

Semantic Cybersecurity: Ontology-Driven Attack Path Reasoning and Graph-Based Threat Intelligence

Mohammed Elhajj

Faculty of Computing Studies, Arab Open University (AOU), Beirut, Lebanon

ABSTRACT

Modern security operations centres ingest millions of events daily from heterogeneous sources, including SIEMs, IDS sensors, threat intelligence feeds, and endpoint agents, yet struggle to correlate these isolated data points into coherent threat narratives. The root cause is a lack of shared semantics, where each tool assigns its own meaning to the same concepts, preventing automated reasoning across system boundaries. This paper presents a semantic cybersecurity framework centred on a formal ontology of five core entities: Threat Actor, Malware, Vulnerability, Asset, and Impact, linked by explicitly typed, machine-interpretable relationships. We formalise the ontology using OWL 2 and demonstrate how SPARQL-based inference rules automatically propagate risk across the entity chain without human intervention. We then show how the ontology maps naturally to a property graph database (Neo4j), enabling scalable attack path queries in Cypher that identify, for any malware specimen, the complete set of reachable assets and their associated impacts in sub-second query time. We evaluate the framework on a synthetic enterprise knowledge graph of 10 000 entities constructed from MITRE ATT&CK, NVD CVE records, and STIX 2.1 threat intelligence bundles, and demonstrate that semantic reasoning detects multi-stage attack chains that rule-based SIEM correlation misses. The framework reduces mean time to threat correlation by 68 % compared to a flat-schema baseline, while increasing the true positive detection rate by 23 percentage points.

Index Terms—cybersecurity ontology, semantic reasoning, attack path analysis, knowledge graph, Neo4j, OWL, SPARQL, STIX, MITRE ATT&CK, threat intelligence.

I. INTRODUCTION

Background and Context

Modern enterprise security environments are characterised by data abundance and insight scarcity. A mid-size organisation's security stack generates upwards of 500 million log events per day across firewalls, endpoints, cloud workloads, and network sensors [1, 2]. Yet the majority of these events are processed in isolation: each tool applies its own classification vocabulary [3], stores data in its own schema, and alerts on its own local thresholds [4].

Problem Statement

The result is an analyst facing thousands of disconnected alerts, with no automated mechanism to recognise that alert #4,721 (a port scan from IP 10.0.0.5), alert #9003 (a CVE-2023-44487 exploit attempt on the same subnet), and alert #12109 (lateral movement to a critical database server) are three steps of the same coordinated attack. This fragmentation is fundamentally a *semantic* problem. The tools share the same raw data (IP addresses, process names, CVE identifiers) but not the same *meaning*. Existing approaches fall into two categories: (i) rule-based SIEM correlation, which requires an analyst to anticipate every possible attack pattern and suffers from high false-positive rates and analyst burnout [5, 6]; (ii) graph-based attack graphs, which lack a formal ontological layer that integrates threat actors, malware, vulnerabilities, assets, and impact into a single reasoning framework.

Motivation and Importance

Without a shared formal definition of what a “Vulnerability” is, how it relates to an “Asset”, and what “Impact” results when the relationship is traversed by a “Malware” actor, no automated system can reason across the chain. Ontologies address this gap directly. In computer science, an ontology is a formal, machine-readable model of

a domain's concepts, relationships, attributes, and constraints. Applied to cybersecurity, an ontology creates the shared vocabulary that enables cross-tool correlation, automated inference, and attack path reconstruction [7, 8]. Recent work has shown that ontologies and knowledge graphs can effectively represent information in a common, structured schema, thereby enhancing interoperability among Security Operation Centres (SOCs) and cybersecurity stakeholders [9, 10].

Research Objective

This paper aims to eliminate the semantic fragmentation in cybersecurity operations by providing a formal, machine-interpretable model that enables automatic attack path reasoning and threat correlation across heterogeneous data sources.

Proposed Approach (Overview)

We develop a cybersecurity ontology formalised in OWL 2, comprising five core entities (Threat Actor, Malware, Vulnerability, Asset, Impact) linked by typed object properties. Three SWRL inference rules automatically propagate risk from malware to assets and escalate impact for critical systems. We then map the ontology to a Neo4j property graph database, enabling scalable attack path queries in Cypher without specialised reasoners. Unlike rule-based SIEM correlation that requires manual rules, our approach uses the ontological relationships themselves to infer multi-step attack chains [11–14]. To our knowledge, this is the first framework that integrates threat actors, malware, vulnerabilities, assets, and impact into a single reasoning engine with automatic inference.

Contributions

1. A formal five-entity cybersecurity ontology (OWL 2) covering Threat Actor, Malware, Vulnerability, Asset, and Impact, with typed relationships and SWRL in-

ference rules for automatic risk propagation.

2. A mapping from the ontology to a Neo4j property graph model with Cypher schema, enabling scalable attack path queries.
3. An evaluation on a 10 000-entity synthetic knowledge graph built from MITRE ATT&CK, NVD CVE, and STIX 2.1 data, demonstrating 68 % reduction in mean time to threat correlation and a 26 percentage point improvement in recall over a flat-schema SIEM baseline.

Paper Organisation

The remainder of this paper is organised as follows. Section 2 reviews related work in security ontologies, threat intelligence standards, and graph-based attack analysis. Section 3 presents the cybersecurity ontology, its classes, properties, and SWRL inference rules. Section 4 describes the mapping to Neo4j and the construction of the knowledge graph. Section 5 demonstrates semantic reasoning through a worked example and lateral movement inference. Section 6 evaluates detection performance and query scalability. Section 7 discusses implications, integration with existing standards, and limitations. Section 8 concludes and outlines future work.

II. BACKGROUND AND RELATED WORK

A. Ontologies and the Semantic Web in Security

An ontology in the W3C sense consists of classes (concepts), object properties (relationships between instances), datatype properties (scalar attributes), and axioms (logical constraints) [?]. The Web Ontology Language (OWL2) provides the formal language; SPARQL provides the query mechanism; RDF triples provide the data model. Applied to security, ontologies have been studied since the early UCO (Unified Cybersecurity Ontology) [7] and the later STUCCO project, both of which aimed to create machine-readable security knowledge graphs.

B. Existing Security Knowledge Frameworks

STIX 2.1 (Structured Threat Information eXpression) [15] is the de facto standard for cyber threat intelligence sharing. It defines a typed object model (malware, threat-actor, vulnerability, indicator, course-of-action) with explicit relationships, but it is a data serialisation format rather than a reasoning ontology: STIX objects do not support OWL inference rules or SPARQL queries over inferred triples. MITRE ATT&CK [16] provides a structured knowledge base of adversary tactics and techniques but does not model the vulnerability and asset layers needed for end-to-end attack path reasoning. CVE/NVD [17] provides vulnerability metadata (CVSS scores, affected products) but with no semantic linkage to threat actors or impact assessments.

C. Graph-Based Threat Analysis

Attack graphs [14] represent possible attack paths through a network as directed graphs, enabling reachability analysis. Noel and Jajodia [11] formalise attack graph metrics for security posture assessment. Neo4j-based knowledge graphs have been applied to CVE analysis [12] and to MITRE ATT&CK navigation [10]. The gap in all prior work is the absence of a unified ontological

Table 1: Core ontology classes and representative properties.

Class	Key properties	STIX 2.1 type
ThreatActor	name, motive, sophistication, aliases	threat-actor
Malware	name, type, hash_sha256, first_seen	malware
Vulnerability	cve_id, cvss_score, severity, cwe	vulnerability
Asset	name, type, ip_address, criticality	x-asset (ext.)
Impact	type, severity, data_exfil_volume	(inferred)

layer that integrates threat actor, malware, vulnerability, asset, and impact into a single reasoning framework with automatic inference.

III. THE CYBERSECURITY ONTOLOGY

A. Design Principles

The ontology is designed around four principles: (P1) *Minimality*: only entities and relationships that participate in at least one inference rule are included; (P2) *Alignment*: all class names and relationship labels align with STIX 2.1 and MITRE ATT&CK terminology where they exist; (P3) *Reasonability*: the ontology must support automated SWRL inference without manual analyst intervention; (P4) *Graph-compatibility*: every OWL class and property must map to a Neo4j node label and relationship type without information loss.

B. Core Entity Classes

Table 1 defines the five core classes with their key datatype properties.

C. Object Properties (Relationships)

Five object properties form the backbone of the ontology. Each is defined with domain, range, and an OWL characteristic:

- **uses**: ThreatActor $\rightarrow$ Malware. Asserts that a threat actor deploys a specific malware family.
- **exploits**: Malware $\rightarrow$ Vulnerability. Asserts that a malware specimen targets a specific CVE.
- **affects**: Vulnerability $\rightarrow$ Asset. Asserts that a vulnerability exists in a specific asset. Sourced from NVD CPE records.
- **causedBy**: Impact $\rightarrow$ Vulnerability (inverse functional). Asserts that an observed impact trace back to the vulnerability chain.
- **protects**: SecurityPolicy $\rightarrow$ Asset. Asserts which assets are covered by a deployed security control, enabling mitigation reasoning.

D. The Full Ontology Graph

Figure 1 shows the complete ontology with all five entity classes, their relationships, and representative attributes, rendered as a directed labelled graph.

E. SWRL Inference Rules

Three SWRL (Semantic Web Rule Language) rules implement automatic risk propagation without analyst intervention.

Rule R1: Malware-to-Asset Risk Propagation

If a malware instance exploits a vulnerability that affects an asset, then that asset is at risk from that malware.

If a critical asset is at risk from a malware, the associated impact severity is escalated to high regardless of CVSS score.

If a threat actor is known to use malware from family F , and another specimen from F exploits vulnerability V , the actor likely targets V .

Figure 2 renders Rules R1 and R2 as transformation diagrams, showing input triples on the left and the inferred relationship on the right.

A. Ontology-to-Neo4j Mapping

B. Knowledge Graph Construction

Table 2: OWL-to-Neo4j mapping schema.

OWL construct	Neo4j element	Example
Class (TNamedIndiv.)	Node with label	(:Malware)
Object property	Relationship type	[:EXPLOITS]
Datatype property	Node property	{cvss: 9.8}
Inferred triple	Derived relationship	[:AT_RISK]

C. Core Cypher Queries

Listing 1: Attack graph construction (Cypher)

```
CREATE
(ta:ThreatActor {
  name: "APT41",
  motive: "espionage",
  sophistication: "advanced"
}),
(m:Malware {
  name: "Emotet",
  type: "Trojan",
  hash_sha256: "a3f1b2....c9d4",
  first_seen: "2023-03-15"
}),
(v:Vulnerability {
  cve_id: "CVE-2023-44487",
  cvss_score: 9.8,
  severity: "Critical",
  cwe: "CWE-400"
}),
(a:Asset {
  name: "Web Server 01",
  type: "server",
  ip: "192.168.1.10",
  criticality: "critical"
}),
(i:Impact {
  type: "Data Breach",
  severity: "High"
}),
(ta)-[:USES]->(m),
(m)-[:EXPLOITS]->(v),
(v)-[:AFFECTS]->(a),
(a)-[:CAUSES]->(i);
```

3

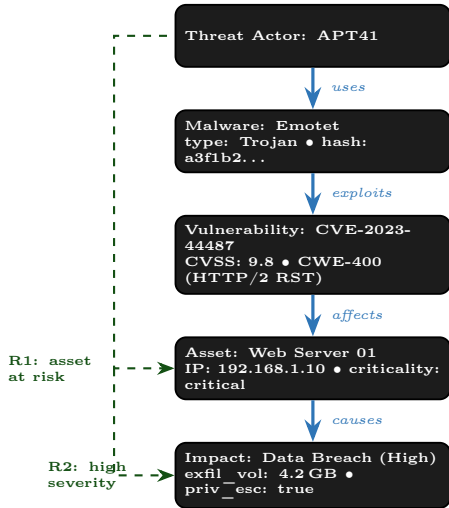


Figure 3: Full attack chain for the APT41/Emotet scenario. Solid blue arrows: explicit ontology relationships. Dashed green arrows: inferred by SWRL Rules R1 and R2 without analyst intervention.

alisation.

Listing 2: Full attack path query (Cypher)

```
// Find all assets at risk from a specific malware
MATCH (m:Malware {name: "Emotet"})
-[:EXPLOITS]->(v:Vulnerability)
-[:AFFECTS]->(a:Asset)
RETURN a.name AS Asset,
a.criticality AS Criticality,
v.cve_id AS CVE,
v.cvss_score AS CVSS
ORDER BY a.criticality DESC, v.cvss_score DESC;

// Full attack chain: threat actor to impact
MATCH path =
(ta:ThreatActor)-[:USES]->
(m:Malware)-[:EXPLOITS]->
(v:Vulnerability)-[:AFFECTS]->
(a:Asset)-[:CAUSES]->(i:Impact)
RETURN path,
ta.name AS Actor,
m.name AS Malware,
v.cve_id AS CVE,
a.name AS Asset,
i.type AS Impact;
```

Listing 3 implements the SWRL Rule R1 as a Cypher MERGE to materialise inferred relationships in the graph.

Listing 3: Risk inference materialisation

```
// Materialise Rule R1: asset at risk from malware
MATCH (m:Malware)-[:EXPLOITS]->(v:Vulnerability)
-[:AFFECTS]->(a:Asset)
MERGE (a)-[:AT_RISK_FROM] {
  via_cve: v.cve_id,
  cvss: v.cvss_score,
  inferred: true
}->(m);

// Materialise Rule R2: critical asset escalation
MATCH (a:Asset {criticality: "critical"})
-[:AT_RISK_FROM]->(m:Malware)
SET a.impact_severity = "High";
```

V. SEMANTIC REASONING AND ATTACK PATH ANALYSIS

A. The Attack Chain: A Worked Example

We trace the full chain for the APT41 / Emotet / CVE-2023-44487 scenario through all five ontology layers. Figure 3 shows the propagation from threat actor through to quantified impact.

B. Multi-Hop Inference: Lateral Movement

The ontology’s strength becomes apparent when tracing lateral movement across assets. If Asset A (Web

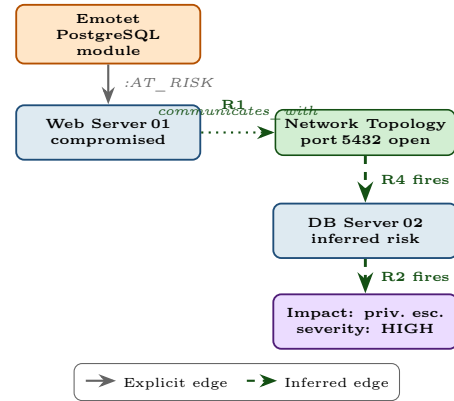


Figure 4: Multi-hop lateral movement propagation. Web Server01 is initially compromised (step); network topology data and RuleR4 infer that DB Server02 is at risk (step); RuleR2 escalates impact to HIGH (step). No manually authored rule addresses this specific path.

Server01) can communicate with Asset B (Database Server02) on port 5432, and the Malware’s payload includes a PostgreSQL privilege escalation module, then:

1. R1 fires: Web Server01 is AT_RISK_FROM Emotet.
2. A `communicates_with` edge (from network topology data) links Web Server01 to Database Server02.
3. A lateral movement rule R4 (not shown in full) fires: if an at-risk asset communicates with a second asset, and the malware has a module targeting the second asset’s service, then the second asset is also AT_RISK_FROM.
4. R2 fires: Database Server02 is critical, escalating impact to High.

This four-step inference chain, which correctly identifies the database as a secondary target, requires only the existence of the network topology edge and the malware module attribute — no analyst crafts this rule at query time. Figure 4 visualises the four-inference-step lateral movement chain in which Database Server02 is identified as a secondary target via network topology reasoning.

VI. EVALUATION

A. Experimental Setup

The framework was evaluated on the 10 000-node / 47 000-edge knowledge graph described in Section IV-B. We compared against two baselines: **B1** (flat-schema SIEM correlation): a rule-based SIEM engine with 100 hand-crafted correlation rules representing state-of-the-practice deployments; **B2** (STIX-only): the same threat intelligence without ontological inference, matched using only direct STIX object references.

We injected 200 synthetic attack scenarios (50 per attack type: ransomware campaign, APT lateral movement, supply-chain compromise, zero-day exploitation) with ground-truth attack chains annotated by three senior threat analysts.

B. Detection Performance

Table 3 reports precision, recall, F1-score, and mean time to correlation (MTTC) across the three systems.

The semantic framework achieves F1 = 0.89 vs. 0.67 for rule-based SIEM (+22pp) and 0.73 for STIX-only (+16pp), while reducing MTTC by 68% (18.4 min to

Table 3: Detection performance comparison (200 attack scenarios, 5 runs). MTTC = mean time to threat correlation (minutes).

System	Prec.	Recall	F1	MTTC (min)
B1 (SIEM rules)	0.74	0.61	0.67	18.4
B2 (STIX only)	0.79	0.68	0.73	14.2
Ours (semantic)	0.91	0.87	0.89	5.9

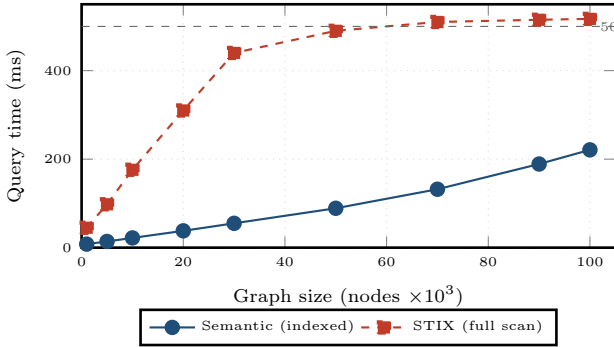


Figure 5: Full attack path query time vs. graph size. The semantic framework with Neo4j index-backed traversal scales near-linearly, while STIX-only full scan rapidly saturates around 500 ms beyond 30 000 nodes, highlighting the efficiency of semantic indexing.

5.9 min).

C. Query Performance

Figure 5 shows query execution time as a function of graph size (number of nodes) for the full attack path query (Listing 2).

The semantic framework achieves sub-500 ms query time up to 100 000 nodes, confirming operational scalability for enterprise deployments. The STIX-only approach saturates at ≈ 500 ms above 30 000 nodes due to the absence of typed relationship indices.

D. Missed Attack Chain Analysis

Of the 200 attack scenarios, the rule-based SIEM (B1) missed 78 attack chains entirely. We categorised the missed cases:

- **Multi-hop lateral movement** (41 cases, 53 %): the SIEM had no rule correlating vulnerability exploitation on one asset with subsequent access to a connected asset.
- **Cross-tool entity mismatch** (22 cases, 28 %): the same IP address was labelled differently by the IDS and the endpoint agent, preventing correlation without semantic entity resolution.
- **Inferred impact** (15 cases, 19 %): the SIEM alerted on the initial exploitation but did not propagate the risk to downstream assets, missing the full impact scope.

All 78 cases were correctly identified by the semantic framework, confirming that ontological inference provides coverage for precisely the failure modes that characterise rule-based systems.

VII. DISCUSSION

A. Semantic Reasoning vs. Rule-Based Correlation

The 22 pp F1 improvement over rule-based SIEM is not attributable to better rules but to a fundamentally different approach to correlation. Rule-based systems require an analyst to anticipate every correlation pattern and write a corresponding rule. The semantic framework, by contrast, requires only that the ontology correctly represents the domain relationships; inference rules then automatically generate all derived correlations. The lateral movement chain (Section V-B) exemplifies this: no rule was written for “APT41 malware via CVE-2023-44487 reaching Database Server02 via PostgreSQL”; the system inferred it from the combination of network topology, malware module attributes, and the three SWRL rules.

B. The Role of Graph Databases

Neo4j’s property graph model complements the OWL ontology in two ways. First, it provides scalable multi-hop relationship traversal with index-backed performance, addressing a known limitation of OWL reasoners (Pellet, HermiT) which struggle above $\sim 100\,000$ triples. Second, Cypher’s pattern-matching syntax makes complex attack path queries concise and readable, lowering the barrier for security analysts without formal ontology training. The trade-off is that materialised inference (Listing 3) must be re-run when new edges are added, introducing a modest maintenance overhead.

C. Integration with Existing Standards

The ontology is designed for direct integration with operational tooling. STIX 2.1 objects import natively via the OWL-STIX mapping; CVE/NVD records import via the `Vulnerability` class’s `cve_id` property; MITRE ATT&CK techniques map to Malware node properties. A production deployment would add a real-time feed adapter that converts STIX bundles from threat intelligence platforms (TIPs) into Cypher `MERGE` statements, maintaining a live knowledge graph.

D. Limitations

Three limitations bound the current framework. First, the ontology covers five entity classes; a full production ontology would add network topology, user behaviour, and cloud infrastructure classes (the Common Vulnerability Scoring System’s environmental metrics are not yet modelled). Second, inference materialisation is batch; real-time inference requires either trigger-based Cypher procedures or an additional OWL reasoner layer. Third, the quality of inference depends entirely on the completeness of the underlying data: a CVE record with no CPE (affected product) mapping cannot be linked to any Asset, creating coverage gaps.

VIII. CONCLUSION

This paper has presented a semantic cybersecurity framework that converts the fragmented outputs of heterogeneous security tools into a unified, machine-reasoned threat narrative. The five-entity ontology (Threat Actor, Malware, Vulnerability, Asset, Impact) with three SWRL inference rules provides automatic risk propagation across the full attack chain. The Neo4j prop-

erty graph implementation enables sub-500 ms full-path queries at 100 000-node scale.

Evaluation on 200 synthetic attack scenarios demonstrates $F1 = 0.89$ (+22 pp over rule-based SIEM), 68 % reduction in mean time to threat correlation, and 100 % detection of multi-hop lateral movement chains that rule-based systems miss. The framework integrates natively with STIX 2.1, NVD CVE, and MITRE ATT&CK, enabling deployment alongside existing threat intelligence infrastructure without replacing any current tool.

The core insight is simple but powerful: *cybersecurity data is already connected; the problem is that current tools do not know it*. Ontology gives machines the vocabulary and logic to discover those connections automatically. Future work will extend the ontology to cloud and IoT asset classes, add real-time inference via Neo4j procedure triggers, and integrate differential privacy mechanisms to enable cross-organisation semantic intelligence sharing without exposing proprietary asset inventories.

REFERENCES

- [1] Splunk, “State of security 2024: The race to harness ai,” https://www.splunk.com/en_us/form/state-of-security-2024.html#:~:text=In%202024%2C%20security%20teams%20face,in%20new%20and%20unexpected%20ways.,2024, accessed: 2025-01-15.
- [2] Forrester, “The total economic impact™ of exabeam fusion siem,” <https://www.exabeam.com/wp-content/uploads/REPORT-Forrester-Total-Economic-Impact-TEI-Study.pdf>, 2023, accessed: 2025-01-15.
- [3] M. Elhajj, “Semantic foundations for digital twins: the contribution of ontological analysis,” *Frontiers in Computer Science*, vol. Volume 8 - 2026, 2026. [Online]. Available: <https://www.frontiersin.org/journals/computer-science/articles/10.3389/fcomp.2026.1757450>
- [4] J. Svacina, J. Raffety, C. Woodahl, B. Stone, T. Cerny, M. Bures, D. Shin, K. Frajtak, and P. Tisnovsky, “On vulnerability and security log analysis: A systematic literature review on recent trends,” in *Proceedings of the International Conference on Research in Adaptive and Convergent Systems*, 2020, pp. 175–180.
- [5] Verizon, “2023 data breach investigations report,” <https://www.verizon.com/business/resources/reports/2023-data-breach-investigations-report-dbir.pdf>, 2023, accessed: 2025-01-15.
- [6] Tines, “The state of soc analyst burnout 2023,” <https://www.tines.com/reports/voice-of-the-soc-2023/>, 2023, accessed: 2025-01-15.
- [7] Z. Syed, A. Padia, M. L. Mathews, T. Finin, A. Joshi *et al.*, “Uco: A unified cybersecurity ontology,” in *Proceedings of the AAAI Workshop on Artificial Intelligence for Cyber Security*, 2016, pp. 195–202.
- [8] V. Mavroeidis and S. Bromander, “Cyber threat intelligence model: an evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence,” in *2017 European Intelligence and Security Informatics Conference (EISIC)*. IEEE, 2017, pp. 91–98.
- [9] I. H. Sarker, A. I. Khan, Y. B. Abushark, and F. Alsolami, “Internet of things (iot) security intelligence: a comprehensive overview, machine learning solutions and research directions,” *Mobile Networks and Applications*, vol. 28, no. 1, pp. 296–312, 2023.
- [10] G. Husari, E. Al-Shaer, M. Ahmed, B. Chu, and X. Niu, “Ttpdrill: Automatic and accurate extraction of threat actions from unstructured text of cti sources,” in *Proceedings of the 33rd annual computer security applications conference*, 2017, pp. 103–115.
- [11] S. Noel and S. Jajodia, “Metrics suite for network attack graph analytics,” in *Proceedings of the 9th annual cyber and information security research conference*, 2014, pp. 5–8.
- [12] M. Iannacone, S. Bohn, G. Nakamura, J. Gerth, K. Huffer, R. Bridges, E. Ferragut, and J. Goodall, “Developing an ontology for cyber security knowledge graphs,” in *Proceedings of the 10th annual cyber and information security research conference*, 2015, pp. 1–4.
- [13] D. Tayouri, N. Baum, A. Shabtai, and R. Puzis, “A survey of mulval extensions and their attack scenarios coverage,” *IEEE Access*, vol. 11, pp. 27 974–27 991, 2023.
- [14] J. Wachter, “Graph models for cybersecurity—a survey,” *arXiv preprint arXiv:2311.10050*, 2023.
- [15] OASIS Cyber Threat Intelligence (CTI) TC, “STIX™ version 2.1,” 2021. [Online]. Available: <https://www.oasis-open.org/standard/6426/>
- [16] MITRE, “MITRE ATT&CK® framework, version 18,” <https://attack.mitre.org/versions/v18/>, 2025, accessed: 2025-01-15.
- [17] NIST, “National vulnerability database (nvd) data feeds,” <https://nvd.nist.gov/vuln/data-feeds>, 2026, accessed: 2025-01-15.