Interactive Visualization Dashboard for Common Attack Pattern Enumeration Classification

Mounika Vanamala
Dept. of Computer Science
University of Wisconsin Eau Claire
Rice Lake, WI, USA
vanamalm@uwec.edu

Walter Smith
Dept. of Computer Science
North Carolina Agricultural and Technical State Univ.
Greensboro, NC, USA
xhyuan@ncat.edu

Xiaohong Yuan
Dept.of Computer Science
North Carolina Agricultural and Technical State Univ.
Greensboro, NC, USA
wsmith12@aggies.ncat.edu

Joi Bennett
Dept. of Computer Science
North Carolina Agricultural and Technical State Univ.
Greensboro, NC, USA
jbbennett1@aggies.ncat.edu

***Abstract*— Attack patterns represent computer attackers' tools, methodologies, and perspective. The Common Attack Pattern Enumeration Classification (CAPEC) provides information about attack patterns which include descriptive textual fields, relationships between different attack patterns, execution flow, mitigations and related Common Weaknesses Enumeration (CWE) weakness and external Mapping. This paper describes an interactive visualization dashboard we developed for displaying the hierarchically structured CAPEC information. The dashboard includes a tree map and a network graph. The tree map visualization displays the hierarchy of CAPEC in a rectangular region in a space-filling manner. The network graph displays the parent child-taxonomy from the CAPEC using nodes and links between nodes. The visualization dashboard displays the external mapping of CAPEC to CWE, Adversarial tactics, techniques, and common knowledge (ATT&CK), The Open Web Application Security Project (OWASP) and Web Application Security Consortium (WASC) taxonomy. This visualization tool improves usability and provides a range of new capabilities for understanding and interacting with the rich content and relationships in CAPEC.**

Keywords-Attack patternss; Common Attack Pattern Enumeration and Classification (CAPEC); visualization, Network graph, tree map.

I. INTRODUCTION

To understand the methods for hacking information systems, cyber security experts must consider the attacker's point of view. The Common Attack Pattern Enumeration and Classification (CAPEC) provides a publicly available catalog of common attack patterns that helps users understand how adversaries exploit weaknesses in applications and other cyber-enabled capabilities [1].

CAPEC provides descriptive textual fields, also called elements, for each attack pattern. The current CAPEC release includes a list of 572 specific attack patterns. Each attack pattern may include up to 30 elements to describe the attack details. While CAPEC has addressed the need to create a standard for representing and defining attacks from an attacker's perspective, issues of usability arise because CAPEC does not provide a consistent level of documentation for some elements among the attack patterns. In many cases, attack pattern elements are missing completely. The inconsistent use of elements makes it problematic to discern the relationship, if any, between the descriptive fields. The goal of this research is to improve usability and provide a range of new capabilities for understanding and interacting with the rich content and relationships in CAPEC. Effective visualization helps users analyze and reason about data and evidence. It makes complex data more accessible, understandable, and usable. The goal is to communicate information clearly and efficiently to users. Visualization helps developers to better comprehend large and complex systems [2].

Navigation of CAPEC web content relies on following parent-child hyperlinks in textual content. This makes it difficult to comprehend CAPEC's overall hierarchical structure. To address this, we introduce a new web-based interactive visualization dashboard for CAPEC attack patterns. The visualization includes tree maps consisting of rectangles that represent the hierarchy of a system, and network graph. We describe tree map and network graph visualizations to visualize the parent-child taxonomy of CAPEC. CAPEC attack patterns are mapped onto external databases such as CWE, ATT&CK, OWASP and WASC. We describe how external mappings are displayed in this dashboard. This dashboard can be used by software engineers and security engineers who make use of CAPEC attack patterns for secure software development or other security activities.

The rest of the paper is organized as follows. In Section 2, we discuss the background of CAPEC, visualization techniques and implementation platform. Section 3 describes the visualization dashboard, and Section 4 discusses related work. Section 5 concludes the paper.

ISBN: 978-1-61208-997-3

II. BACKGROUND

This section provides background on CAPEC, tree map, network graph as well as Dash and Heroku we used to implement the visualization.

A. CAPEC

CAPEC attack patterns help people understand how weaknesses could be exploited and ways to defend against the weakness. CAPEC attack patterns are organized with parent and child hierarchy. For example, CAPEC 122 “Privilege Abuse” is a top level attack pattern with five child attack patterns: CAPEC-1 “Accessing Functionality Not Properly”, CAPEC-17 “Using Malicious Files”, CAPEC-180 “Exploiting Incorrectly Configured Access”, CAPEC-221 “WebView Exposure”, and CAPEC 503 “Data Serialization External Entities Blowup”. Each of these attack patterns has several child attack patterns, for example, “CAPEC-58 Restful Privilege Elevation” is the child attack pattern of “CAPEC-1 Accessing Functionality Not Properly”. Fig. 1 shows an example of the hierarchical parent child relationship in CAPEC attack patterns.

B. Tree Map

Trees maps are one of the most used structures to visualize relational information. Tree maps use a rectangular space-filling layout [3]. Space-filling techniques make maximal use of the display space. The two most common approaches to generating space-filling hierarchies are rectangular and radial layouts. In the basic tree map, a rectangle is recursively divided into slices, alternating horizontal and vertical slicing, based on the populations of the subtrees at a given level. There are many variations of tree maps such as square tree maps [4], and nested tree maps which are used to emphasize the hierarchical structure. These methods are structured using horizontal and vertical divisions to convey the hierarchy. For these and other space-filling techniques, color can be used to convey any attributes, such as a value associated with the node (e.g., classification) or it may display the hierarchical relationships, e.g., siblings and parents may have similarities in color. Symbols and other markings may also be embedded in the rectangular or circular segments to communicate other data features.

In additional rectangular layout, one way to pack a tree display into a smaller space is to employ a radial layout [2]. Also, there is a class of visualization techniques that represents tree relationships implicitly [5], rather than explicitly drawing vertices and edges.

C. Network Graph

The most common representation used to visualize trees or hierarchical relationships is a node-link diagram. This is the most popular form of non-space filling layout. This type of visualization shows how things are interconnected using nodes / vertices and link lines to represent their connections and help illuminate the type of relationships between a group of entities. In network graphs, not all of the nodes and links are created equally: additional variables can be visualized, for example, by making the node size or link stroke weight proportion to an assigned value. By mapping out connected systems, network graphs can be used to interpret the structure of a network through looking for any clustering of the nodes, how densely nodes are connected or by how the diagram layout is arranged [3].

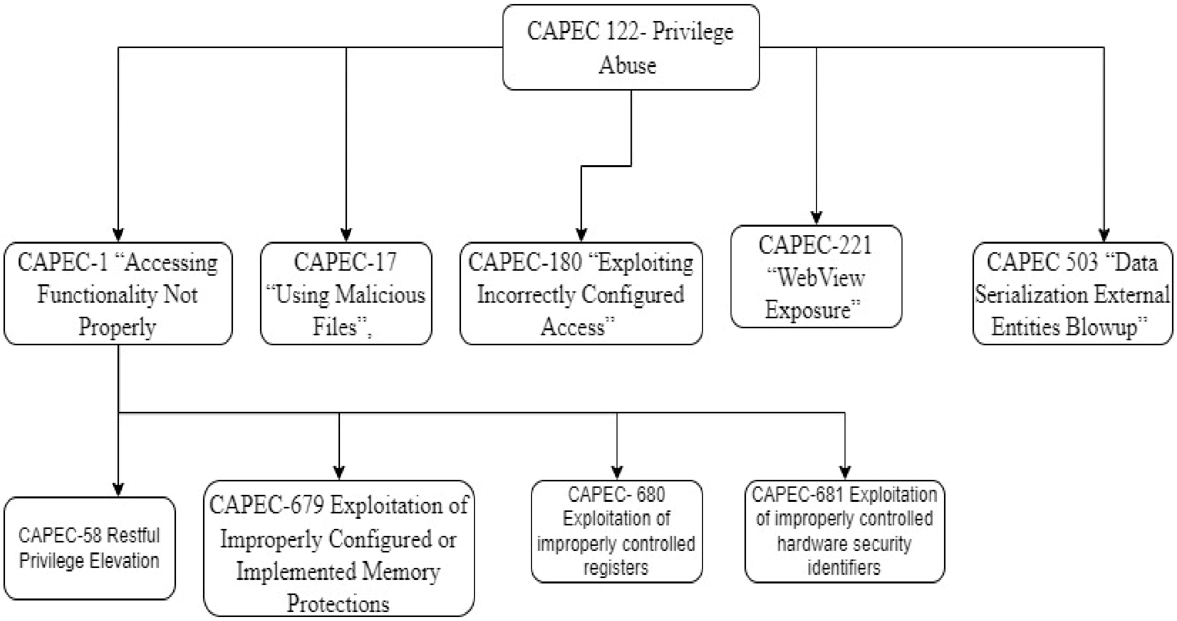

CAPEC 122- Privilege Abuse
CAPEC-1 “Accessing Functionality Not Properly
CAPEC-17 “Using Malicious Files”,
CAPEC-180 “Exploiting Incorrectly Configured Access”
CAPEC-221 “WebView Exposure”
CAPEC 503 “Data Serialization External Entities Blowup”
CAPEC-58 Restful Privilege Elevation
CAPEC-679 Exploitation of Improperly Configured or Implemented Memory Protections
CAPEC- 680 Exploitation of improperly controlled registers
CAPEC-681 Exploitation of improperly controlled hardware security identifiers

Fig. 1. The hierarchical parent-child relationship in CAPEC attack patterns

D. Dash

Plotly's dash library [6][7] provides a declarative Python interface for developing full-stack web applications ("Dash apps"). In addition to the main dash library, the dash-html-components and dash-core-components packages comprise the building blocks of a Dash app. dash-html-components provides an interface for building the layout of a Dash application that mimics the process of building the layout of a website; dash-core-components is a suite of common tools used for interactions with a Dash app (e.g., dropdowns, text inputs, and sliders) and includes Dash Core Components (DCC). Graph components for interactive graphs are made with plotly.py.

Interactivity is implemented with callbacks. These allow for reading the values of inputs in the Dash app (e.g., text inputs, dropdowns, and sliders), which can subsequently be used to compute the value of one or more "outputs", i.e., properties of other components in the app. The function that computes the outputs is wrapped in a decorator that specifies the aforementioned inputs and outputs; together, they form a callback. The callback is triggered whenever one of the specified inputs changes in value.

E. Heroku

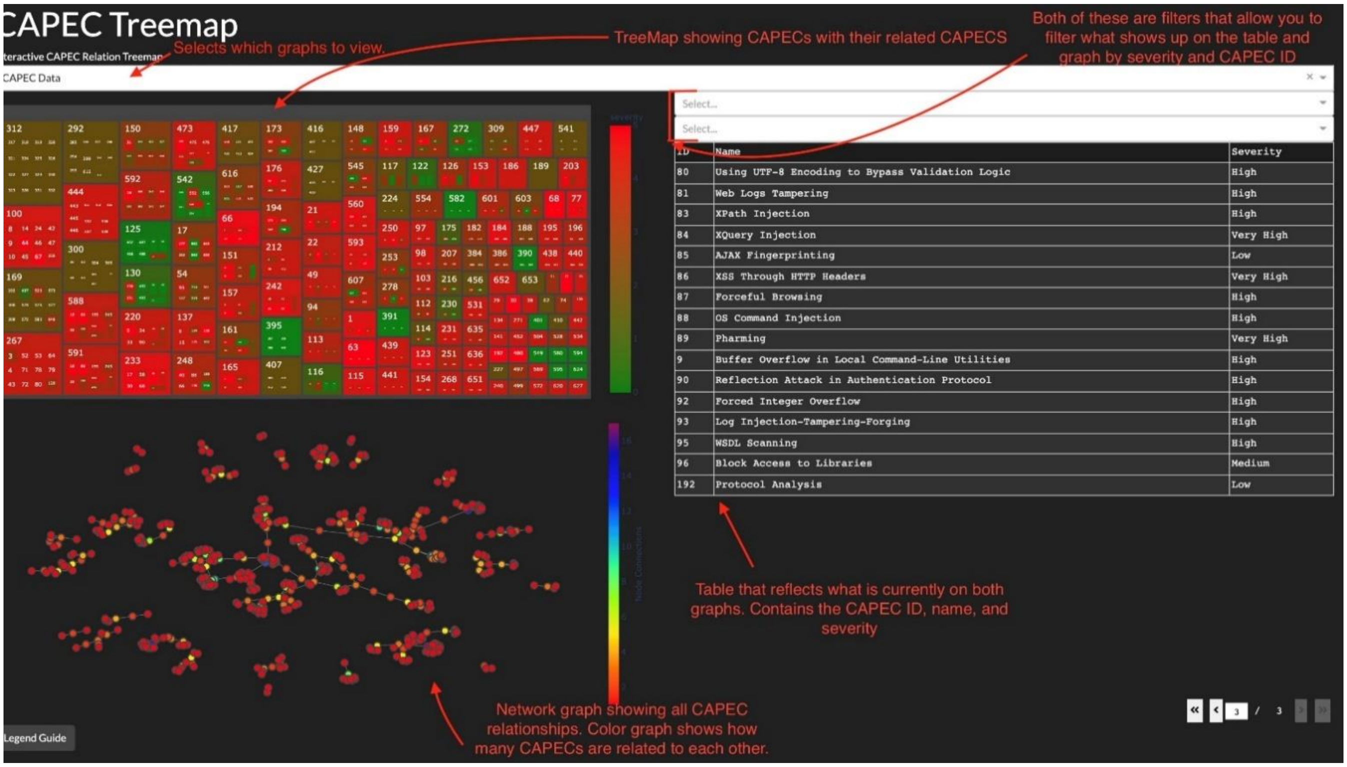
Heroku is a container-based cloud Platform as a Service (PaaS) [8]. It is flexible, easy to use, and offers developers a simple path to getting their apps to market. Heroku is open and extensible, meaning that developers can build any language they desire to use such as: Nodejs, Ruby, PHP, Python, Java and so on. For this research, we use Heroku because of its benefit of security and performance and the large quantity of add-ons available.

III. THE CAPEC VISUALIZATION DASHBOARD

This section describes the visualization dashboard for CAPEC attack patterns. The Visualization Dashboard presents different visualization techniques for navigating the CAPEC taxonomy. Our interactive visualization interfaces including tree maps and network graphs are implemented using Dash, Plotly, Heroku, and MongoDB. The data obtained from MITRE'S CAPEC website in the form of a CSV is hosted on a MongoDB database. The user can choose to visualize the CAPEC attack pattern data or CAPEC attack pattern external mapping.

A. Visualization of the CAPEC Data

Fig. 2 shows the visualization of CAPEC Data. There is a selection menu at the left top from which the user can choose what graph to display: CAPEC Data or External Mapping. The left top part of the visualization is a tree map created using Plotly. Given the data, a CSV is created with the columns parent, child, and severity, which refers to the parent attack pattern ID, the child attack pattern ID, and the severity of the parent attack pattern. The tree map shows the parent attack pattern ID with its child attack patterns within the rectangular space of the parent attack patten. The colors in the tree map are chosen based on the severity with a key on the right of the graph showing the color scale for the severity. If a user hovers over them, they can get the exact numerical value for severity.


CAPEC Treemap
Interactive CAPEC Relation Treemap
Selects which graphs to view.
TreeMap showing CAPECs with their related CAPECS
Both of these are filters that allow you to filter what shows up on the table and graph by severity and CAPEC ID
CAPEC Data
Select...
Select...
312
292
150
473
417
173
416
148
159
167
272
309
447
541
176
427
545
117
122
126
153
186
189
203
592
542
616
444
224
554
582
601
603
68
77
100
66
194
21
560
125
17
250
97
175
182
184
188
195
196
300
212
22
593
151
253
98
207
384
386
390
438
440
169
130
54
49
607
278
103
216
456
652
653
588
157
242
112
230
531
220
137
94
1
391
161
395
114
231
635
267
113
63
439
591
233
248
165
407
123
251
636
116
115
441
154
268
651
severity
Table that reflects what is currently on both graphs. Contains the CAPEC ID, name, and severity
Network graph showing all CAPEC relationships. Color graph shows how many CAPECs are related to each other.
Legend Guide
3 / 3

ID	Name	Severity
80	Using UTF-8 Encoding to Bypass Validation Logic	High
81	Web Logs Tampering	High
83	XPath Injection	High
84	XQuery Injection	Very High
85	AJAX Fingerprinting	Low
86	XSS Through HTTP Headers	Very High
87	Forceful Browsing	High
88	OS Command Injection	High
89	Pharming	Very High
9	Buffer Overflow in Local Command-Line Utilities	High
90	Reflection Attack in Authentication Protocol	High
92	Forced Integer Overflow	High
93	Log Injection-Tampering-Forging	High
95	WSDL Scanning	High
96	Block Access to Libraries	Medium
192	Protocol Analysis	Low

Fig. 2. Tree map and network graph to visualize CAPEC data

Copyright (c) IARIA, 2023. ISBN: 978-1-61208-997-3

Below the tree map in Fig. 2 is a network graph showing how a CAPEC attack pattern is related to other CAPEC attack patterns. Each note represents an attack pattern. The link from one node to another node shows the parent/child relationship between the two nodes linked together. The color of the node indicates the number of nodes connected to this node. The color scale to the right side of the network graph shows a purple color, indicating higher number of connected nodes, and a red color indicating lower number of connected nodes.

The right side of Fig. 2 shows a table that lists all the CAPEC attack patterns shown on the tree map and the network graph. It displays the CAPEC ID, the name of the CAPEC attack pattern, and the severity of the attack pattern. On the top of the table, there are two drop down menus allowing users to select the CAPEC attack pattern IDs, and a severity value to filter out specific CAPEC attack patterns or attack patterns of a selected severity level to display in the tree map, network graph and the table.

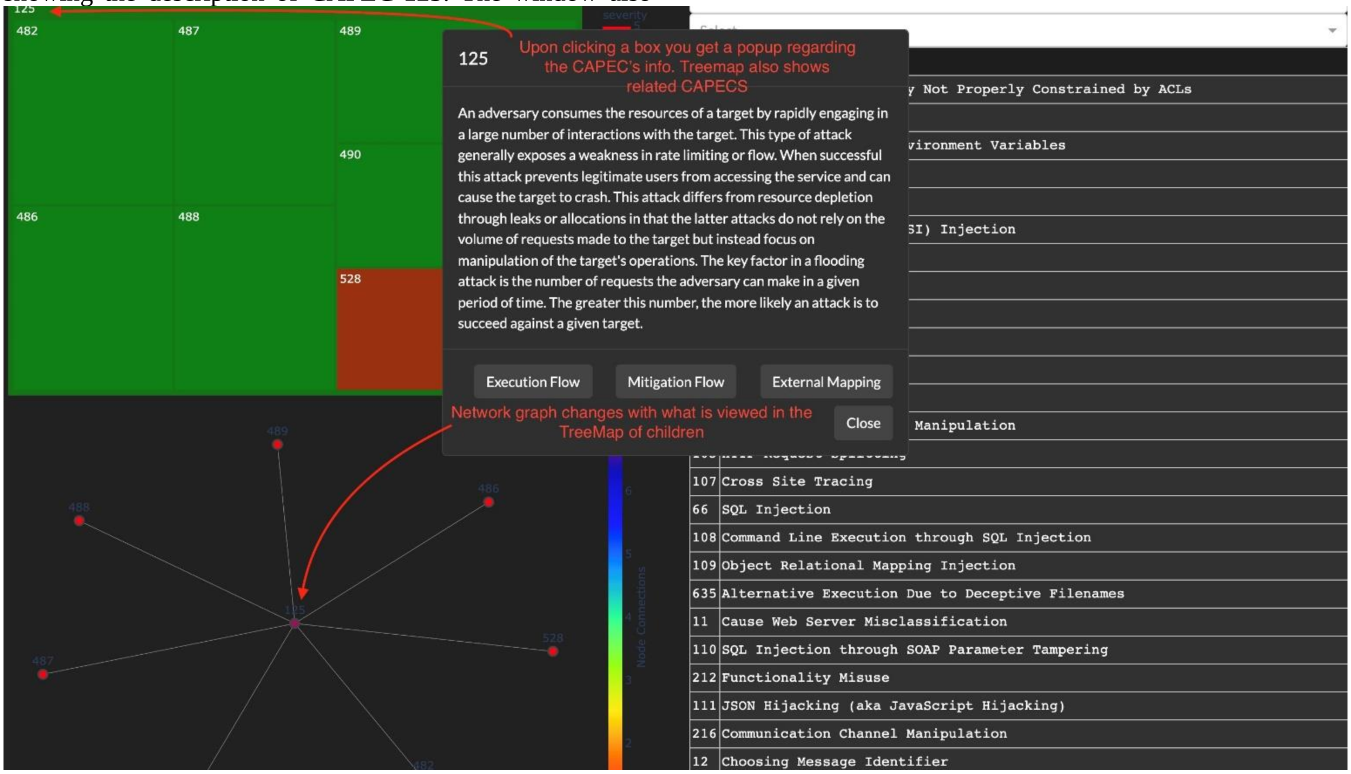
From Fig. 2, we can see that tree map effectively utilizes the space it takes to display hierarchical relationship. In the network graph, there is a lot of empty space and a lot of clusters.

The Visualization dashboard allows users to select a CAPEC ID from the tree map to view detailed information of the CAPEC attack pattern. For example, in Fig. 3, if a user clicks on CAPEC-125 in the tree map, a window will pop up showing the description of CAPEC-125. The window also includes buttons "taxonomy", "execution flow", and "mitigation flow". Clicking on each button will provide the corresponding information for CAPEC-125. The network graph changes correspondingly, showing only CAPEC-125 and all the child attack patterns connected to CAPEC-125.

B. *Visualization of the CAPEC External Mapping*

When users select external mapping from the dropdown menu on top of the tree map, the visualization dashboard will show the external mappings of CAPEC attack patterns. The CAPEC attack patterns are mapped to ATT&K [9], OWASP [10], WASC [11] and CWE weaknesses [12]. Fig. 4 shows the attack patterns that are mapped to these taxonomies. In Fig. 4, the left-top rectangle in blue color shows the CAPEC IDs that are mapped to the ATT&K taxonomy. The left-bottom rectangle in orange color shows the CAPEC IDs that are mapped to OWASP, and the right-bottom rectangle in green color shows the CAPEC IDs that are mapped to WASC. The right side of the dashboard is a table that shows the CAPEC ID and the related CWE weaknesses.

The user can select an CAPEC ID, and the dashboard will show the external mapping of the particular attack pattern. For example, in Fig. 5, CAPEC-168 is chosen. A popup window displays the CAPEC attack pattern ID:168, its related CWE weaknesses: 212, 69, and related ATT&CK. A hyperlink to the CAPEC page for this attack pattern is also given.



125

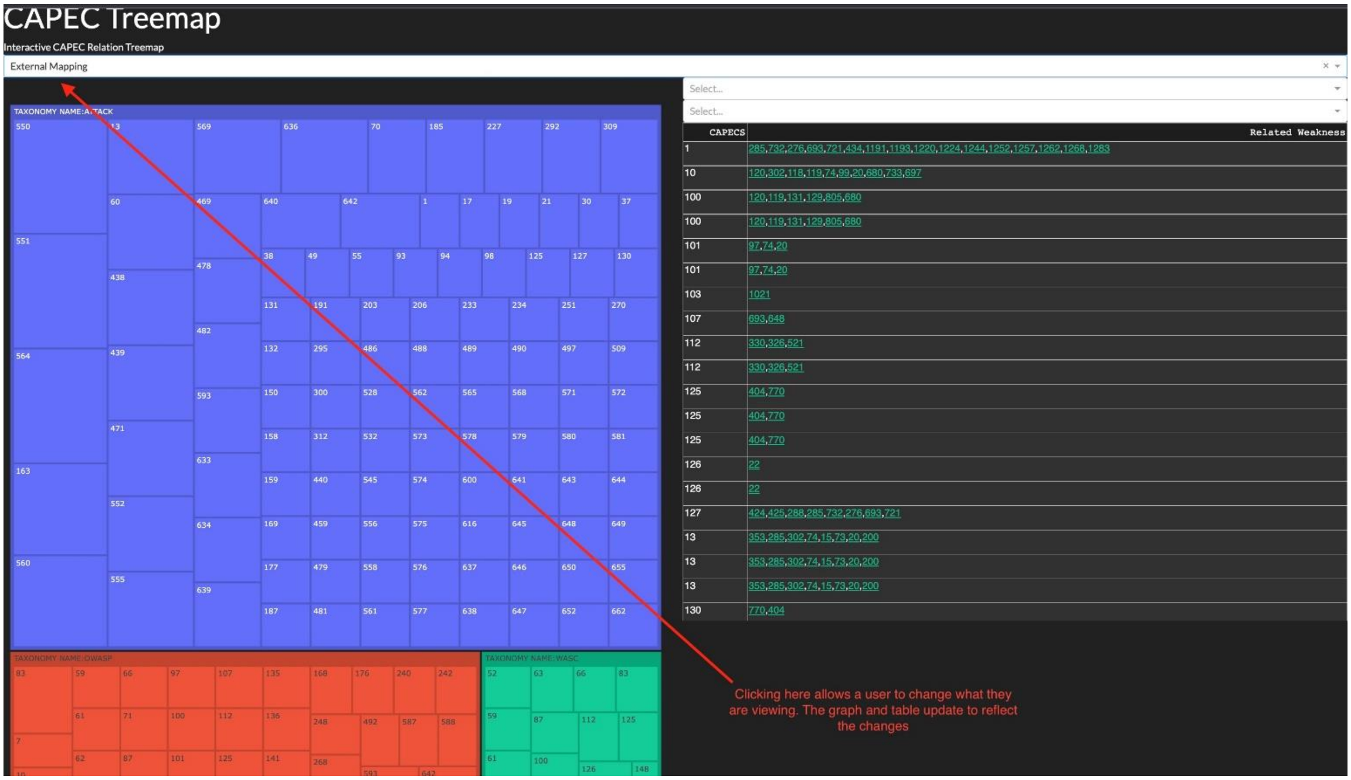
Upon clicking a box you get a popup regarding the CAPEC's info. Treemap also shows related CAPECS

An adversary consumes the resources of a target by rapidly engaging in a large number of interactions with the target. This type of attack generally exposes a weakness in rate limiting or flow. When successful this attack prevents legitimate users from accessing the service and can cause the target to crash. This attack differs from resource depletion through leaks or allocations in that the latter attacks do not rely on the volume of requests made to the target but instead focus on manipulation of the target's operations. The key factor in a flooding attack is the number of requests the adversary can make in a given period of time. The greater this number, the more likely an attack is to succeed against a given target.

Network graph changes with what is viewed in the TreeMap of children

	y Not Properly Constrained by ACLs
	vironment Variables
	SI) Injection
	Manipulation
107	Cross Site Tracing
66	SQL Injection
108	Command Line Execution through SQL Injection
109	Object Relational Mapping Injection
635	Alternative Execution Due to Deceptive Filenames
11	Cause Web Server Misclassification
110	SQL Injection through SOAP Parameter Tampering
212	Functionality Misuse
111	JSON Hijacking (aka JavaScript Hijacking)
216	Communication Channel Manipulation
12	Choosing Message Identifier

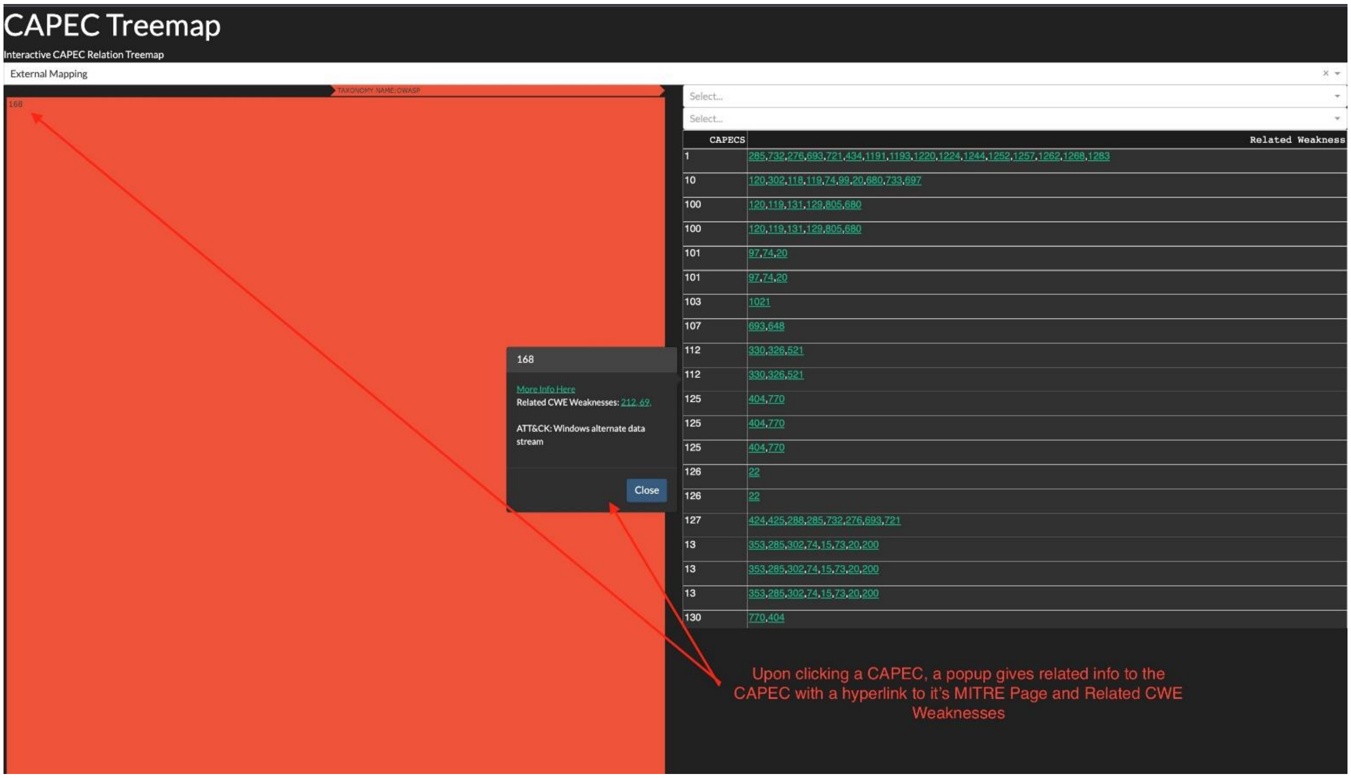
Fig. 3. Interaction between the tree map and network graph.



CAPECS	Related Weakness
1	285,732,276,693,721,434,1191,1193,1220,1224,1244,1252,1257,1262,1268,1283
10	120,302,118,119,74,99,20,680,733,697
100	120,119,131,129,805,680
100	120,119,131,129,805,680
101	97,74,20
101	97,74,20
103	1021
107	693,648
112	330,326,521
112	330,326,521
125	404,770
125	404,770
125	404,770
126	22
126	22
127	424,425,288,285,732,276,693,721
13	353,285,302,74,15,73,20,200
13	353,285,302,74,15,73,20,200
13	353,285,302,74,15,73,20,200
130	770,404

Clicking here allows a user to change what they are viewing. The graph and table update to reflect the changes

Fig. 4. External mappings of CAPEC attack patterns



168

More Info Here
Related CWE Weaknesses: 212, 69,

ATT&CK: Windows alternate data stream

CAPECS	Related Weakness
1	285,732,276,693,721,434,1191,1193,1220,1224,1244,1252,1257,1262,1268,1283
10	120,302,118,119,74,99,20,680,733,697
100	120,119,131,129,805,680
100	120,119,131,129,805,680
101	97,74,20
101	97,74,20
103	1021
107	693,648
112	330,326,521
112	330,326,521
125	404,770
125	404,770
125	404,770
126	22
126	22
127	424,425,288,285,732,276,693,721
13	353,285,302,74,15,73,20,200
13	353,285,302,74,15,73,20,200
13	353,285,302,74,15,73,20,200
130	770,404

Upon clicking a CAPEC, a popup gives related info to the CAPEC with a hyperlink to it's MITRE Page and Related CWE Weaknesses

Fig. 5. Pop up window to display external mapping information of a CAPEC attack pattern

IV. Related work

Existing information visualization techniques are usually limited to the display of a few thousand items or avoid the problem of visualizing large number of items by using aggregation, sampling, and extracting, or by not managing occlusion and overlapping [13]. Among the popular techniques is the use of scatter plots connected to interactive controls such as in Dynamic Queries. We have used the tree visualization in our dashboard to show the parent child taxonomy in CAPEC.

Noel [14] visualized the overall hierarchical structure of CAPEC attack patterns using network graph, Sunburst visualization, Circular tree map, and Voronoi tree map. They also conducted text mining and computed hierarchical clusters and grouped related attack patterns through automated analysis. They used bipartite graph to visualize cross references from CAPEC attack patterns to CWE weaknesses. Our tool used rectangular tree map and network graph to visualize the hierarchical structure of CAPEC attack patterns and display external mapping information including CWE, ATT&CK, OWASP and WASC. The goal of our tool is to allow users to retrieve such information easily.

Regainia and Salva [15] proposed a methodology that takes as inputs CAPEC attack patterns, and infers relationships between attacks, weaknesses, security principles and patterns to generate the classification and Attack Defense Trees. Seehusen [16] proposed to use CAPEC for Risk-Based Security Testing. Vanamala et al. [17] and Vanamala et al. [18] have used software repositories like CVE (Common Vulnerabilities and Exposures) and CAPEC to help develop secure software. They have developed a recommender system that recommends attack patterns relevant to the system under development based on software requirements documents. A software developer can develop security requirements and secure design based on these attack patterns [19].

V. Conclusion and Future Work

In this paper, we describe a web-based interactive visualization dashboard for the CAPEC attack patterns. This includes visualizing the overall hierarchical structure of CAPEC attack patterns using tree map and network graph. Our visualization techniques provide a range of new capabilities for understanding and interacting with the rich content and relationships in CAPEC. The goal is to be able to effectively communicate this information to a user in a user-friendly way. This tool will help users to make use of CAPEC attack patterns in developing secure software or conducting other security activities such as threat modeling, security testing, training, and education. Our future work is to conduct a user study for the dashboard to assess the effectiveness of the tool in helping users understand the structure of CAPEC and to make use of CAPEC attack patterns in their security tasks.

Acknowledgment

This work is partially supported by the National Center of Academic Excellence in Cybersecurity (NCAE-C) under the grant H98230-20-1-0404. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsor.

References

[1] The Common Attack Pattern Enumeration and Classification. [Online]. Available from: https://capec.mitre.org/ [last accessed on 2022.10.12]

[2] M. Balzer, A. Noack, O. Deussen, and C. Lewerentz, “Software landscapes: visualizing the structure of large software systems,” IEEE TCVG, 2004. doi: 10.2312/VisSym/VisSym04/261-266

[3] B. Johnson, “TreeViz: treemap visualization of hierarchically structured information,” Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, June 1992, pp. 369-370.

[4] W. Scheibel, M. Trapp, D. Limberger, and J. Döllner, “A taxonomy of tree map visualization techniques” VISIGRAPP (3: IVAPP), 2020, pp. 273-280.

[5] R. Vliegen, J. J. Van Wijk, and E. J. van der Linden, “Visualizing business data with generalized treemaps,” IEEE Transactions on visualization and computer graphics, vol. 12, no. 5, pp. 789-796, 2006.

[6] I. Stančin and A. Jović, “An overview and comparison of free Python libraries for data mining and big data analysis,” The 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), May 2019, pp. 977-982.

[7] Plotly. "Introduction to Dash". [Online]. Available from: https://dash.plotly.com/introduction [last accessed on 2022.10.12]

[8] Heroku. [Online]. Available from: https://www.heroku.com/home [last accessed on 2022.10.12]

[9] MITRE ATT&CK. [Online]. Available from: https://attack.mitre.org/ [last accessed on 2022.10.12]

[10] OWASP. [Online]. Available from: https://owasp.org/ [last accessed on 2022.10.22]

[11] The Web Application Security Consortium (WASC). Threat Classification. [Online]. Available from: http://projects.webappsec.org/w/page/13246978/Threat%20Classification [last accessed on 2022.10.12]

[12] Common Weakness Enumeration. [Online]. Available from. https://cwe.mitre.org/ [last accessed on 2022.10.12]

[13] J. D. Fekete and C. Plaisant, “Interactive information visualization of a million items,” The IEEE Symposium on Information Visualization, (INFOVIS 2002), 2002, pp. 117-124.

[14] S. Noel, “Interactive visualization and text mining for the CAPEC cyber-attack catalog,” Proceedings of the ACM Intelligent User Interfaces Workshop on Visual Text Analytics, 2015. pp. 1-8.

[15] L. Regainia and S. Salva, “A Methodology of security pattern classification and of attack-defense tree generation,” The 3rd International Conference on Information Systems Security and Privacy, 2017, pp. 136-146, doi:10.5220/0006198301360146.

[16] F. Seehusen, “Using CAPEC for risk-based security testing,” International Workshop on Risk Assessment and Risk-driven Testing, 2015, Springer, Cham, pp. 77-92.

[17] M. Vanamala, X. Yuan and K. Roy, “Topic modeling and classification of common vulnerabilities and exposures database,” 2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD 2020), IEEE, 2020, pp. 1-5. doi: 10.1109/icABCD49160.2020.9183814

[18] M. Vanamala, X. Yuan and K. Bandaru, “Analyzing CVE database using unsupervised Topic Modelling," 2019 International Conference on Computational Science and Computational Intelligence (CSCI), 2019, pp. 72-77, doi: 10.1109/CSCI49370.2019.00019.

[19] M. Vanamala, J. Gilmore, X. Yuan and K. Roy, “Recommending attack patterns for software requirements document,” The 2020 International Conference on Computational Science and Computational Intelligence (CSCI), 2020, pp. 1813-1818, doi: 10.1109/CSCI51800.2020.00334.