterprise knowledge base, while Figure 8 presents the results when incorporating the ATT&CK Enterprise knowledge base. The maximum heap memory usage is 313 MB and 13,297 MB, respectively.

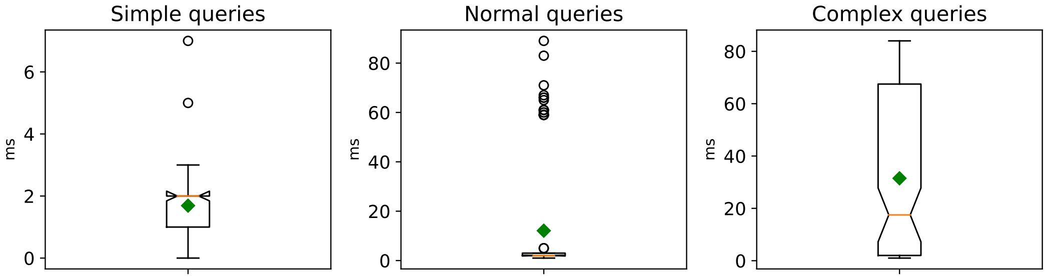


Figure 7. *Openllet* query response times *without* the MITRE ATT&CK Enterprise knowledge base.

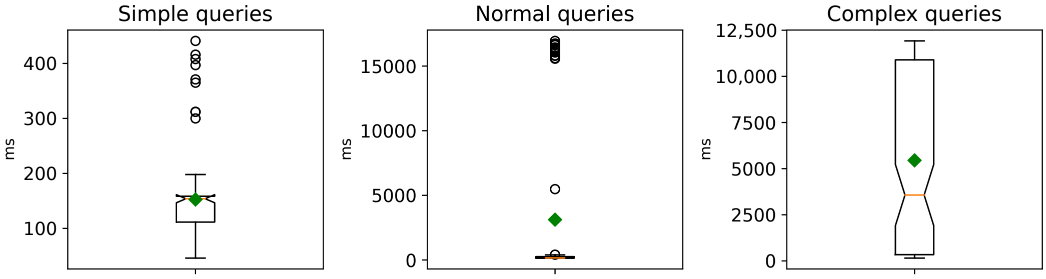


Figure 8. *Openllet* query response times *with* the MITRE ATT&CK Enterprise knowledge base.

The presented performance benchmarks affirm the practical viability of our approach, even though more intricate queries may experience a higher query response time when applied to a sizable knowledge base. Nevertheless, if the focus of the security analyst is solely on managing ML attacks and defenses—i.e., excluding the traditional tactics, techniques, and procedures from MITRE ATT&CK—then the overhead can be further minimized. Nonetheless, given the expanding scientific literature and knowledge base on adversarial ML, careful monitoring is essential to uphold query response times below an acceptable threshold.

4.2.4. ELK Reasoner: Query Response Times

The Protégé tool also ships with the ELK reasoner [16]. The main difference with the two previous OWL reasoners is that ELK is an OWL 2 EL reasoner. It is more restrictive in that it limits the use of certain OWL constructs, such as disjunctions and certain types of cardinality restrictions. It is designed to provide a good balance between expressivity and computational tractability. This makes ELK well-suited for scalable reasoning over large ontologies.

We replicate the same pair of experiments conducted previously. The response times for queries using ELK reasoner (version 0.5.0) are illustrated in Figures 9 and 10 for both scenarios. The maximum heap memory usage is 207 MB and 2361 MB, respectively. Evidently, the ELK reasoner emerges as the top-performing choice, showcasing query response times consistently below 1 s.

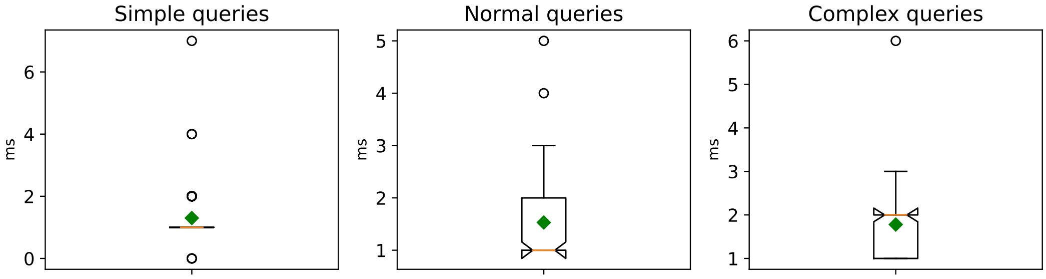


Figure 9. *ELK* query response times *without* the MITRE ATT&CK Enterprise knowledge base.

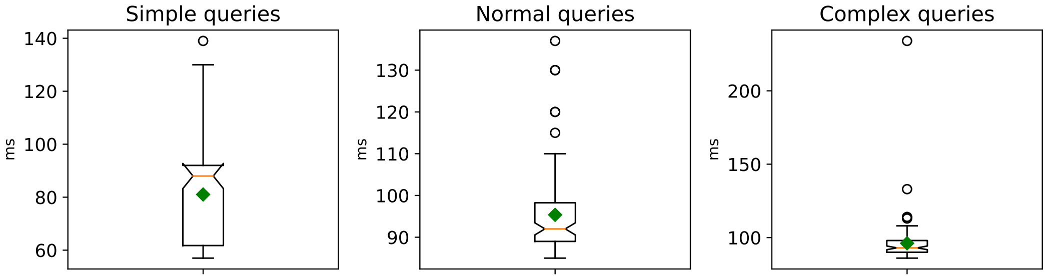


Figure 10. *ELK* query response times *with* the MITRE ATT&CK Enterprise knowledge base.

For the kinds of queries included in our experiments, we did not observe any impact from the restrictions imposed by the OWL 2 EL reasoner. Consequently, the ELK reasoner has not only demonstrated practical feasibility but has also indicated that the knowledge base can be further extended without causing significant concerns about query response times.

4.3. Discussion and Limitations

Our cybersecurity framework and proof-of-concept implementation meets the targeted objectives, but it is not without its limitations. While we are able to model new knowledge about ML attacks and defenses, including the arms race between them, it is very difficult to take into account the probabilistic nature of ML. Indeed, outcomes of attacks and defenses are measured by probabilities (for example, for a given ML model, dataset and implemented defenses, one previously successful attack may now only succeed in 10% of the cases, whereas a new one may have a higher attack success rate), and this poses a non-trivial difficulty in representing how well a defense can mitigate an attack, or vice versa. Quantifying the efficacy of either an attack or a defense, and subsequently engaging in reasoning based on these metrics, falls outside the realm of this research.

Our framework draws significant inspiration from the MITRE ATT&CK and ATLAS knowledge bases, but that also limits its scope. For example, an AI system that makes automated decisions without human intervention or processes sensitive data may raise important privacy and ethics concerns. Extending our framework with a knowledge base and reasoning capabilities for these kinds of privacy threats is an opportunity for further research.

Our design choices have led to a distinct separation between the ontology classes 'Technique' (also known as 'AttackPattern' in STIX language) and 'Mitigation' (also known as 'CourseOfAction' in STIX language). However, in specific application scenarios, adversarial ML attacks can transition from being considered attacks to serving as defense mechanisms. For instance, a technique designed to generate an adversarial example of a person's picture could be categorized as an impersonation attack if the goal is to deceive the ML classifier of a face recognition system [29] to gain unauthorized access. Interestingly, the same method can be employed as a defense strategy against mass surveillance when the objective is to mislead the face recognition model into identifying the adversarial face example as someone else [30]. The classification of a method as either an attack or a defense depends on the stakeholder's perspective, whether it be the developer of the security system or the individual whose picture is involved. The current framework lacks support for this multi-stakeholder perspective.

5. Conclusions

In this research, we enhance existing cybersecurity ontologies to address the unique challenges of applications, systems, and services that heavily rely on AI, including ML and DL, for their decision-making processes. Our primary focus is on semantically documenting prevalent threats in the realm of ML and formulating effective countermeasures. This includes an exploration of the mechanisms through which emerging attacks can potentially break existing defenses, along with the ongoing arms race between these threats and defenses.

Our cybersecurity framework draws inspiration from the MITRE ATT&CK and ATLAS knowledge bases. While these taxonomies provide a hierarchical structure to categorize concepts related to adversarial behavior, they face difficulties in expressing complex relationships and in effectively reasoning about both attacks and defenses. This challenge becomes particularly pronounced in the dynamic context of the ongoing arms race between adversarial techniques and defensive measures. Our solution can systematically formalize a comprehensive body of knowledge. This knowledge is tailored to augment existing taxonomies and threat modeling approaches, specifically for applications powered by AI. Leveraging advanced semantic reasoning capabilities, our framework facilitates the automated assessment of these applications. Moreover, through experimental performance benchmarks with our framework, we not only showcased its practical feasibility but also revealed that the computational impact of employing sophisticated ontology reasoners is within acceptable bounds.

As next steps, we plan to continually expand the knowledge base with new attacks and defenses found in the scientific literature. Other application areas we aim to include are the MITRE ATT&CK knowledge bases and threat matrices for Mobile and Industrial Control Systems (ICSs), especially when they cover AI-related threats and defenses [31,32]. Additionally, we plan to enhance the front end of our framework, offering an enriched user interface akin to well-known threat navigator dashboards. This continuous process ensures that the knowledge base behind our framework remains up-to-date, adaptable, and accessible in a user-friendly manner in an ever-evolving landscape of cybersecurity threats and defenses of AI-enabled systems and applications.

Author Contributions: Conceptualization, methodology, software, and validation, D.P.; writing—original draft preparation, D.P.; writing—review and editing, D.P.; supervision, D.P. and W.J.; project administration, D.P.; funding acquisition, W.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research is partially funded by the Research Fund KU Leuven and by the Flemish Research Programme Cybersecurity. Work for this paper was supported by the European Commission through the Horizon Europe project KINAITICS (https://kinaitics.eu/ (accessed on 23 January 2024)) under grant agreement 101070176.

Data Availability Statement: The data presented in this study are available in this article.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Processing the STIX 2.1 Knowledge Base with Python

Listings A1 and A2 show how to process STIX 2.0 JSON files, respectively, with the Python stix2 (version 3.0.1) package and the Python mitreattack-python (version 3.0.2) package.

Listing A1. Processing the ATT&CK (version 14.1) Enterprise knowledge base in STIX format and filtering the first entry with an external reference to https://attack.mitre.org/techniques/T1111 (accessed on 23 January 2024).

```python
from stix2 import MemoryStore
from stix2 import Filter

src = MemoryStore()
src.load_from_file("enterprise-attack/enterprise-attack-14.1.json")
t1111 = src.query([ Filter("external_references.external_id", "=", "T1111"),
   Filter("type", "=", "attack-pattern") ])[0]
print(t1111)
```

Listing A2. Processing the ATT&CK (version 14.1) Enterprise knowledge base in STIX format with the mittreattack-python (version 3.0.2) instead of the stix2 (version 3.0.1) package.

```python
from mitreattack.stix20 import MitreAttackData

mitre_attack_data = MitreAttackData("enterprise-attack/enterprise-attack-14.1.json")
for attack_pattern in mitre_attack_data.get_objects_by_type("attack-pattern"):
   for external_reference in attack_pattern["external_references"]:
      if "external_id" in external_reference and external_reference["external_id"] == "T1111":
         t1111 = attack_pattern
         break
print(t1111)
```

Appendix B. YAML Representation of ATLAS Tactics, Techniques, and Mitigations

Listings A3, A4, and A5 respectively depict a tactic, technique, and mitigation from the MITRE ATLAS knowledge base in YAML format.

Listing A3. YAML description of one of the tactics in the ATLAS Machine Learning Threat Matrix.

```yaml
- id: AML.TA0002
  name: Reconnaissance
  object-type: tactic
  ATT&CK-reference:
    id: TA0043
    url: https://attack.mitre.org/tactics/TA0043/
  description: ’The adversary is trying to gather information about the machine
    learning system they can use to plan future operations.

    Reconnaissance consists of techniques that involve adversaries actively or passively
    gathering information that can be used to support targeting.

    Such information may include details of the victim organization‘s machine learning
    capabilities and research efforts.

    This information can be leveraged by the adversary to aid in other phases of
    the adversary lifecycle, such as using gathered information to obtain relevant
    ML artifacts, targeting ML capabilities used by the victim, tailoring attacks
    to the particular models used by the victim, or to drive and lead further Reconnaissance
    efforts.
    ’
```

Listing A4. YAML description of the *Journals and Conference Proceedings (AML.T0000.000)* subtechnique in MITRE ATLAS.

```
- id: AML.T0000.000
  name: Journals and Conference Proceedings
  object-type: technique
  description: ’Many of the publications accepted at premier machine learning conferences
    and journals come from commercial labs.

    Some journals and conferences are open access, others may require paying for
    access or a membership.

    These publications will often describe in detail all aspects of a particular
    approach for reproducibility.

    This information can be used by adversaries to implement the paper.
    ’
  subtechnique-of: AML.T0000
```

Listing A5. YAML description of the *Model Hardening (AML.M0003)* mitigation in MITRE ATLAS.

```
- id: AML.M0003
  name: Model Hardening
  object-type: mitigation
  category:
  - Technical - ML
  ML-lifecycle:
  - Data Preparation
  - ML Model Engineering
  description: ’Use techniques to make machine learning models robust to adversarial
    inputs such as adversarial training or network distillation.’
  techniques:
  - id: AML.T0015
    use: ’Hardened models are more difficult to evade.’
  - id: AML.T0031
    use: ’Hardened models are less susceptible to integrity attacks.’
```

Appendix C. Semantic Representation of Scientific Literature

Listing A6 illustrates how scientific literature is semantically represented with the SPAR ontologies.

Listing A6. Semantic representation of scientific literature and newer work.

```
@prefix : <http://www.sparontologies.net/example/> .
@prefix application: <http://purl.org/NET/mediatypes/application/> .
@prefix fabio: <http://purl.org/spar/fabio/> .
@prefix frbr: <http://purl.org/vocab/frbr/core#> .
@prefix prism: <http://prismstandard.org/namespaces/basic/2.0/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix cito: <http://purl.org/spar/cito/> .

:urlnet2018 rdf:type fabio:ResearchPaper ;
 dcterms:creator :hungle, :quangpham, :doyensahoo, :stevenchhoi ;
 frbr:realization :urlnet2018-version-of-record .

:urlnet2018-version-of-record rdf:type fabio:Article ;
 dcterms:title "URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection" ;
 fabio:hasPublicationYear "2018"^^xsd:gYear ;
 prism:doi "10.48550/arXiv.1802.03162" ;
 frbr:embodiment :pdf .

:pdf rdf:type fabio:DigitalManifestation ;
 dcterms:publisher :arxiv ;
 dcterms:format application:pdf ;
 prism:publicationDate "2018-03-02"^^xsd:date .

:arxiv rdf:type foaf:Organization ;
 foaf:name "arXiv" .

:hungle rdf:type foaf:Person ;
 foaf:name "Hung Le" ;
 foaf:givenName "Hung" ;
 foaf:familyName "Le" .

:quangpham rdf:type foaf:Person ;
 foaf:name "Quang Pham" ;
 foaf:givenName "Quang" ;
 foaf:familyName "Pham" .

:doyensahoo rdf:type foaf:Person ;
 foaf:name "Doyen Sahoo" ;
 foaf:givenName "Doyen" ;
 foaf:familyName "Sahoo" .

:stevenchhoi rdf:type foaf:Person ;
 foaf:name "Steven C.H. Hoi" ;
 foaf:givenName "Steven" ;
 foaf:familyName "Hoi" .


#################################################################
# Citations
#################################################################

:grambeddings2023 rdf:type fabio:ResearchPaper ;
 frbr:realization :grambeddings2023-version-of-record .

:grambeddings2023-version-of-record rdf:type fabio:JournalArticle ;
 dcterms:title "GramBeddings: A New Neural Network for URL Based Identification of Phishing Web Pages
   Through N-gram Embeddings" ;
 fabio:hasPublicationYear "2023"^^xsd:gYear ;
 prism:doi "10.1016/j.cose.2022.102964" .

:grambeddings2023 cito:usesMethodIn :urlnet2018 .
```

References

1. Syed, Z.; Padia, A.; Finin, T.; Mathews, L.; Joshi, A. UCO: A Unified Cybersecurity Ontology. UMBC Student Collection. 2016. Available online: https://www.researchgate.net/publication/287195565_UCO_A_Unified_Cybersecurity_Ontology (accessed on 23 January 2024).
2. MITRE. ATLAS—Adversarial Threat Landscape for Artificial-Intelligence Systems (Website v3.6.0, Data v4.5.0). 2023. Available online: https://oecd.ai/en/catalogue/tools/atlas-adversarial-threat-landscape-for-artificial-intelligence-systems (accessed on 23 January 2024).
3. Roy, S.; Panaousis, E.; Noakes, C.; Laszka, A.; Panda, S.; Loukas, G. SoK: The MITRE ATT&CK Framework in Research and Practice. *arXiv* **2023**, arXiv:2304.07411.
4. Al-Sada, B.; Sadighian, A.; Oligeri, G. MITRE ATT&CK: State of the Art and Way Forward. *arXiv* **2023**, arXiv:2308.14016.
5. Chen, L.; Ye, Y.; Bourlai, T. Adversarial machine learning in malware detection: Arms race between evasion attack and defense. In Proceedings of the 2017 European Intelligence and Security Informatics Conference (EISIC), Attica, Greece, 11–13 September 2017; pp. 99–106.
6. Li, D.; Li, Q.; Ye, Y.; Xu, S. Arms race in adversarial malware detection: A survey. *ACM Comput. Surv.* **2021**, *55*, 15. [CrossRef]
7. Onwubiko, C. Cocoa: An ontology for cybersecurity operations centre analysis process. In Proceedings of the 2018 International Conference on Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA), Glasgow, UK, 11–12 June 2018; pp. 1–8.
8. Mozzaquatro, B.A.; Agostinho, C.; Goncalves, D.; Martins, J.; Jardim-Goncalves, R. An ontology-based cybersecurity framework for the internet of things. *Sensors* **2018**, *18*, 3053. [CrossRef] [PubMed]
9. Martins, B.F.; Serrano, L.; Reyes, J.F.; Panach, J.I.; Pastor, O.; Rochwerger, B. Conceptual characterization of cybersecurity ontologies. In Proceedings of the IFIP Working Conference on the Practice of Enterprise Modeling, Riga, Latvia, 25–27 November 2020; Springer: Cham, Switzerland, 2020; pp. 323–338.
10. ENISA. *Securing Machine Learning Algorithms*; ENISA: Heraklion, Greece, 2021.
11. *ISO 27000*; Information Technology, Security Techniques, Information Security Management Systems, Overview and Vocabulary. International Organization for Standardization ISO: Geneve, Switzerland, 2009.
12. Tidjon, L.N.; Khomh, F. Threat assessment in machine learning based systems. *arXiv* **2022**, arXiv:2207.00091.
13. Mauri, L.; Damiani, E. Modeling threats to AI-ML systems using STRIDE. *Sensors* **2022**, *22*, 6662. [CrossRef] [PubMed]
14. Jordan, B.; Piazza, R.; Darley, T. *STIX*, version 2.1; OASIS Standard: Burlington, MA, USA, 2021.
15. Glimm, B.; Horrocks, I.; Motik, B.; Stoilos, G.; Wang, Z. HermiT: An OWL 2 reasoner. *J. Autom. Reason.* **2014**, *53*, 245–269. [CrossRef]
16. Kazakov, Y.; Krötzsch, M.; Simančík, F. ELK: A reasoner for OWL EL ontologies. *Syst. Descr.* **2012**. Available online: https://www.uni-ulm.de/fileadmin/website_uni_ulm/iui.inst.090/Publikationen/2012/KazKroSim12ELK_TR.pdf (accessed on 23 January 2024).
17. Musen, M.A. The protégé project: A look back and a look forward. *AI Matters* **2015**, *1*, 4–12. [CrossRef]
18. Ben-Kiki, O.; Evans, C.; Net döt, I. YAML Ain't Markup Language (YAML™) Version 1.2. 2009. Available online: https://yaml.org/spec/1.2/spec.html (accessed on 23 January 2024).
19. Johnson, C.; Badger, M.; Waltermire, D.; Snyder, J.; Skorupka, C. Guide to Cyber Threat Information Sharing. 2016. Available online: https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-150.pdf (accessed on 23 January 2024). [CrossRef]
20. Peroni, S.; Shotton, D. The SPAR Ontologies. In Proceedings of the Semantic Web—ISWC 2018, Monterey, CA, USA, 8–12 October 2018; Vrandečić, D., Bontcheva, K., Suárez-Figueroa, M.C., Presutti, V., Celino, I., Sabou, M., Kaffee, L.A., Simperl, E., Eds.; Springer: Cham, Switzerland, 2018; pp. 119–136.
21. Le, H.; Pham, Q.; Sahoo, D.; Hoi, S.C.H. URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection. *arXiv* **2018**, arXiv:1802.03162.
22. Bozkir, A.S.; Dalgic, F.C.; Aydos, M. GramBeddings: A new neural network for URL based identification of phishing web pages through n-gram embeddings. *Comput. Secur.* **2023**, *124*, 102964. [CrossRef]
23. Li, X.; Li, F. Adversarial Examples Detection in Deep Networks with Convolutional Filter Statistics. *arXiv* **2016**, arXiv:1612.07767.
24. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv* **2015**, arXiv:1412.6572.
25. Carlini, N.; Wagner, D. Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec '17, New York, NY, USA, 30 October–3 November 2017; pp. 3–14. [CrossRef]
26. Cinà, A.E.; Grosse, K.; Demontis, A.; Vascon, S.; Zellinger, W.; Moser, B.A.; Oprea, A.; Biggio, B.; Pelillo, M.; Roli, F. Wild Patterns Reloaded: A Survey of Machine Learning Security against Training Data Poisoning. *ACM Comput. Surv.* **2023**, *55*, 294. [CrossRef]
27. Horridge, M.; Drummond, N.; Goodwin, J.; Rector, A.L.; Stevens, R.; Wang, H. The Manchester OWL syntax. In Proceedings of the OWLed, Athens, GA, USA, 10–11 November 2006; Volume 216.
28. Lam, A.N.; Elvesæter, B.; Martin-Recuerda, F. A Performance Evaluation of OWL 2 DL Reasoners using ORE 2015 and Very Large Bio Ontologies. In Proceedings of the DMKG 2023: 1st International Workshop on Data Management for Knowledge Graphs, Hersonissos, Greece, 29 May 2023.
29. Vakhshiteh, F.; Nickabadi, A.; Ramachandra, R. Adversarial Attacks Against Face Recognition: A Comprehensive Study. *IEEE Access* **2021**, *9*, 92735–92756. [CrossRef]

30. Shan, S.; Wenger, E.; Zhang, J.; Li, H.; Zheng, H.; Zhao, B.Y. Fawkes: Protecting Personal Privacy against Unauthorized Deep Learning Models. *arXiv* **2020**, arXiv:2002.08327.
31. Gómez, A.; Muñoz, A. Deep Learning-Based Attack Detection and Classification in Android Devices. *Electronics* **2023**, *12*, 3253. [CrossRef]
32. Abdelaty, M.; Doriguzzi-Corin, R.; Siracusa, D. DAICS: A deep learning solution for anomaly detection in industrial control systems. *IEEE Trans. Emerg. Top. Comput.* **2021**, *10*, 1117–1129. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.