Threat Knowledge Graphs

Supervisor: Dr. Cogan Shimizu

Presented by: Rakesh Kandula

Contents

1) A Practical Approach to Constructing a Knowledge Graph for Cybersecurity
   i. Motivation behind research
   ii. Related Work
   iii. Framework
   iv. Future Work
2) Uncovering Product Vulnerabilities with Threat Knowledge Graphs
   i. Motivation behind research
   ii. Expected Outcome
   iii. Related Work
   iv. Structure of Knowledge Graph(KG)
   v. Example Predictions
   vi. Research Directions
3) Usage of current research in our research

```
or_mod = modifier_ob
t mirror object to mirror
irror_mod.mirror_object
peration == "MIRROR_X":
irror_mod.use_x = True
irror_mod.use_y = False
irror_mod.use_z = False
_operation == "MIRROR_Y":
irror_mod.use_x = False
irror_mod.use_y = True
irror_mod.use_z = False
_operation == "MIRROR_Z":
irror_mod.use_x = False
irror_mod.use_y = False
irror_mod.use_z = True

election at the end -add
_ob.select= 1
er_ob.select=1
ntext.scene.objects.acti
("Selected" + str(modifier
irror_ob.select = 0
bpy.context.selected_ob
ata.objects[one.name].se

int("please select exactl

-- OPERATOR CLASSES ----

types.Operator):
X mirror to the selected
ject.mirror_mirror_x"
ror X"

ontext):
ext.active_object is not
```

A Practical Approach to Constructing a Knowledge Graph for Cybersecurity

Motivation behind Research

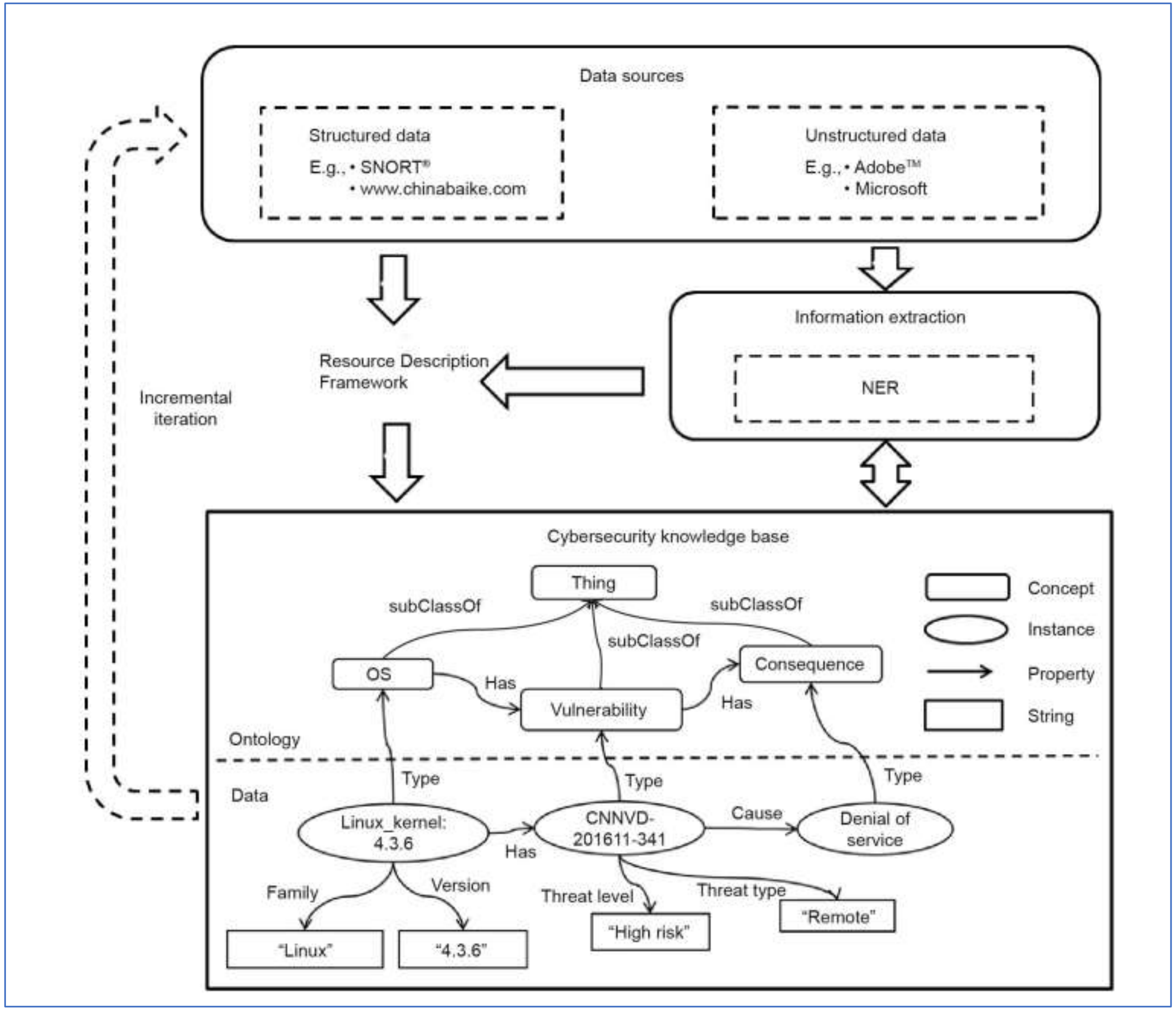
1. The main idea is to develop a knowledge graph for the cybersecurity.
2. Integrating the cybersecurity knowledge from different sources into an interconnected CSKG.
3. The usage of knowledge graphs in domain specific applications are giving better/mature results.
4. Linking different databases like vulnerabilities, attacks, assets etc. provides a benchmark for developing intelligent applications in this field.

Related Work

1. Ontology Construction
2. Information Extraction
3. Cybersecurity Knowledge base

Framework

1. Data Sources
2. Ontology Construction
3. Information Extraction
4. Knowledge base Generation
5. Knowledge Deduction



Future Work

1. The paper develops an ontology and knowledge graph integrating vulnerability data to represent cybersecurity concepts and their relationships. The Stanford NER tool was used to extract entities from text, but further improvement in accuracy is needed.
2. Deduction capabilities were demonstrated for enriching the knowledge model. Next steps are expanding the knowledge base and applying it to cybersecurity use cases like intrusion detection.

Uncovering Product Vulnerabilities with Threat Knowledge Graphs

Motivation behind research

1. Analyzing the threat databases like Common Platform Enumeration (CPE), Common Vulnerabilities and Exposures (CVE) and Common Weakness Enumeration (CWE) and integrating them together to reveal the new insights about the product vulnerabilities.
2. Merging the CPE, CVE, CWE databases to create the threat knowledge graph.
3. To uncover the hidden associations of the product and its vulnerabilities.

Expected Outcome

Authors propose and implement the concept of Threat knowledge graph by doing:

i. Translating the entries in threat databases and their associations into triples that forms Knowledge Graph (KG).

ii. Embedding the KG onto a vector space that can be used for link prediction

Related Work

- Prior work has leveraged threat databases like CVE and CWE to gain insights about software vulnerabilities through data mining.
- However, the works treated the databases separately.
- Extracting related CPE entries from CVE descriptions using natural language processing techniques.
- Existing works produced knowledge graphs from CVE-CWE entries for querying and visualization. Did not consider CPE products.

Structure of Knowledge Graph

- CPE Entries:
  - 858,409
- CVE Entries:
  - 183,368
- CWE Entries:
  - 924

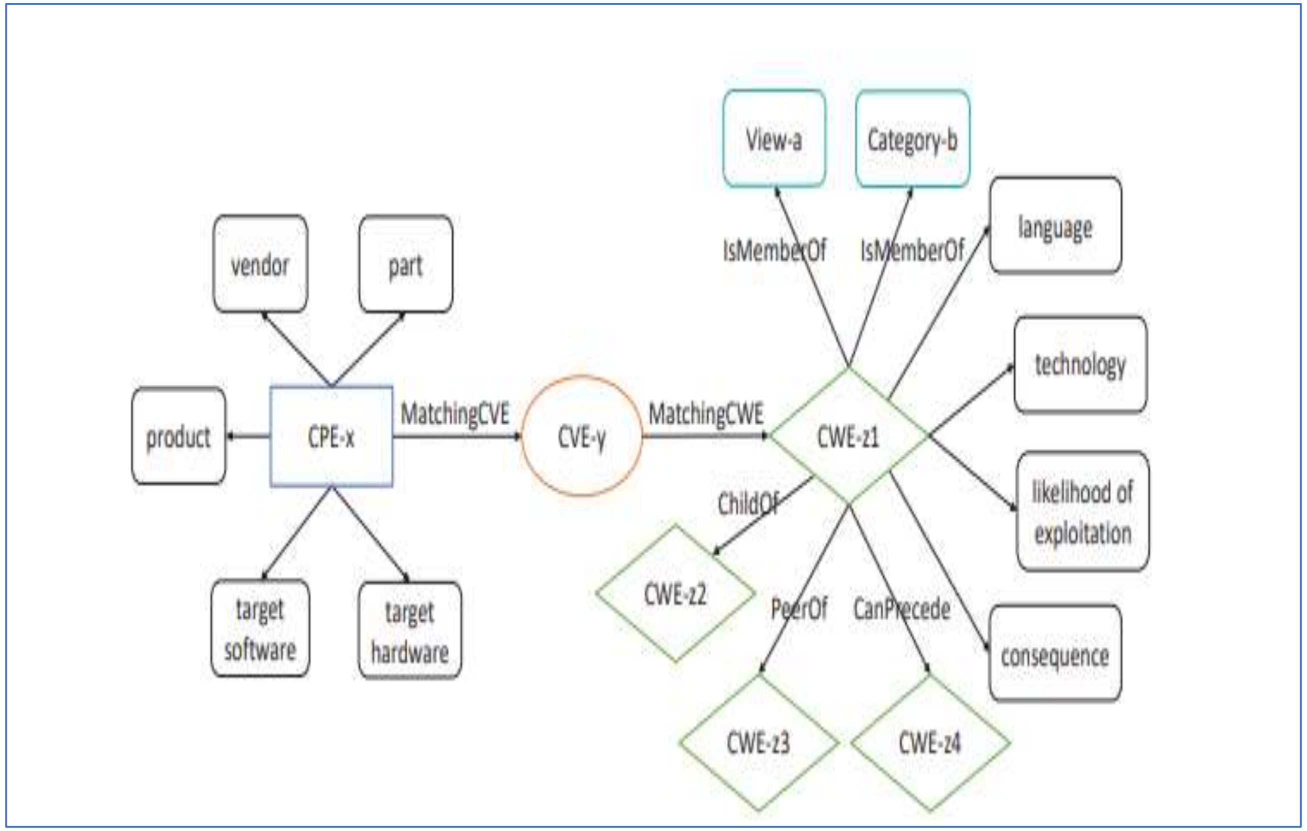
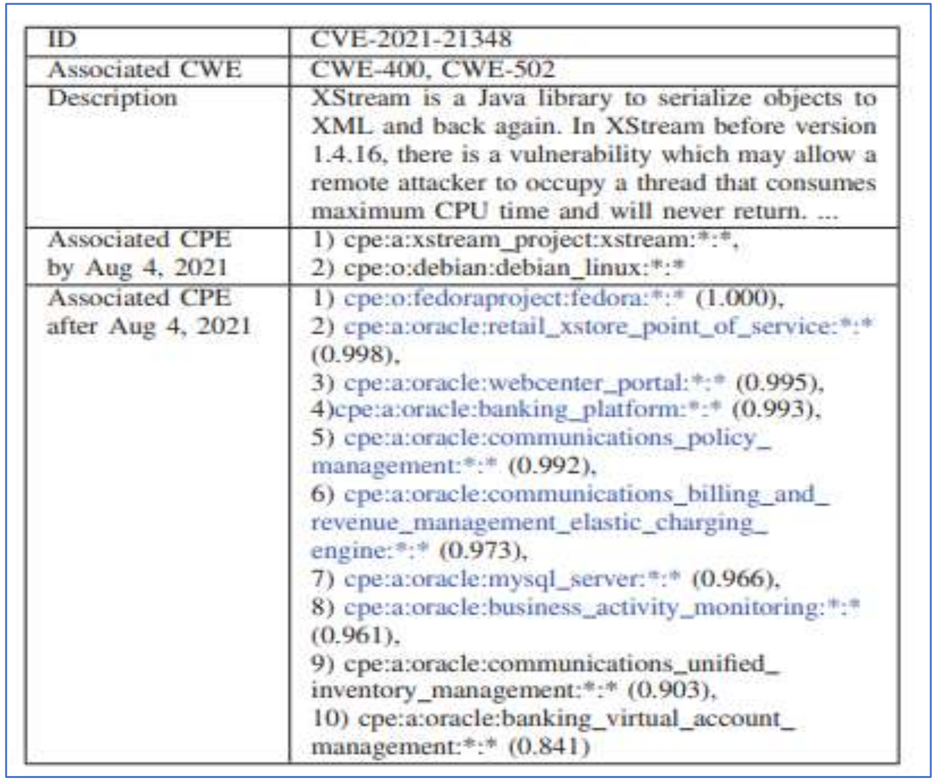


Figure: Complete structure of the threat knowledge graph

Example predictions



ID	CVE-2021-21348
Associated CWE	CWE-400, CWE-502
Description	XStream is a Java library to serialize objects to XML and back again. In XStream before version 1.4.16, there is a vulnerability which may allow a remote attacker to occupy a thread that consumes maximum CPU time and will never return. ...
Associated CPE by Aug 4, 2021	1) cpe:a:xstream_project:xstream:*:*, 2) cpe:o:debian:debian_linux:*:*
Associated CPE after Aug 4, 2021	1) cpe:o:fedoraproject:fedora:*:* (1.000), 2) cpe:a:oracle:retail_xstore_point_of_service:*:* (0.998), 3) cpe:a:oracle:webcenter_portal:*:* (0.995), 4)cpe:a:oracle:banking_platform:*:* (0.993), 5) cpe:a:oracle:communications_policy_management:*:* (0.992), 6) cpe:a:oracle:communications_billing_and_revenue_management_elastic_charging_engine:*:* (0.973), 7) cpe:a:oracle:mysql_server:*:* (0.966), 8) cpe:a:oracle:business_activity_monitoring:*:* (0.961), 9) cpe:a:oracle:communications_unified_inventory_management:*:* (0.903), 10) cpe:a:oracle:banking_virtual_account_management:*:* (0.841)

Figure: Prediction results

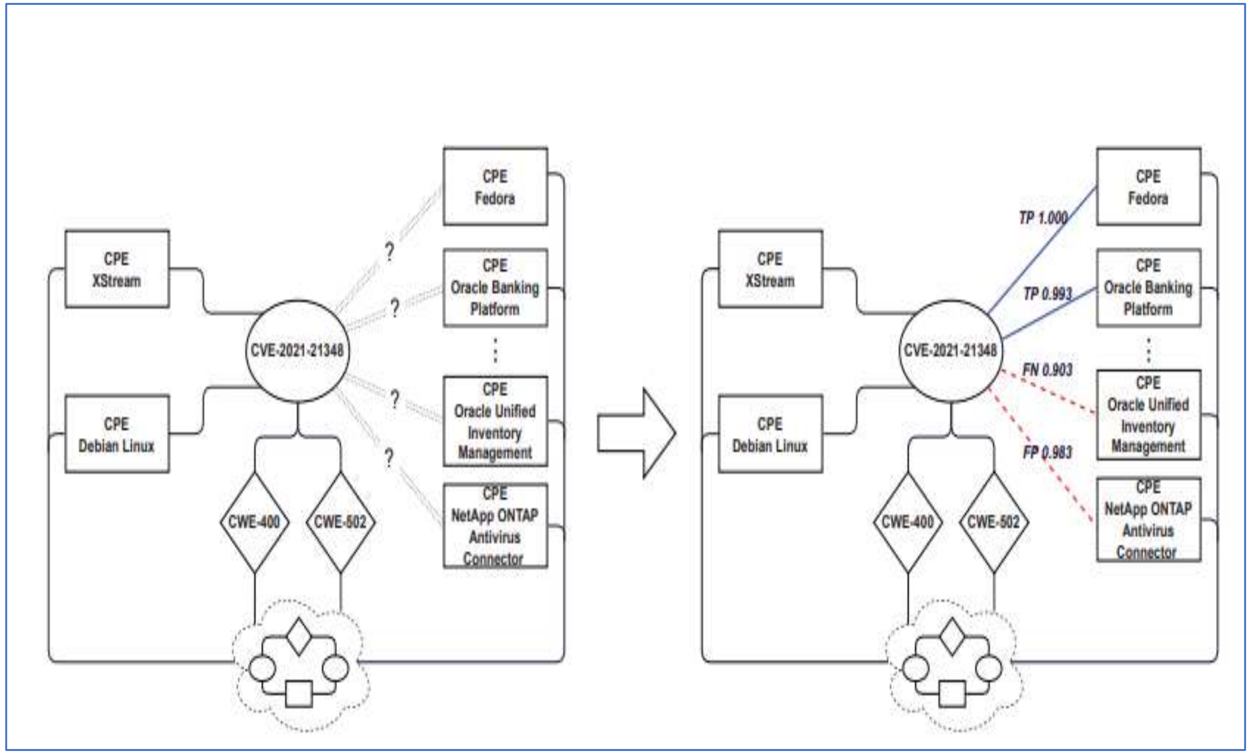

CPE XStream
CPE Debian Linux
CVE-2021-21348
CWE-400
CWE-502
?
?
?
?
CPE Fedora
CPE Oracle Banking Platform
CPE Oracle Unified Inventory Management
CPE NetApp ONTAP Antivirus Connector
CPE XStream
CPE Debian Linux
CVE-2021-21348
CWE-400
CWE-502
TP 1.000
TP 0.993
FN 0.903
FP 0.983
CPE Fedora
CPE Oracle Banking Platform
CPE Oracle Unified Inventory Management
CPE NetApp ONTAP Antivirus Connector

Figure: Prediction process

Research directions

- Develop specialized knowledge graph embedding techniques tailored for cyber threat data that can improve prediction performance.
- Incorporate additional threat databases like CAPEC (Common Attack Pattern Enumeration and Classification) into the knowledge graph for more comprehensive analysis.
- Study how knowledge graphs can enhance existing threat modeling tools and methodologies.

Usage of current research in our research

The idea of aggregating and interlinking data from different hardware-focused threat databases like Common Platform Enumeration (CPE), Common Vulnerabilities and Exposures (CVE) etc. into a knowledge graph can be applied to create a hardware-specific graph.

Relationships between hardware components, vendors, known vulnerabilities, certifications etc. can be captured in the knowledge graph. Graph embedding techniques can then be leveraged for inference tasks.

Link prediction methodology presented in the paper can be adapted to predict associations between hardware components and undisclosed vulnerabilities affecting trustworthiness.

Knowledge graph can integrate unstructured data like vulnerability reports, security advisories related to hardware components using information extraction techniques discussed.

Graph reasoning and link prediction approaches can help uncover hidden vulnerabilities in hardware components that impact trust. For example, predicting components vulnerable to new exploits.

Continuously expanding the hardware knowledge graph with new data sources can enable updating trust assessments of components.

thank you