ATT&CK to CVE: A Large-Scale Automated Knowledge Graph for Threat Intelligence

Moqsadur Rahman*, Monika Akbar*, Joseph T. Aguayo†, Mahmud Shahriar Hossain*, Tina M. Ellis†, Siddhartha Shankar Das†, Mohammad F. Babar‡, Monowar Hasan‡, Aaron Sanchez‡, Luis Fernando de la Torre‡, Mahantesh Halappanavar†

*The University of Texas at El Paso, USA
{mrahman20, makbar, mhossain}@utep.edu

†Pacific Northwest National Laboratory, USA
{joseph.aguayo, tina.ellis, siddhartha.das, hala}@pnnl.gov

‡Washington State University, USA
{m.babar, monowar.hasan, aaron.m.sanchez, luis.delatorre}@wsu.edu

***Abstract*—Cyber threat intelligence (CTI) includes collecting and analyzing cybersecurity-related information across diverse and heterogeneous sources. Those sources may include repositories that curate vulnerabilities, weaknesses, attack pattern related data in different formats with varying degrees of detail. Analysts must continuously reconcile these sources to gain a coherent view of the evolving threat landscape, yet this process is often manual, incomplete, and error-prone. In this work, we present a fully automated cybersecurity knowledge graph pipeline that systematically ingests and normalizes data from the National Vulnerability Database (CVE and CVSS), MITRE CWE, CAPEC, and ATT&CK frameworks, and integrates them into a Neo4j graph database. Our system extracts a rich set of properties and inter- and intra-entity relationships from each dataset, including hierarchical links, cross-domain mappings, and temporal metadata, while ensuring daily synchronization with upstream feeds. The resulting knowledge graph contains hundreds of thousands of interconnected entities and edges, enabling multi-hop analysis in various directions (e.g., from attack techniques to vulnerabilities, from vulnerabilities to attack techniques). We demonstrate the usefulness of the knowledge graph for comprehensive weakness analysis, linking adversary groups to exploited vulnerabilities, and graph-based inference tasks that include variable-length path traversal and link discovery. Our case studies on CISA advisories show that hierarchical graph traversals uncover CVEs beyond those explicitly listed, bridging critical gaps in threat intelligence. By combining automation, completeness, and semantic richness, our knowledge graph provides a scalable, continuously updated foundation for cyber defense analytics, supporting both operational decision-making and advanced research in graph learning and reasoning over CTI.**

***Index Terms*—Cyber Threat Intelligence, Knowledge Graph, Graph Databases, CWE, CVE, ATT&CK, CAPEC**

I. Introduction

The modern cyber threat landscape is vast, dynamic, and fragmented across multiple authoritative but heterogeneous sources of intelligence. The National Vulnerability Database (NVD) [1] provides detailed records of disclosed Common Vulnerabilities and Exposures (CVEs [2]) and their severity scores, while MITRE [3] curates complementary catalogs: Common Weakness Enumeration (CWE [4]), which lists hardware and software security weakness classes; Common Attack Pattern Enumeration and Classification (CAPEC [5]), which describes adversary attack patterns; and Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK [6]), which captures tactics, techniques, and procedures (TTPs) observed in cyberattacks. Each of these resources is indispensable in its own domain, yet their siloed formats, differing schemas, and inconsistent cross-references present significant hurdles for analysts who need an integrated and holistic view that interconnects vulnerabilities, weaknesses, attack patterns, and adversarial behaviors.

Existing efforts to unify CTI often focus on one or two sources, rely on static or manually curated mappings, or lack mechanisms to keep pace with daily updates. As a result, organizations are left with incomplete or outdated views of how newly published vulnerabilities may be exploited, which weaknesses cause recurring attacks, or how defensive measures align with adversary techniques. Without automation and the full-spectrum integration of diverse security-related data sources, the potential for utilizing CTI in proactive defense and explainable analytics remains significantly limited.

In this paper, we introduce a fully automated cybersecurity knowledge graph that bridges these gaps. Our system develops dedicated ingestion pipelines for CVE, CWE, CAPEC, and ATT&CK, handling their native formats (e.g., JSON, XML, STIX bundles), normalizing properties into a unified schema, and encoding explicit relationships among entities in a Neo4j graph database (detailed in §III). This automation ensures that knowledge is refreshed regularly, requiring no manual intervention while continuously incorporating new information. The resulting knowledge graph is both comprehensive and expressive: it captures hundreds of thousands of nodes representing vulnerabilities, weaknesses, attack patterns, techniques, malware, tools, intrusion sets, mitigations, etc. The entities are linked through rich, typed relationships. This enables analysts to run multi-hop queries that explain, for example, how a specific CVE maps to underlying weaknesses, which

CAPEC attack patterns exploit them, and which ATT&CK techniques adversaries have used in real-world operations.

Beyond integration, the graph unlocks advanced analytics. Analysts can identify software weaknesses most frequently linked to techniques, perform shortest-path traversals across heterogeneous entities, or enrich vulnerabilities with contextual data such as Common Platform Enumerations (CPEs) [7], mitigations, and Common Vulnerability Scoring System (CVSS) [8] severity. Moreover, through graph-based topology analysis, the system anticipates potential future or missing connections between entities. In doing so, it elevates CTI from disparate feeds into a continuously updated, queryable, and comprehensible foundation for proactive defense, threat hunting, and cyber risk assessment (detailed in §IV).

The key contributions of this work are as follows.

- **End-to-end automation:** We develop a fully automated pipeline that ingests CVE, CWE, CAPEC, and ATT&CK data sources, enabling bidirectional reasoning between vulnerabilities and adversary techniques.
- **Rich extraction of nodes and relationships:** We construct a large-scale knowledge graph that captures extensive nodes, relationships, and properties, ensuring semantic richness and comprehensive cross-domain coverage.
- **Comprehensive ATT&CK framework representation:** We provide a fine-grained representation of all ATT&CK entities, including techniques, tactics, mitigations, and related objects, and thus enabling a structured exploration of adversary behaviors.
- **Topology-based link discovery:** We introduce topology-based methods that reveal plausible but unobserved relationships in the graph, demonstrating how structural similarity can uncover meaningful connections between weaknesses and attack patterns.
- **Neo4j-based querying and analytics:** We deploy the graph in Neo4j and leverage the expressive `Cypher` query language to support advanced analytics, including multi-hop reasoning, shortest-path discovery, and pattern analysis.

II. Related Work

Cybersecurity knowledge graphs (KGs) have emerged as a powerful paradigm for integrating heterogeneous threat intelligence, enabling advanced reasoning and decision support. Several efforts have focused on constructing KGs by linking existing security standards and repositories such as CVE, CWE, CAPEC, ATT&CK, and CPE. One of the earliest systematic efforts was the SEPSES Cybersecurity Knowledge Graph, which provided vocabularies and a continuously updated KG integrating sources such as CVE, CWE, and CAPEC [9]. Similarly, the CyberGraph framework demonstrated the automatic construction of a Neo4j-based KG unifying CVE, CWE, CAPEC, and CPE, allowing for advanced graph querying and visualization [10]. However, these works primarily focus on structural integration of a limited subset of sources and do not scale to large, enriched graphs with extensive properties.

Beyond structural integration, several works have leveraged KGs for reasoning and predictive tasks. Shi *et al.* developed a threat knowledge graph linking CVE, CWE, and CPE, and applied KG completion techniques to uncover previously unknown associations, evaluated using ranking metrics such as MRR and Hits@N [11]. Ampel *et al.* proposed CVET, a self-distillation model that maps CVEs to MITRE ATT&CK tactics, thus bridging vulnerability-centric data with adversary behaviors [12]. Das *et al.* introduced V2W-BERT, a Transformer-based hierarchical multi-class classification framework for automated mapping of CVEs to CWEs, achieving state-of-the-art performance even for rare classes [13]. While these approaches introduce reasoning capabilities, they often depend on small-scale datasets and lack an end-to-end automated pipeline that continuously incorporates updates from all relevant threat intelligence repositories.

In parallel, large-scale multi-source integration has also been studied. Hemberg *et al.* introduced the BRON knowledge graph, which links ATT&CK, CWE, CAPEC, and CVE, supporting bidirectional traversal across tactics, weaknesses, and vulnerabilities for cyber threat hunting applications [14]. Shen *et al.* explored a similar approach in the industrial control systems (ICS) domain, using data-driven extraction and graph construction techniques to model ICS-specific vulnerabilities and threats [15]. These works achieve broader coverage, but remain domain-specific, rely on static integration, and do not incorporate enriched property sets needed for fine-grained reasoning and explainability.

Beyond vulnerability-centric graphs, countermeasure-oriented KGs have also been proposed. The D3FEND framework, developed by MITRE, provides a semantically rigorous representation of defensive techniques and their mapping to ATT&CK TTPs, thereby supporting explainable reasoning about mitigation strategies [16]. Recently, retrieval-augmented generation (RAG) approaches such as CyKG-RAG have integrated KGs with language models, leveraging structured cybersecurity knowledge for improved reasoning over unstructured CTI reports [17], [18]. These initiatives focus either on defensive knowledge or integration with language models, but they do not provide a large-scale, unified, and automated knowledge graph that captures both vulnerabilities and attack techniques in an explainable manner.

In contrast to the above works, our approach advances the state of the art by delivering a fully automated, end-to-end pipeline for constructing large-scale cybersecurity knowledge graphs. Our KG integrates CVE, CWE, CAPEC, and ATT&CK while scaling to more than 300K nodes and 250K relationships with 86 property types, offering a semantically enriched representation that supports advanced threat intelligence. We deploy the KG in Neo4j, enabling powerful query capabilities, link discovery based on graph topology, and adversarial activity analytics, thereby addressing both the scalability and analytical gaps identified in prior research.

III. Methodology

Our methodology automates the collection and integration of heterogeneous cyber threat intelligence from CVE, CWE, CAPEC, and MITRE ATT&CK. Through cross-source normalization and automated upsertion into Neo4j, the system produces a unified, continuously updated knowledge graph that preserves semantic links across vulnerabilities, weaknesses, attack patterns, and techniques. This enables efficient querying and analysis without manual curation (see Fig. 1).

A. Automation Process

To ensure that the cybersecurity knowledge graph remains comprehensive, up to date, and consistent across multiple threat intelligence sources, we developed an automated multi-source data acquisition and integration framework. This framework consists of dedicated pipelines for four key datasets—CVE, CWE, CAPEC, and MITRE ATT&CK—that systematically retrieve the latest releases from their respective authoritative sources, process and normalize the information, and ingest it into a Neo4j graph database. Each pipeline is tailored to handle the specific data formats and update mechanisms of its source, enabling seamless integration while preserving data fidelity. A lightweight scheduler orchestrates the execution of these pipelines on a fixed daily cycle, ensuring that the graph continuously reflects the most recent threat intelligence without the need for manual intervention.

The automation framework executes on a fixed 24-hour cycle with integrated logging and error handling. On a weekly basis, it updates the MITRE ATT&CK dataset, while daily runs refresh CAPEC, CWE, and CVE (with a full CVE synchronization on the first pass, followed by incremental updates). ATT&CK is updated weekly since its framework evolves more slowly, with new techniques, groups, and mitigations published less frequently, whereas the other datasets either exhibit frequent changes or are lightweight to process–particularly CVE, which receives daily vulnerability disclosures. The update order follows the dependency chain—ATT&CK→CAPEC→CWE→CVE—since relationships originate from CAPEC to ATT&CK, from CWE to CAPEC, and from CVE to CWE. This design ensures that upstream entities are processed before downstream links are established, keeping the knowledge graph continuously synchronized without manual intervention. The following narrative describes the dataset-specific pipelines:

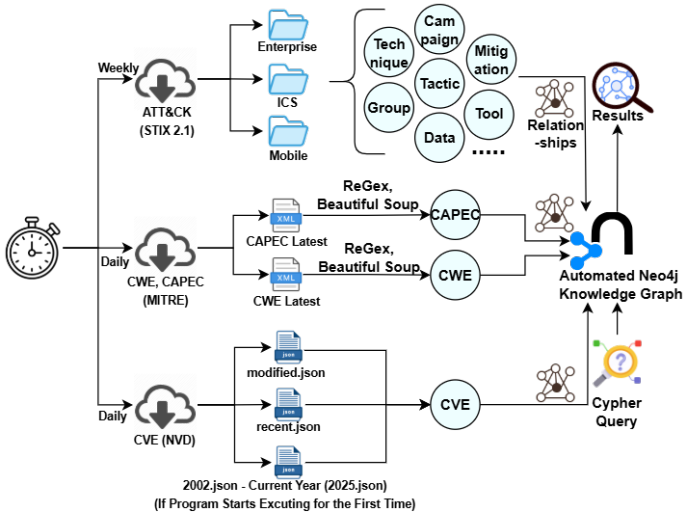

Fig. 1. End-to-end pipeline for constructing an automated cybersecurity knowledge graph.

Common Vulnerabilities and Exposures (CVE) system is a globally recognized catalog of disclosed software and hardware vulnerabilities. Our framework retrieves CVE data from the NVD CVE 2.0 JSON feeds, initially ingesting all yearly archives from 2002 through 2025. After this baseline, the system continuously synchronizes with the "recent" and "modified" datasets, incorporating new or updated vulnerabilities without reprocessing history. Data is downloaded, extracted, normalized for schema consistency, and upserted into Neo4j, ensuring the graph always reflects the latest disclosures.

Common Weakness Enumeration (CWE) catalog, curated by MITRE, defines recurring software and hardware weakness types. Unlike CVE's instance-level vulnerabilities, CWE abstracts structural flaws such as buffer overflows or input validation errors. Our system downloads the CWE XML release, parses all `Weakness` entries, captures metadata, descriptions, and cross-references, normalizes them, and ingests the results into Neo4j as an updated graph representation of weaknesses.

Common Attack Pattern Enumeration and Classification (CAPEC) catalog describes adversarial attack patterns that exploit weaknesses. Our framework downloads the CAPEC archive, parses `Attack_Pattern` entries, and extracts metadata, descriptions, and hierarchies. The processed data is upserted into Neo4j, producing a continuously updated attack-pattern entity linked directly to CWE weaknesses.

MITRE ATT&CK framework captures tactics, techniques, and procedures (TTPs) across the *Enterprise*, *Mobile*, and *ICS* domains. Our system retrieves ATT&CK STIX bundles, extracts entities such as techniques, tactics, intrusion sets, tools, courses of action, etc. and processes their relationship objects. Data is normalized and ingested into Neo4j with typed edges from STIX references, producing a refreshed knowledge layer that spans the full ATT&CK ecosystem and supports graph-based analytics.

These pipelines ensure that the cybersecurity knowledge graph remains comprehensive, consistent, and continuously synchronized with authoritative sources. By automating data ingestion, normalization, and integration across CVE, CWE, CAPEC, and ATT&CK, the framework eliminates manual curation and provides a reliable foundation for downstream analytics and reasoning tasks.

B. Nodes, Properties, and Relationships

This section provides a detailed overview of the properties collected from each source, the relationships encoded between them, and the overall distribution of nodes and edges within the constructed knowledge graph (Fig. 2).

CVE (NVD): The CVE pipeline collects the following

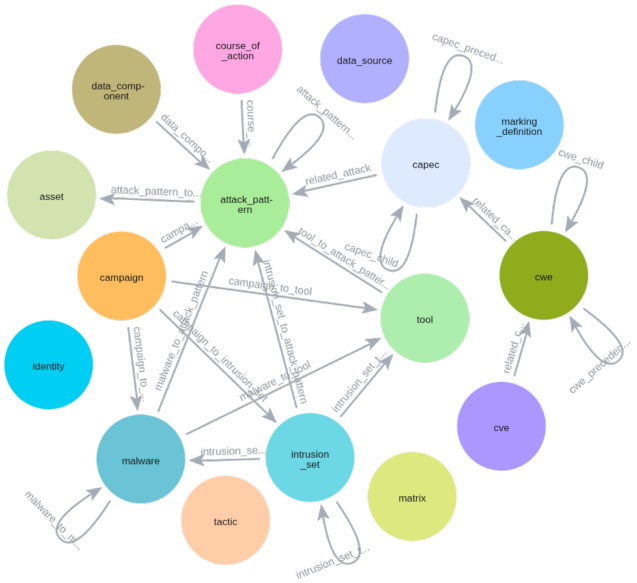

Fig. 2. Schema visualization of node labels and relationship types in the Neo4j cybersecurity knowledge graph.

properties: the vulnerability identifier (`id`), mapped CWE identifiers, a textual description, CVSS metrics (including vector string, base score, and severity — where CVSS, the *Common Vulnerability Scoring System*, is an industry-standard method for rating the severity of software vulnerabilities), exploitability and impact scores, and extracted CPE (*Common Platform Enumeration*) criteria, which specify the affected hardware, operating systems, or software products.

Relationships created: The framework establishes relationships that link CVE to its corresponding CWE (`CVE` → `CWE`).

CWE (MITRE): The CWE extraction process gathers identifiers and metadata (ID, name, abstraction level, structural type, and status), rich text fields (description, extended description, background), and a wide range of structured or nested elements, including alternative terms, applicable platforms, modes of introduction (describing when and how the weakness can be introduced during development), likelihood of exploit, weakness ordinalities (sequence in which weaknesses might occur), common consequences, detection methods, potential mitigations etc.

Relationships created: The pipeline forms hierarchical parent–child links between CWEs (`CWE` → `CWE`), precedence links that indicate which weaknesses can occur before others, and cross-references connecting CWEs to related CAPEC attack patterns (`CWE` → `CAPEC`).

CAPEC (MITRE): For CAPEC data, the collected properties include identifiers and metadata (ID, name, abstraction, status), description, likelihood, severity, mid-level structured sections such as mitigations, prerequisites, indicators, required skills and resources, example instances, notes, consequences, execution flow (sequence of attacker actions), and content history.

Relationships created: We establish hierarchical parent–child links between CAPEC entries (`CAPEC` → `CAPEC`), precedence links, and mappings from CAPEC attack patterns to ATT&CK techniques (`CAPEC` → `ATT&CK`).

ATT&CK (Enterprise/Mobile/ICS STIX): Across all supported ATT&CK domains, the pipeline collects common properties such as ID, external ID, object type, name, description, external references, timestamps/versioning, and domains. Depending on the object type, additional fields may include platforms (target operating systems or environments), aliases (alternative names), detection guidance etc. Extracted nodes include `attack-pattern`, `campaign`, `course-of-action`, `intrusion-set`, `malware`, `tool`, `x-mitre-asset`, `x-mitre-data-source`, `x-mitre-tactic`, `x-mitre-matrix`, etc.

Relationships created: In our system, STIX-defined relationship objects are materialized as graph edges by leveraging the `relationship_type`, `source_ref`, and `target_ref` fields, or, when APOC is unavailable, through a fallback naming scheme of the form `sourceType_to_targetType`.

As shown in Table I, the largest node set in the constructed knowledge graph comes from the CVE dataset, with 307,142 vulnerability entries. This is followed by 1,106 ATT&CK technique (STIX attack_pattern) nodes from the ATT&CK framework. The third-largest group consists of 968 CWE nodes, reflecting common software and hardware weaknesses, while the fourth is 795 malware nodes, also extracted from ATT&CK. The 615 CAPEC nodes rank fifth, capturing common adversarial attack patterns.

TABLE I
NODE TYPES AND THEIR CORRESPONDING ENTITY COUNTS IN THE NEO4J KNOWLEDGE GRAPH, SORTED FROM HIGHEST TO LOWEST COUNT.

Node Type	Entity Count	Node Type	Entity Count
cve	307,142	tool	93
attack_pattern	1,106	campaign	51
cwe	968	data_source	42
malware	795	tactic	40
capec	615	asset	14
course_of_action	334	matrix	4
intrusion_set	185	identity	1
data_component	122	marking_definition	1

The knowledge graph encodes a rich set of relationships across entities, capturing both structural and behavioral connections. The largest is the `related_cwe` relation (CVE→CWE) with 227,981 edges, followed by `malware_to_attack_pattern` (10,694 edges), `intrusion_set_to_attack_pattern` (4,095 edges), `data_component_to_attack_pattern` (2,522 edges), `course_of_action_to_attack_pattern` (1,915 edges), `related_capec` (CWE→CAPEC, 1,212 edges), `cwe_child` (1,151 edges), etc.

Together, these interconnected entities and their rich set of relationships form a comprehensive, multi-perspective knowledge graph that represents vulnerabilities, weaknesses, attack patterns, malware, and adversarial behaviors across multiple MITRE knowledge sources. By capturing hierarchical, se-

quential, and cross-domain associations among nodes such as courses of action, intrusion sets, tools, tactics, and campaigns, the graph enables unified and in-depth analysis of the complex landscape of cybersecurity threats and defenses.

C. Graph Querying and Multi-Hop Traversal

One of the key advantages of representing cyber threat intelligence as a knowledge graph is the ability to perform *multi-hop traversal* across heterogeneous entities. Unlike traditional tabular or siloed repositories, the graph structure allows analysts to traverse explicit paths that connect vulnerabilities (CVEs) to weaknesses (CWEs), to adversarial attack patterns (CAPEC), and ultimately to real-world adversary techniques in ATT&CK. These traversals not only provide the intermediate steps involved in analyzing a threat, but also enhance individual data points (e.g., a CVE) by connecting them to a broader threat context—such as severity, affected platforms, available mitigations, and associated adversarial techniques.

For instance, consider a forward traversal that begins with the vulnerability CVE-2016-6225, which affects Percona Xtra-Backup and stems from improper handling of the initialization vector (IV) in its xbcrypt encryption utility. This flaw, which allows chosen-plaintext attacks against encrypted backup files, is associated with the weakness CWE-326 (Inadequate Encryption Strength). This weakness connects to the attack pattern CAPEC-112 (Brute Force), which in turn maps to the ATT&CK technique T1110 (Brute Force). This traversal illustrates how a specific cryptographic vulnerability can be situated within a broader chain of adversarial behavior, revealing plausible exploitation methods and aligning low-level technical flaws with high-level attacker tactics. The corresponding Cypher query for the forward path is shown in Fig. 3 (left), and its graphical representation in Fig. 4.

Conversely, a reverse traversal starting from the ATT&CK technique T1110 (Brute Force) uncovers associated upstream weaknesses and vulnerabilities that could facilitate brute-force exploitation. Specifically, the knowledge graph links this technique to the attack pattern CAPEC-112, which maps to multiple underlying weaknesses: CWE-326 (Inadequate Encryption Strength), CWE-330 (Use of Insufficiently Random Values), and CWE-521 (Weak Password Requirements). These weaknesses are in turn linked to various CVEs, enabling defenders to identify and prioritize vulnerable systems likely to be targeted using brute-force methods. The Cypher query for this reverse traversal is shown in Fig. 3 (right), and the corresponding path visualization is provided in Fig. 5.

Together, these forward and reverse queries and their corresponding visualizations demonstrate the bidirectional multi-hop traversal capabilities of knowledge graphs—supporting both top-down and bottom-up threat analysis across the cyber kill chain.

D. Topology-Based Link Scoring in Cybersecurity Knowledge Graphs

One of the most valuable analytical capabilities of our integrated cybersecurity knowledge graph is the ability to compute

```
MATCH path =
(c:cve {id:
"CVE-2016-6225"})
-[:related_cwe]->
(w:cwe)
-[:related_capec]->
(p:capec)
-[:related_attack]->
(a:attack_pattern)
RETURN a.external_id AS
attack_id, path
ORDER BY a.external_id
LIMIT 100;
```

```
MATCH path =
(a:attack_pattern
{external_id: "T1110"})
<-[:related_attack]-
(p:capec)
<-[:related_capec]-
(w:cwe)
<-[:related_cwe]-
(c:cve)
RETURN c.id AS
cve_id, path
ORDER BY c.id
LIMIT 100;
```

Fig. 3. Cypher queries for traversing the knowledge graph. Left: forward traversal from CVE→CWE→CAPEC→ATT&CK. Right: reverse traversal from ATT&CK back to CVE.

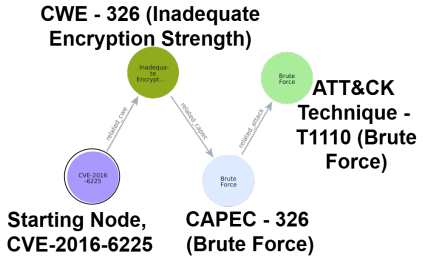

Fig. 4. Path visualization for CVE ("CVE-2016-6225") to ATT&CK.

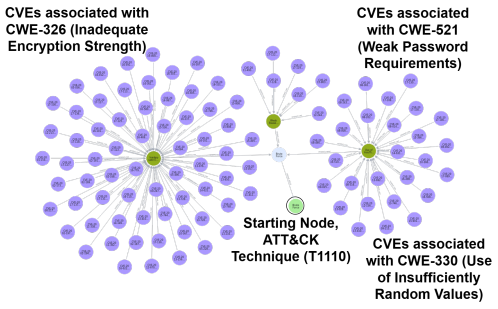

Fig. 5. Path visualization for ATT&CK ("T1110") to CVEs.

topology-based link scores. Here, only the graph's structural properties (topology) are leveraged to quantify how strongly two entities are related, without requiring external features. For example, such scoring allows analysts to identify plausible connections between entities in a graph, such as CVE↔CWE, CWE↔CAPEC, or CAPEC↔ATT&CK techniques.

The Neo4j Graph Data Science (GDS) library provides several well-established structural similarity algorithms [19]. We highlight two representative measures below (with Cypher examples shown in Fig. 6):

- **Adamic–Adar (AA):** Assigns higher weight to shared neighbors that are themselves rare:

$$AA(u,v) = \sum_{w \in \Gamma(u) \cap \Gamma(v)} \frac{1}{\log |\Gamma(w)|}.$$

- **Common Neighbors (CN):** Simply counts the number of shared neighbors:

$$CN(u,v) = |\Gamma(u) \cap \Gamma(v)|.$$

```
// Adamic Adar
MATCH (c1:cwe{id: "308"})
MATCH (c2:capec{id: "151"})
RETURN gds.alpha.linkprediction.adamicAdar(c1, c2) AS
score

// Common Neighbors
MATCH (c1:cwe{id: "308"})
MATCH (c2:capec{id: "151"})
RETURN gds.alpha.linkprediction.commonNeighbors(c1, c2) AS
score
```

Fig. 6. Cypher queries for selected link discovery algorithms between CWE and CAPEC nodes.

Notations: u and v denote two nodes in the knowledge graph, and $\Gamma(u)$ represents the set of neighbors directly connected to node u.

- $|\Gamma(u)|$ is the degree of node u (i.e., the number of its neighbors).
- $\Gamma(u) \cap \Gamma(v)$ is the set of common neighbors of u and v.
- w is an individual node belonging to the set of common neighbors.
- The functions assign similarity scores, where a higher score indicates a greater structural proximity between u and v, and thus a higher likelihood of an unobserved link.

Other metrics such as *Preferential Attachment* (which assumes high-degree nodes are more likely to connect), *Resource Allocation* (a variant of Adamic–Adar that weights neighbors by the inverse of their degree), *Same Community* (binary score based on graph community detection), and *Total Neighbors* (combined neighborhood size) can also be applied to enrich the analysis.

By applying these measures, we can infer *plausible but unobserved relationships*—such as identifying CWEs that may be associated with CAPECs. These insights enable analysts to anticipate potential threat paths before they are explicitly reported in advisories, supporting proactive defense.

IV. Empirical Evaluation

We demonstrate how our knowledge graph enables comprehensive and actionable cyber threat intelligence analysis, supporting end-to-end workflows. We highlight mappings from top software weaknesses to ATT&CK techniques; technique-centric views that connect to groups, tools, and mitigations; and group-to-vulnerability traversals for Scattered Spider (G1015) and Volt Typhoon (G1017). Case studies on CISA advisories and link scoring further demonstrate how the graph discovers missing relationships.

A. Top 25 Software CWEs to ATT&CK Techniques

We leveraged our automated knowledge graph pipeline to connect CWEs with MITRE ATT&CK techniques through CAPEC attack patterns. In particular, we focused on the *CWE Top 25 Most Dangerous Software Weaknesses* list, an annually curated resource maintained by the MITRE Corporation in collaboration with the U.S. Department of Homeland Security (DHS) and the Cybersecurity and Infrastructure Security Agency (CISA). This list highlights the most common and impactful software weaknesses, based on real-world data such as CVEs, CVSS scores, and their prevalence in vulnerability databases. It helps developers and security teams prioritize mitigation efforts by identifying the most frequently exploited programming flaws and guiding secure software design. Using automated graph traversals, we observe that not all 25 CWEs yield direct mappings to ATT&CK techniques.

As an illustrative example, shown in Fig. 7 and Table II, consider CWE-20 (Improper Input Validation). It is linked to multiple ATT&CK techniques such as `T1539: Steal Web Session Cookie` and `T1027: Obfuscated Files or Information`. This mapping underscores the criticality of input validation failures: adversaries routinely exploit unchecked inputs to manipulate control flow, inject malicious payloads, or bypass security mechanisms, making this CWE foundational to a wide range of attacks. Similarly, CWE-94 (Code Injection) maps directly to `T1027.006: HTML Smuggling`, highlighting how code injection vulnerabilities directly enable adversaries to deliver and execute obfuscated malicious code, often as part of initial access or delivery phases. These cases illustrate how our knowledge graph produces *multi-hop paths* from critical software weaknesses to threat actor methodologies, bridging the gap between software engineering errors and operational TTPs.

By enumerating and analyzing such mappings, defenders can proactively prioritize mitigation efforts around weakness classes most commonly tied to adversary behaviors, ensuring that secure coding practices, patching strategies, and monitoring controls align with real-world attack techniques.

Fig. 7. Mapping of a top software CWE (CWE-287) to its corresponding CAPECs (light blue circles) and ATT&CK Techniques (light green circles).

B. ATT&CK Techniques, Groups, and Mitigations

Our pipeline centers each ATT&CK technique within its surrounding ecosystem by materializing typed, cross-domain relationships to the entities that cyber analysts regularly use: adversary groups (intrusion sets and campaigns), operational capabilities (tools and malware), and defensive knowledge (courses of action/mitigations, data sources, and

TABLE II
MAPPING OF THE TOP 25 SOFTWARE CWES TO ATT&CK TECHNIQUES VIA CAPEC, LIMITED TO CWES WITH AVAILABLE MAPPINGS.

CWE ID	ATT&CK Techniques
20 (Improper Input Validation)	T1539 (Steal Web Session Cookie); T1036.001 (Invalid Code Signature); T1027 (Obfuscated Files or Information); T1562.003 (Impair Command History Logging)
94 (Improper Control of Generation of Code ('Code Injection'))	T1027.006 (HTML Smuggling)
200 (Exposure of Sensitive Information to an Unauthorized Actor)	T1057 (Process Discovery); T1087 (Account Discovery); T1124 (System Time Discovery); T1018 (Remote System Discovery); T1135 (Network Share Discovery); T1007 (System Service Discovery); T1046 (Network Service Discovery); T1033 (System Owner/User Discovery); T1069 (Permission Groups Discovery); T1120 (Peripheral Device Discovery); T1082 (System Information Discovery); T1083 (File and Directory Discovery); T1134.001 (Token Impersonation/Theft); T1217 (Browser Information Discovery); T1562.003 (Impair Command History Logging); T1016 (System Network Configuration Discovery); T1111 (Multi-Factor Authentication Interception); T1036.005 (Match Legitimate Resource Name or Location)
269 (Improper Privilege Management)	T1548 (Abuse Elevation Control Mechanism)
287 (Improper Authentication)	T1505.003 (Web Shell); T1040 (Network Sniffing); T1557 (Adversary-in-the-Middle); T1134 (Access Token Manipulation); T1185 (Browser Session Hijacking); T1548 (Abuse Elevation Control Mechanism)
400 (Uncontrolled Resource Consumption)	T1499 (Endpoint Denial of Service)
434 (Unrestricted Upload of File with Dangerous Type)	T1574.010 (Services File Permissions Weakness)
798 (Use of Hard-coded Credentials)	T1078.001 (Default Accounts); T1552.001 (Credentials In Files)
862 (Missing Authorization)	T1211 (Exploitation for Defense Evasion)

data components). Because all ATT&CK object types are represented as distinct node labels with native relationships in Neo4j, technique-centric queries become straightforward: starting from a single technique, analysts can pivot *outward* to who uses it, *how* it is executed, *what* to monitor, and *how* to mitigate—without manual cross-referencing. This design supports both forward (technique→entities) and reverse (entities→technique) traversals, enabling explainable paths that tie behaviors to controls and evidence.

As shown in Fig. 8, a Cypher query can directly extract all entities connected to a given ATT&CK technique, spanning mitigations, data components, groups, malware, tools, and campaigns. These results can then be rendered as a graph structure (Fig. 9), providing analysts with an immediate view of how the technique relates to both adversary behaviors and defensive measures.

```
WITH "T1110.001" AS ap_id
MATCH (a:attack_pattern {external_id: ap_id})
MATCH p = (a)-[
:malware_to_attack_pattern
|course_of_action_to_attack_pattern
|data_component_to_attack_pattern
|intrusion_set_to_attack_pattern
|campaign_to_attack_pattern
|tool_to_attack_pattern
*1..1]-(n)
RETURN p
LIMIT 100;
```

Fig. 8. Cypher Query to obtain the one-hop neighborhood around a given ATT&CK technique ("T1110.001"), paths from the technique to connected mitigation, data components, groups, malware, tools, and campaigns.

Applied to a concrete technique (e.g., `T1110.001`) the one-hop neighborhood already yields actionable context: the graph links the technique to mitigations (Account Use Policies, Multi-factor Authentication, Password Policies, Update Software), to detection-relevant telemetry via data components (Application Log Content, User Account Authentication), to adversary groups (intrusion sets APT28 and APT29), and to concrete capabilities observed in the wild, including malware (China Chopper, Emotet, HermeticWizard, Lucifer, P.A.S. Webshell, Pony, SpeakUp, Xbash) and tools (CrackMapExec). By modeling these entities as nodes and typed edges, the graph contextualizes a single technique within an integrated framework that captures adversaries (*who*), capabilities (*with what*), relevant telemetry (*where to look*), and mitigations (*how to respond*).

Fig. 9. Visualizing ATT&CK "T1110.001" mapped to its respective Groups (in blue), Mitigations (in brown), and Tools (in pink).

Operationally, this structure accelerates common workflows: pivoting from techniques to groups for threat hunting and reporting; expanding to tools/malware for behavioral detections and emulation; and anchoring prevention plans by mapping courses of action to control requirements and relevant telemetry. In practice, the technique-centric view becomes a repeatable playbook for prioritization, detection engineering, and defensive validation—grounded in explicit, queryable relationships that remain current through automated updates.

C. Group to Weakness and Vulnerability Mapping

Our knowledge graph enables automatic extraction of the *CWE* classes and concrete *CVE* vulnerabilities that any given intrusion set is positioned to exploit. Starting from an `intrusion_set`, we traverse `intrusion_set` → `attack_pattern` ↔ `capec` ↔ `cwe` → `cve`. For each attack technique, we preserve provenance (which CAPEC produced which CWE) and compute the *union* of CWE identifiers per technique. When required, we extend further by one hop to enumerate the corresponding CVEs. The summarized outputs for two representative groups– Scattered Spider (G1015) and Volt Typhoon (G1017)—are shown in Tables III, and IV, with corresponding path visualizations in Figs. 10 and 11.

Scattered Spider (G1015) is a financially motivated intrusion set notorious for its social-engineering expertise. In September 2023, the group was behind the MGM Resorts breach, where operators successfully tricked a helpdesk employee into resetting credentials, leading to a multi-day outage that disrupted hotel operations and slot machines in Las Vegas [20]. Despite many of its members reportedly being teenagers, Scattered Spider has demonstrated advanced persistence by bypassing MFA protections and abusing trust relationships.

Table III summarizes how Scattered Spider's techniques map through CAPEC patterns to software weaknesses. A closer look at the first two rows illustrates the dynamics:

- T1018 (Remote System Discovery) links via CAPEC-292 to CWE-200 (Exposure of Sensitive Information to an Unauthorized Actor), highlighting how adversaries in G1015 abuse weaknesses in information exposure to enumerate networked assets and identify targets for lateral movement.
- T1083 (File and Directory Discovery) aggregates CAPEC-127 and CAPEC-497, yielding a diverse set of CWEs such as CWE-732 (Incorrect Permission Assignment), CWE-693 (Protection Mechanism Failure), and CWE-425 (Direct Request). This breadth shows that Scattered Spider leverages multiple file-system and access-control weaknesses to gain insight into system configurations and stored assets.

Beyond reconnaissance, techniques such as T1539 (Steal Web Session Cookie) connect through CAPEC-31 to a large family of weaknesses, including CWE-311 (Missing Encryption of Sensitive Data) and CWE-642 (External Control of Critical State Data), indicating the group's ability to abuse poor session and credential protections. Likewise, T1553.002 (Subvert Code Signing) and T1556.006 (Modify Authentication Process) map to critical weaknesses in trust boundaries (e.g., CWE-284: Improper Access Control), underscoring Scattered Spider's focus on authentication and persistence mechanisms.

TABLE III
ATT&CK TECHNIQUES WITH RELATED CAPECS AND CWES FOR INTRUSION SET - SCATTERED SPIDER (G1015)

Attack Technique	CAPEC IDs	CWE IDs
T1018	292	200
T1083	127, 497	732, 693, 425, 424, 288, 285, 276, 200
T1217	169	200
T1539	31	642, 602, 565, 539, 472, 384, 315, 311, 302, 20, 113
T1552.001	191	798
T1552.004	485, 474	330, 522
T1553.002	206, 68	732, 328, 325, 1326
T1556.006	578	284

Volt Typhoon (G1017) is a Chinese state-sponsored threat group disclosed in 2023 for its long-term infiltration of U.S. critical infrastructure networks, including telecommunications, energy, transportation, and water systems [21]. The group is distinguished not by immediate data theft but by "pre-positioning" — maintaining covert access and deploying malware in ways that could enable sabotage during a U.S.–China conflict. Intrusions have even been discovered on Guam, a strategic U.S. territory in the Pacific, raising concerns about operational resilience in crisis scenarios.

Table IV presents Volt Typhoon's mappings through CAPEC patterns to CWEs. One technique stands out:

- T1005 (Data from Local System) connects through CAPECs 37, 204, 545, 647 to an extensive set of CWEs, including CWE-1239 (Improper Zeroization), CWE-1258 (Sensitive Information Uncleared Before Release), and CWE-311 (Missing Encryption of Sensitive Data). This large mapping demonstrates that Volt Typhoon's ability to retrieve sensitive artifacts from local systems is rooted in wide-ranging systemic failures in data handling and confidentiality.
- T1078 (Valid Accounts) maps via CAPEC-560 to CWEs such as CWE-262 (Not Using Password Aging), CWE-263 (Password Aging with Insufficient Complexity), and CWE-654 (Reliance on a Single Factor), underscoring the group's reliance on credential abuse and insufficient account protection as a persistence vector.

Other reconnaissance-oriented techniques—including T1007 (System Service Discovery), T1016 (System Network Configuration Discovery), and T1018 (Remote System Discovery)—again concentrate on CWE-200 (Information Exposure), mirroring patterns observed with Scattered Spider. Meanwhile, techniques such as T1083 (File and Directory Discovery) and T1505.003 (Web Shell) connect to broader CWE sets (e.g., CWE-732, CWE-287, CWE-553), illustrating Volt Typhoon's use of misconfigured permissions and inadequate service controls to deepen operational footholds.

Beyond the CWE-level mappings, extending one additional hop in the graph yields the concrete *CVE* vulnerabilities associated with each intrusion set. For example, in the cases of G1015 and G1017, the resulting mappings are visualized in Figs. 12 and 13, where the diagrams displayed with fewer

TABLE IV
ATT&CK TECHNIQUES WITH RELATED CAPECS AND CWES FOR INTRUSION SET - VOLT TYPHOON (G1017)

Attack Technique	CAPEC IDs	CWE IDs
T1005	204, 647, 545, 37	524, 311, 1258, 1239, 285, 1330, 1323, 1278, 1272, 1266, 1243, 525, 318, 315, 314, 312, 226, 1301
T1007	574	200
T1016	309	200
T1018	292	200
T1033	577	200
T1036.005	616	200
T1046	300	200
T1057	573	200
T1069	576	200
T1078	560	654, 522, 309, 308, 307, 263, 262, 1273
T1082	312, 580, 313	200, 208, 205, 204
T1083	127, 497	732, 693, 425, 424, 288, 285, 276, 200
T1090.001	465	441
T1112	203	15
T1113	648	267
T1120	646	200
T1124	295	200
T1217	169	200
T1505.003	650	553, 287
T1552.004	485, 474	330, 522
T1614	694	497

nodes than actual. In both cases, the full traversal reveals 30,433 CVEs for G1015 and 19,242 for G1017, highlighting the wide range of exploitable vulnerabilities. This highlights how systemic weaknesses (CWEs) serve as bridges to a broad set of concrete vulnerabilities, offering both fine-grained detail for defenders and a scalable means of tracking group-level exploit potential.

D. CISA Advisory

CISA advisories list MITRE ATT&CK techniques and selected CVEs, but they do not fully map high-level TTPs to the broader set of relevant vulnerabilities. Traditionally, analysts must manually connect ATT&CK techniques to CAPECs, CWEs, and CVEs, a time-consuming and error-prone process. Our knowledge graph automates this bridging by traversing semantic relationships, including direct links (`related_capec`, `related_cwe`) and hierarchical expansions (child and precedence relations up to depth three), as illustrated in the Cypher query shown in Fig. 14.

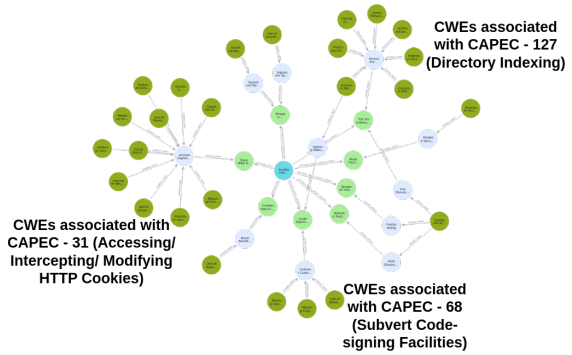

Fig. 10. Visualizing group-to-CWE mappings for Scattered Spider (G1015). The blue node represents group G1015, and dark green nodes represent CWEs.

Case Study 1: Russian GRU Targeting Western Logistics Entities and Technology Companies. On May 21, 2025, CISA released advisory AA25-141A, which highlights a Russian state-sponsored cyber campaign attributed to the GRU's 85th Main Special Service Center (Unit 26165). The campaign has targeted Western logistics entities and technology companies involved in coordinating, transporting, and delivering foreign assistance to Ukraine. Since 2022, these sectors have faced an elevated risk as targets. The advisory warns that executives and network defenders in these industries should recognize the heightened threat level, increase monitoring for known TTPs and indicators of compromise (IOCs), and adopt defensive postures that assume persistent targeting.

The advisory details a set of initial access techniques used by the threat actors, including brute force login attempts (T1110.001, T1110.003), spearphishing (T1566), exploitation of Internet-facing infrastructure (T1133, T1190), WinRAR and Outlook vulnerabilities (CVE-2023-38831, CVE-2023-23397), and abuse of SOHO devices (T1665). These techniques only partially reveal the broader vulnerability surface. When executed in the canonical order (Technique →CAPEC→CWE→CVE), the query does not return all CVEs cited in the advisory. Incorporating hierarchical relations such as `capec_precedence`, `capec_child`, `cwe_precedence`, and `cwe_child` surfaces `CVE-2023-23397`, a critical Outlook NTLM vulnerability. With two-hop expansions, the query retrieves not only the explicitly mentioned CVEs but also nearly 60% of all CVEs in the graph, showing how our framework operationalizes advisories to uncover latent vulnerabilities beyond those listed in CISA reports.

Case Study 2: #StopRansomware Medusa Ransomware. In February 2025, the FBI, CISA, and MS-ISAC released a joint #StopRansomware advisory on the Medusa ransomware variant. Medusa, first identified in 2021, has impacted more

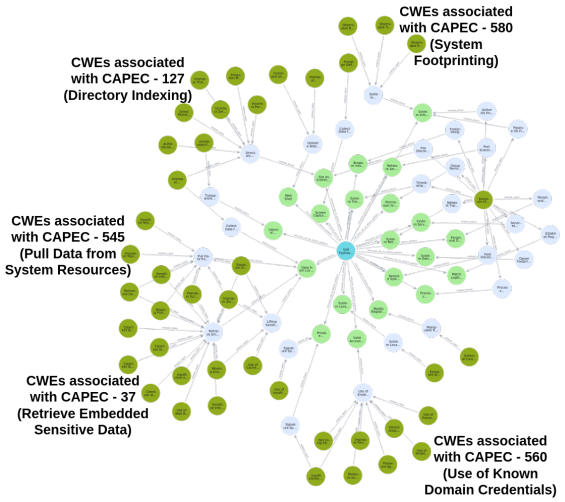

Fig. 11. Visualizing group-to-CWE mappings for Volt Typhoon (G1017). The blue node represents group G1017, and dark green nodes represent CWEs.

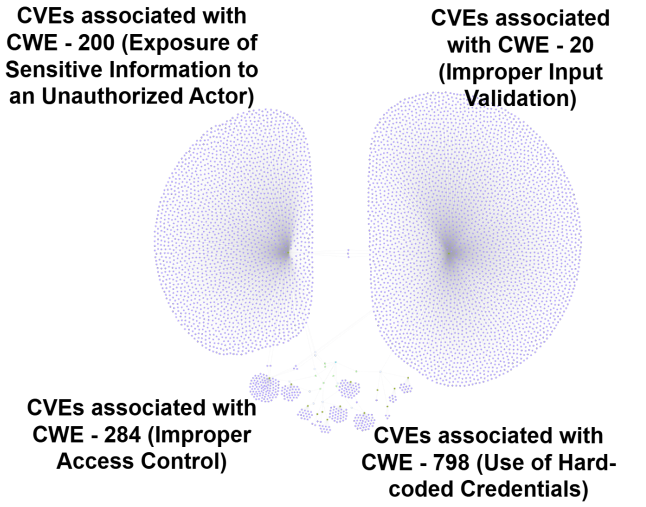

Fig. 12. Mappings for intrusion set G1015 to CVEs. Blue nodes represent groups, green nodes represent CWEs, and violet represents CVEs.

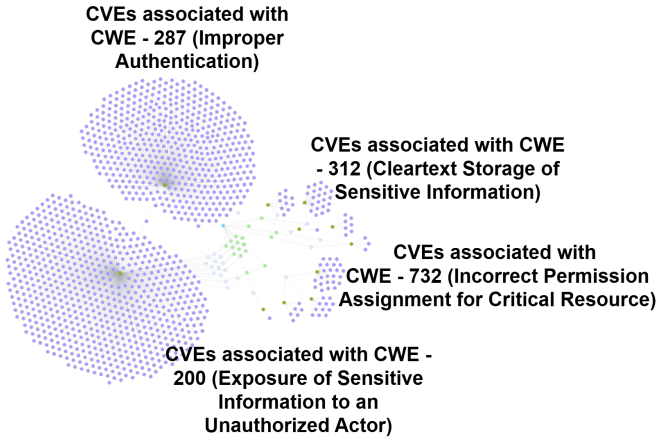

Fig. 13. Mappings for intrusion set G1017 to CVEs. Blue nodes represent groups, green nodes represent CWEs, and violet represents CVEs.

than 300 victims across healthcare, education, legal, insurance, technology, and manufacturing. Unlike MedusaLocker or Medusa mobile malware, Medusa is a distinct ransomware-as-a-service (RaaS) operation, with affiliates frequently purchasing access from initial access brokers (IABs).

The advisory highlights common TTPs for initial access, including phishing campaigns [T1566] and exploitation of unpatched software vulnerabilities [T1190]. Specifically, Medusa affiliates leveraged the ScreenConnect vulnerability `CVE-2024-1709` (CWE-288: Authentication Bypass) and the Fortinet EMS SQL injection vulnerability `CVE-2023-48788` (CWE-89: SQL Injection). These examples illustrate how Medusa combines social engineering with the exploitation of critical software flaws.

When we mapped the advisory's ATT&CK techniques into our knowledge graph, their connections to CWEs and CVEs did not immediately retrieve the specific CVEs listed in the advisory using direct traversals (Technique $\rightarrow$ CAPEC $\rightarrow$ CWE $\rightarrow$ CVE). However, by enabling two-hop hierarchical expansions across CAPEC and CWE relations, the graph

```
MATCH (a:attack_pattern)
WHERE a.external_id IN [
"T1110.001","T1110.003","T1566", "T1133","T1190","T1665"
]

// CAPECs (base + expansion)
MATCH (capec0:capec)-[:related_attack]->(a)
OPTIONAL MATCH (capec0)-[:capec_precedence|
capec_child*1..2]-(capecX:capec)
WITH [c IN (collect(DISTINCT capec0) + collect(DISTINCT
capecX)) WHERE c IS NOT NULL] AS capecs
UNWIND capecs AS capec
WITH DISTINCT capec

// Base CWEs
MATCH (capec)-[:related_capec]-(cwe0:cwe)

// CWE expansion
OPTIONAL MATCH (cwe0)-[:cwe_precedence|
cwe_child*1..2]-(cweX:cwe)
WITH [w IN (collect(DISTINCT cwe0) + collect(DISTINCT
cweX)) WHERE w IS NOT NULL] AS cwes
UNWIND cwes AS cwe
WITH DISTINCT cwe

// To CVEs
MATCH (cve:cve)-[:related_cwe]-(cwe)
RETURN DISTINCT cve.id AS cve_id
ORDER BY cve_id;
```

Fig. 14. Cypher Query to traverse from ATT&CK techniques to CVEs via CAPECs and CWEs.

successfully revealed all of the reported CVEs while also identifying further related vulnerabilities.

Together, these case studies demonstrate how the knowledge graph can operationalize CISA advisories by systematically bridging ATT&CK techniques to underlying CWEs and CVEs. By incorporating hierarchical expansions, the framework not only recovers explicitly listed vulnerabilities but also uncovers a broader set of latent CVEs, providing defenders with a more comprehensive view of potential exploitation surfaces.

E. Topology-Based Discovery of Missing Links

Topology-based similarity measures can be applied to the cybersecurity knowledge graph to identify *plausible but unobserved relationships* between entities. For example, while some CWE and CAPEC nodes are not explicitly linked, their structural context within the graph may suggest a strong latent association. Using Cypher queries, we can systematically enumerate candidate CWE–CAPEC pairs within a bounded hop distance and rank them by metrics such as Common Neighbors, Adamic–Adar, and Preferential Attachment. This allows analysts to focus on the most promising high-scoring pairs, highlighting potential links that warrant examination.

We illustrate this approach through two case studies. In both, the CWE and CAPEC nodes are *not directly connected*, yet their similarity scores from the Neo4j Graph Data Science (GDS) library indicate likely relationships.

Case Study 3 (CWE-308 and CAPEC-151) These nodes share 14 common neighbors and have an Adamic–Adar score of 5.41, indicating strong and distinctive structural overlap. Semantically, CWE-308 (Use of Single-Factor Authentication) and CAPEC-151 (Identity Spoofing) are closely related, as weak authentication mechanisms inherently enable spoofing-based attacks. The structural and semantic evidence together highlight the plausibility of a missing edge.

Case Study 4 (CWE-20 and CAPEC-100) These nodes share 13 common neighbors with an Adamic–Adar score of 4.95, showing significant and informative contextual overlap. CWE-20 (Improper Input Validation) and CAPEC-100 (Overflow Buffers) are conceptually linked, since insufficient input validation frequently leads to buffer overflow vulnerabilities. The alignment of structural similarity with semantic meaning reinforces the likelihood of a missing but meaningful relationship.

These results suggest plausible direct relationships that may be absent in the current graph representation. From an analytical perspective, such scoring provides valuable guidance for uncovering missing links that reflect real-world attack–weakness associations, thereby enriching the knowledge graph and supporting proactive cybersecurity analysis.

V. Conclusion

We presented a fully automated framework for building a cybersecurity knowledge graph that unifies heterogeneous cyber threat intelligence (CTI) sources—including CVE, CWE, CAPEC, and MITRE ATT&CK—into a large-scale, continuously updated Neo4j graph database. Our framework ingests, normalizes, and integrates hundreds of thousands of entities and relationships, capturing vulnerabilities, weaknesses, attack patterns, techniques, adversary groups, tools, defensive measures, etc. within a single queryable structure. By automating cross-domain integration and materializing rich semantic relationships, the graph enables bidirectional, multi-hop traversals that provide analysts with explainable connections from low-level CVEs to high-level ATT&CK techniques, and vice versa.

Through case studies on CISA advisories, group-to-vulnerability traversals, and topology-based link scoring, we demonstrated how the knowledge graph not only recovers explicitly reported vulnerabilities but also uncovers latent and plausible connections. These capabilities highlight the utility of graph-based CTI for proactive defense, adversary emulation, and explainable cyber risk assessment.

Our future work will explore enriching the graph with real-time streaming threat feeds, extending coverage to additional CTI standards (e.g., STIX/TAXII threat sharing, malware repositories, and open-source threat reports), and applying advanced graph learning models such as Graph Neural Networks (GNNs) to enhance predictive reasoning and link prediction. By combining automation, scalable analytics, and predictive modeling, we aim to advance knowledge graphs as a foundational capability for next-generation cyber threat intelligence.

Acknowledgment

This work was supported by the U.S. Department of Energy's Office of Science at the Pacific Northwest National Laboratory, which is operated by Battelle for the U.S. Department of Energy under contract DE-AC05-76RL01830. This work was also partially supported by the National Science Foundation (NSF) under Grant 2442595.

References

[1] "National Vulnerability Database (NVD)," National Institute of Standards and Technology (NIST), 2005–2025, comprehensive repository of standards-based vulnerability management data. [Online]. Available: https://nvd.nist.gov/

[2] "Common Vulnerabilities and Exposures (CVE)," MITRE, 1999–2025, publicly disclosed cybersecurity vulnerabilities. [Online]. Available: https://cve.mitre.org/

[3] "MITRE Corporation," Nonprofit organization managing federally funded research and development centers (FFRDCs), 1958–2025. [Online]. Available: https://www.mitre.org/

[4] "Common Weakness Enumeration (CWE)," MITRE, 2006–2025, community-developed catalog of software and hardware weakness types. [Online]. Available: https://cwe.mitre.org/

[5] "Common Attack Pattern Enumeration and Classification (CAPEC)," MITRE, 2007–2025, catalog of adversary attack patterns. [Online]. Available: https://capec.mitre.org/

[6] "Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK)," MITRE, 2013–2025, knowledge base of adversary tactics, techniques, and procedures. [Online]. Available: https://attack.mitre.org/

[7] "Common Platform Enumeration (CPE)," National Institute of Standards and Technology (NIST), 2007–2025, standardized method of naming software, operating systems, and hardware platforms. [Online]. Available: https://nvd.nist.gov/products/cpe

[8] "Common Vulnerability Scoring System (CVSS)," National Institute of Standards and Technology (NIST), 2005–2025, standardized framework for rating the severity of security vulnerabilities. [Online]. Available: https://nvd.nist.gov/vuln-metrics/cvss

[9] E. Kiesling, A. Ekelhart, K. Kurniawan, and F. Ekaputra, "The sepses knowledge graph: an integrated resource for cybersecurity," in *International Semantic Web Conference*. Springer, 2019, pp. 198–214.

[10] P. Falcarin, R. Dainese, and M. Morelli, "Building a cybersecurity knowledge graph with cybergraph," in *Proc. of EnCyCriS/SVM*, 2024.

[11] Z. Shi, N. Matyunin, K. Graffi, and D. Starobinski, "Uncovering cwe–cve–cpe relations with threat knowledge graphs," *arXiv preprint arXiv:2305.00632*, 2023.

[12] B. Ampel, S. Samtani, S. Ullman, and H. Chen, "Linking common vulnerabilities and exposures to the mitre att&ck framework: A self-distillation approach," *arXiv preprint arXiv:2108.01696*, 2021.

[13] S. S. Das, E. Serra, M. Halappanavar, A. Pothen, and E. Al-Shaer, "V2w-bert: A framework for effective hierarchical multiclass classification of software vulnerabilities," in *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2021, pp. 1–12.

[14] E. Hemberg, J. Kelly, M. Shlapentokh-Rothman, B. Reinstadler, K. Xu, N. Rutar, and U.-M. O'Reilly, "Linking threat tactics, techniques, and patterns with defensive weaknesses, vulnerabilities and affected platform configurations for cyber hunting," *arXiv preprint arXiv:2010.00533*, 2020.

[15] G. Shen, W. Wang, Q. Mu, and M. Yu, "Data-driven cybersecurity knowledge graph construction for industrial control system security," *Wireless Communications and Mobile Computing*, 2020.

[16] P. E. Kaloroumakis and M. J. Smith, "Toward a knowledge graph of cybersecurity countermeasures (d3fend)," MITRE Corporation, Tech. Rep., 2019.

[17] K. Kurniawan, E. Kiesling, and A. Ekelhart, "Cykg-rag: Towards knowledge-graph enhanced retrieval for cybersecurity," in *Proceedings of CEUR Workshop on Knowledge-Augmented AI*, 2025.

[18] M. Rahman, K. O. Piryani, A. M. Sanchez, S. Munikoti, L. De La Torre, M. S. Levin, M. Akbar, M. Hossain, M. Hasan, and M. Halappanavar, "Retrieval augmented generation for robust cyber defense," Pacific Northwest National Laboratory (PNNL), Richland, WA (United States), Tech. Rep., 2024.

[19] Neo4j, Inc., *Link Prediction Algorithms - Neo4j Graph Data Science Library*, 2025, accessed: 2025-08-29. [Online]. Available: https://neo4j.com/docs/graph-data-science/current/algorithms/linkprediction/

[20] Reuters, "Moody's says mgm breach is credit negative as disruption lingers," https://www.reuters.com/technology/moodys-says-breach-mgm-is-credit-negative-disruption-lingers-2023-09-13/, 2023, accessed: 2025-08-26.

[21] Cybersecurity and Infrastructure Security Agency (CISA), "People's republic of china state-sponsored cyber actor living off the land to evade detection," https://www.cisa.gov/news-events/cybersecurity-advisories/aa24-038a, 2024, accessed: 2025-08-26.