The Design of an Ontology for ATT&CK and its Application to Cybersecurity

Khandakar Ashrafi Akbar
KhandakarAshrafi.Akbar@utdallas.edu
University of Texas Dallas
Richardson, Texas, USA

Sadaf Md Halim
sadafmd.halim@utdallas.edu
University of Texas Dallas
Richardson, Texas, USA

Anoop Singhal
anoop.singhal@nist.gov
National Institute of Standards and Technology
Gaithersburg, Maryland, USA

Basel Abdeen
basel.abdeen@utdallas.edu
University of Texas Dallas
Richardson, Texas, USA

Latifur Khan
lkhan@utdallas.edu
University of Texas Dallas
Richardson, Texas, USA

Bhavani Thuraisingham
bxt043000@utdallas.edu
University of Texas Dallas
Richardson, Texas, USA

ABSTRACT

The spread of attacks in computer networks and within systems can have severe consequences for both individuals and organizations. One approach to preventing the spread of attacks is to use ontological aid, which is the use of ontologies to provide a structured representation of knowledge about the attack and its components, especially the ones who often disguise themselves to remain undetected for a long time within the system. As soon as one particular stage of such an attack is detected, it is imperative to reduce the amount of spread so that no permanent damage can be done. For this, the security analyst must boil down to technical details from a behavioral perspective so that proper defensive initiatives can be taken. We propose an ontology that will aid security analysts to find out the list of vulnerabilities to be patched so that an ongoing attack campaign can be prevented from spreading even more.

CCS CONCEPTS

• **Security and privacy** → *Vulnerability management*.

KEYWORDS

Ontology, MITRE ATT&CK Framework, Cyber Response

ACM Reference Format:
Khandakar Ashrafi Akbar, Sadaf Md Halim, Anoop Singhal, Basel Abdeen, Latifur Khan, and Bhavani Thuraisingham. 2023. The Design of an Ontology for ATT&CK and its Application to Cybersecurity. In *Proceedings of the Thirteenth ACM Conference on Data and Application Security and Privacy (CODASPY '23), April 24–26, 2023, Charlotte, NC, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3577923.3585051

1 INTRODUCTION

As computer networks grow ever larger, malicious entities can coordinate large-scale attacks on various systems over these networks. This puts individuals and organizations on the network at risk. Preventing these attacks as well as responding to them remains an ongoing effort, as attackers continue to find newer ways to breach systems and evolve their methods. In this continuous fight, it is essential to develop tools that help security analysts address attacks as soon as they occur, and possibly even preclude them from occurring in the first place.

A key challenge that analysts face is the sheer amount of information that the analyst needs immediate access to, in order to be able to properly analyze the attacks as they come. This information gap can severely hinder the analyst when responding to a time-sensitive threat. There are various tools that security analyst can leverage in their work, and we propose a tool that closes this information gap by bringing relevant knowledge to the analysts' fingertips in very few steps. We do this by first creating an ontology. Ontologies are formal representations of knowledge about a domain, and they provide a structured way to represent the components and relationships of a network. By using ontologies, it is possible to gain a deeper understanding of what can be possibly done next, which can aid in identifying and preventing the spread of attacks.

In this work, we propose an ontology that helps the security analyst identify and evaluate vulnerabilities in the system and understand possible attacks as they happen. In an attack scenario, time is of the essence, and therefore having a readily available, structured knowledge base built using a sophisticated ontology will aid the analyst immensely. It enables the analyst to quickly query for important relationships and identify possible solutions on the fly. It can help the analyst identify particular applications which are vulnerable and, in the worst case, the analyst can take them down until the application is patched to remove the vulnerability.

To construct this knowledge base, we design it as an ontology that contains the relationships between attack tactics and techniques and existing vulnerabilities reported in the CVE database. The relationships themselves can be produced through automated or semi-automated techniques for associating attack tactics and techniques with these vulnerabilities. Furthermore, we bolster this ontology by incorporating knowledge found in the MITRE database. As an illuminating example of how our ontology can be useful, consider the Advanced Persistent Threat (APT) attack scenario. There are seven stages of the APT kill chain which are as follows: 1) Initial Compromise, 2) Establish Foothold, 3) Escalate Privileges, 4) Internal Reconnaissance, 5) Move Laterally, 6) Maintain Presence,

CODASPY '23, April 24–26, 2023, Charlotte, NC, USA

ACM ISBN 979-8-4007-0067-5/23/04.
https://doi.org/10.1145/3577923.3585051

Component Name	Instances Count
Stages	7
Tactics	14
Techniques	193
Sub-Techniques	378
CVETags	313

Table 1: Instances Count of the Ontology Components

and 7) Complete Mission. Each of these APT stages can be mapped to one or multiple tactics from the attack framework. If an ongoing stage of the APT campaign has been identified within a system, we can infer what tactics might be used for the next stage. If for each of those probable tactics, we are able to list down the relevant vulnerabilities and remedy them, then the APT campaign can be stopped. This is where our Ontology can directly come into play, by enabling the security analyst to quickly query and identify important relationships between attack strategies and vulnerabilities.

For these reasons, we propose a framework for the design of an ontology that will help bridge the information gap for security analysts as they tackle an ever-increasing body of different attacks and threat types over computer networks.

2 ONTOLOGICAL DESIGN

In the context of knowledge sharing, the term ontology means the specification of a conceptualization. An ontology is a description (like a formal specification of a program) of the concepts and relationships that can exist for an agent or a community of agents. This definition is consistent with the usage of ontology as set-of-concept definitions, but more general [6].

The ontology of this work has been designed using the cyber threat intelligence from MITRE ATT&CK framework [4] and CVE database [1]. An Advanced Persistent Threat (APT) attack is classified based on the seven stages of the APT kill chain: 1) Initial Compromise, 2) Establish Foothold, 3) Escalate Privileges, 4) Internal Reconnaissance, 5) Move Laterally, 6) Maintain Presence, and 7) Complete Mission. These seven stages can use any of the tactics from the MITRE ATT&CK framework. The tactics have different techniques and those might contain different sub-techniques within the hierarchy. Each of the techniques or sub-techniques is mostly connected to numerous CVE vulnerabilities which are listed in the CVE database. CVE vulnerability data are taken from National Vulnerability Database (NVD) xml feeds provided by the National Institute of Standards and Technology (NIST).

The Resource Description Framework (RDF) is a general framework for representing interconnected data on the web. RDF statements are used for describing and exchanging metadata. RDF is used to integrate data from multiple sources [3]. SPARQL is an RDF query language—that is, a semantic query language for databases—able to retrieve and manipulate data stored in Resource Description Framework format. SPARQL has been used for query execution on top of the RDFs generated as part of the ontology.

The ontology has the following classes: 1) Stages, 2) Tactics, 3) Techniques, and 4) Sub-Techniques. It also has three main object properties named 1) uses, 2) has, and 3) contains. The stages use tactics, each of the tactics has techniques, and each of the techniques might contain sub-techniques. The data property CVETag belongs to both techniques and sub-techniques. The association of the techniques and sub-techniques with CVE tags is done manually

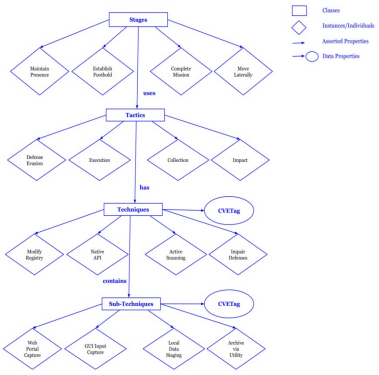

Figure 1: Ontological Representation

```
SPARQL query:

PREFIX attackCVE: <http://www.semanticweb.org/ashrafi/ontologies/2022/11/attackCVE#>
SELECT ?technique ?value
WHERE{
        ?technique attackCVE:CVETag ?value.
        FILTER(?technique = <http://www.semanticweb.org/ashrafi/ontologies/2022/11/attackCVE#Command_and_Scripting_Interpreter>).
}
```

Figure 2: SPARQL Query to Retrieve CVE Tags for a Certain Adversarial Technique (Command and Scripting Interpreter)

```
SPARQL query:

PREFIX attackCVE: <http://www.semanticweb.org/ashrafi/ontologies/2022/11/attackCVE#>
SELECT ?CVE ?Products
WHERE{
        ?CVE attackCVE:residesIn ?Products.
        FILTER(?CVE= <http://www.semanticweb.org/ashrafi/ontologies/2022/11/attackCVE#CVE-2022-24663>).
}
```

Figure 3: SPARQL Query to Retrieve Products that Might Contain a Particular CVE Vulnerability (Complete Mission)

```
SPARQL query:

PREFIX attackCVE: <http://www.semanticweb.org/ashrafi/ontologies/2022/11/attackCVE#>
SELECT ?Products ?Product_Type ?Vendor ?Product ?Version ?Update ?Edition ?Language
WHERE{
        ?Products attackCVE:Product_Type ?Product_Type.
        ?Products attackCVE:Vendor ?Vendor.
        ?Products attackCVE:Product ?Product.
        ?Products attackCVE:Version ?Version.
        ?Products attackCVE:Update ?Update.
        ?Products attackCVE:Edition ?Edition.
        ?Products attackCVE:Language ?Language.
        FILTER(?Products = <http://www.semanticweb.org/ashrafi/ontologies/2022/11/attackCVE#Product_1>).
}
```

Figure 4: SPARQL Query to Retrieve Product Specifics

for now. This association task can be automated or semi-automated using natural language processing (NLP) based techniques. In figure 1, we demonstrate how the different instances of the classes are connected using the object properties and what specific data properties they do have. In a nutshell, we demonstrate our ontological representation of the entities and their relationships. The "stages" use "tactics", the "tactics" "have" techniques, and the "techniques" contain "sub-techniques". In this particular figure 1, we also provided with few examples of each of the entities from the MITRE ATT&CK framework. In table 1, we portray the count of instances of different entities and their properties used to build up the ontology. This ontology provides the association between adversarial techniques and CVE tags.

2.1 Example SPARQL Queries

We discuss some of the queries performed on our ontology to demonstrate the use cases of this ontology from a high level.

SPARQL query:

```
PREFIX attackCVE: <http://www.semanticweb.org/ashrafi/ontologies/2022/11/attackCVE#>
SELECT ?stage ?tactic ?technique ?subtechnique
WHERE{
        ?stage attackCVE:uses ?tactic.
        ?tactic attackCVE:has ?technique.
        ?technique attackCVE:contains ?subtechnique.
        ?technique attackCVE:CVETag "CVE-2019-1943".
}
```

Figure 5: SPARQL Query for Entity Connections to a CVE Tag (CVE-2019-1943)

Stage	Tactic	Technique	Sub-Technique
Complete Mission	Impact	Data Manipulation	Transmitted Data Manipulation
Complete Mission	Impact	Data Manipulation	Stored Data Manipulation
Complete Mission	Impact	Data Manipulation	Runtime Data Manipulation

Figure 6: SPARQL Query Result for Entity Connections to a CVE Tag (CVE-2019-1943)

We can find the list of CVE vulnerabilities that are associated with a certain adversarial technique. In figure 2, we demonstrate such a query example. We can perform queries on the adversarial level information as well. For example, if we want to find certain tactics related to an advanced persistent threat stage or to find techniques belonging to a tactic, we can perform certain queries on the ontology as well. If we want to find certain products which might contain a certain vulnerability, we can get a list of those products using the query in figure 3. In figure 4, we provide query example for finding out product specifics so that the security analyst can see which version of a certain product is available in the system in order to take action as soon as a vulnerability that might be contained within the product, is discovered.

We might want to know given a CVE tag, which tactics and techniques might be connected to this via the ontological relationships e.g., we want to know which tactics, techniques, and sub-techniques are connected via a route or path to a particular CVE tag "CVE-2019-1943". In figure 5, we demonstrate this type of query. The query result(s) for this type of query is provided in figure 6.

3 USE CASE SCENARIOS

3.1 Use Case 1

The primary use case of our ontology is related to detecting a certain stage of an APT campaign. An attacker is to execute all the seven stages of an APT attack sequentially to complete the agenda of the attack in the very first place. If a certain ongoing stage is detected within a system, assuming that the very next stage/stages will be executed by the attacker at any minute, the possible way-outs can be stopped by patching certain vulnerabilities within the system. But due to the vast expansion of technology usage in our day-to-day life and also in enterprise and industrial practice, there lie thousands of vulnerabilities. Which one to patch without any significant delay is an important question to be asked in this particular situation. If we know that an attacker has initially compromised a system, the next thing he will try to do is to establish a foothold in the system. We list down the possible adversarial tactics, techniques, and sub-techniques that might be used by him from a behavioral perspective. Later, we provide the list of the vulnerabilities associated with these techniques and sub-techniques to be patched with immediate attention. An extension of the current ontology with the set of information that can potentially have the severity score of the CVEs, along with what software is prone to contain these particular vulnerabilities can also help the security analysts to take more prompt action.

4 CURRENT WORKS AND LIMITATIONS

Ontologies for network security have been proposed along with the discussion of its prospects. The need to develop new ontologies that relate to distinct aspects of network security, thereby facilitating management tasks was proposed in this work [7]. A more developed and detailed ontology to cover both system and network-level prospects is necessary to gain more coverage of available handy information. There has been ontology-based work in the domain of the Internet of Things (IoT) as well [5]. This is also the issue of being only centered on a specific concept and not giving vast knowledge about APT stages.

An ontology for threat intelligence has been proposed [2] which has the end goal of a system that helps threat intelligence analysts effectively organize and search both open-source intelligence and threat indicators in order to build a comprehensive picture of the threat environment. This type of work gives more of a behavioral perspective rather than technical details which are necessary to stop an ongoing campaign.

5 FUTURE WORK

In future work, we plan to incorporate all necessary data properties to the ontology relating to the details of the CVEs (e.g., which products have been reported to have such vulnerabilities), and what are the severity scores for the vulnerabilities so that the CVEs can be patched in ranked fashion. We will also incorporate defensive countermeasures information from MITRE's 'D3FEND' framework so that associated countermeasures with attack techniques and sub-techniques can be inspected by security analysts to find appropriate analysis tools. We also plan to add CVE as an entity to provide the security analyst with information regarding a specific vulnerability which might help to eradicate the software bringing this into the picture right away, or even providing proper patch information for that software (e.g., using a patched version of an application).

6 CONCLUSION

It is well established that curated knowledge in the field of cybersecurity is essential. As attack incidents require prompt actions with significant effect, proper knowledge management is important for the guidance to be provided to security analysts. Any individual or organization is likely to lessen the damage of any cyber attacks and to ensure that proper information dispersion is the first and foremost thing to be considered important. Our ontology enables the proper curation of knowledge for aiding security analysts in stopping the further spread of an ongoing cyber attack.

REFERENCES

[1] CVEDetails: The ultimate security vulnerability data source. https://www.cvedetails.com (2023)
[2] Falk, C.: An ontology for threat intelligence (07 2016)
[3] Loshin, P.: Resource description framework (rdf). https://www.techtarget.com/searchapparchitecture/definition/Resource-Description-Framework-RDF (2022)
[4] MITRE: Mitre att&ck framework. https://attack.mitre.org (2023)
[5] Mozzaquatro, B.A., Agostinho, C., Goncalves, D., Martins, J., Jardim-Goncalves1, R.: "an ontology-based cybersecurity framework for the internet of things. Sensors (Basel, Switzerland) **18(9): 3053** (2017). https://doi.org/10.3390/s18093053
[6] Ontology: What is an ontology? http://www-ksl.stanford.edu/kst/what-is-an-ontology.html (2023)
[7] Velasco, D., Rodriguez, G.D.: Ontologies for network security and future challenges. ArXiv **abs/1704.02441** (2017)