Visualization Dashboard for Recommending Attack Patterns Using Topic Modeling

Jairen Gilmore
Dept. of Computer Science, North Carolina Agricultural and Technical State UniversityEmail: jgilmore@aggies.ncat.edu

Uriah Moore
Dept. of Computer Science, North Carolina Agricultural and Technical State UniversityEmail: umoore1@aggies.ncat.edu

Xiaohong Yuan
Dept. of Computer Science, North Carolina Agricultural and Technical State UniversityEmail: xhyuan@ncat.edu

Taylor Headen
Dept. of Computer Science, North Carolina Agricultural and Technical State UniversityEmail: theaden@aggies.ncat.edu

Mounika Vanamala
Dept. of Computer Science, University of Wisconsin Eau ClaireEmail: vanamalm@uwec.edu

ABSTRACT

The Common Attack Pattern Enumeration and Classification (CAPEC) database is a great resource that software developers can consult to gain a deeper understanding of techniques used by attackers to exploit the vulnerabilities of a software. Attack patterns define the steps and perquisites needed for an attacker to exploit a system and include mitigations for the given attack pattern. However, the time required to manually search and find relevant attack patterns can quickly become a time-consuming process. We utilize topic modeling to recommend attack patterns relevant to software requirement specifications. This paper introduces a visualization dashboard for recommending CAPEC attack patterns using topic modeling. The tool allows the user to upload a software requirements specification document and select the topic modeling algorithm. The most relevant attack patterns will be returned to the user through visualization. This tool helps developers make use of CAPEC attack pattern efficiently.

CCS CONCEPTS

• **Software and its engineering**; • **Software creation and management**; • **Designing software**;

KEYWORDS

Topic Modeling, Attack Pattern, Visualization, Dashboard

ACM Reference Format:
Jairen Gilmore, Uriah Moore, Xiaohong Yuan, Taylor Headen, and Mounika Vanamala. 2023. Visualization Dashboard for Recommending Attack Patterns Using Topic Modeling. In *2023 12th International Conference on Software and Information Engineering (ICSIE) (ICSIE 2023), November 21–23, 2023, Sharm El-Sheikh, Egypt*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3634848.3634862

ICSIE 2023, November 21–23, 2023, Sharm El-Sheikh, Egypt

ACM ISBN 979-8-4007-0810-7/23/11…$15.00
https://doi.org/10.1145/3634848.3634862

1 INTRODUCTION

In the process of secure software development, developers may create abuse cases that describe how a system can be misused. Developers may consult resources such as the Common Attack Pattern Enumeration and Classification (CAPEC) database which contains 559 attack patterns. [11] These attack patterns are a set of methods and characteristics that an adversary can employ to exploit various physical and digital systems. Included in an attack pattern entry are its description, prerequisites, mitigations, severity, likelihood and other characteristics. Users can search for attack patterns on the CAPEC website using a specific CAPEC ID, a keyword, or by mechanisms of attack or domains of attack. The attack patterns relevant to the software system that is being developed can be used by the software developers to create abuse or misuse cases. However, finding attack patterns that are relevant for a given software manually is a time-consuming task.

We propose a tool for recommending attack patterns using topic modeling (TrapTM) to recommend relevant CAPEC attack patterns through topic modeling and visualization. We use the Latent Dirichlet Allocation (LDA) and Latent Semantic Analysis (LSA) topic models to recommend attack patterns. Users can select from the visualization dashboard to use either the LDA or LSA topic models and get the results of the recommendation. Cosine similarity is used to measure the similarity between attack patterns and the topics from the software requirement specification. The visualization shows the cosine similarity values of the attack patterns to the software requirements specification document user uploaded in a bar chart, displays the recommended attack patterns in a table, and in a word cloud format. This visualization dashboard is easy to use and provides the information contained in the recommended attack patterns in CAPEC to a developer that will help them develop secure software.

2 BACKGROUND

This section will provide some background on the technologies used to create the recommendations. the LDA and LSA topic models, and Dash which is used to create the visualization components of the tool. For this section the LDA and LSA models will be introduced with a description of the topic model and how it was implemented

in the application. Followed by the topic models, will be a section related to DASH a Plotly Python library that describes the library and the reasons it was used in the application. The final component to be discussed is the hosting solution Render and why it is utilized.

2.1 Latent Dirichlet Allocation

The LDA model is a generative probabilistic model that assumes that documents consist of a set of topics, these topics consist of a set of keywords, and both sets have Dirichlet Distributions. [2] By changing the number of topics, the distribution of keywords of a topic changes. Python library Gensim is used to implement the LDA model. The implementation in this library allows for fine tuning of characteristics such as the number of topics, alpha which affects the document-topic distribution, and eta which affects the topic-word distribution. This implementation also allows for the keywords to be collected from various topics.

2.2 Latent Semantic Analysis

The LSA model works by taking a given document and converting it into a matrix and then breaking it up through a process known as singular value decomposition (SVD) to break down the document matrix into a product of three matrices.[3] It is these three matrices that represent the topic model. Of the three matrices, one of the matrices represents the topics contained within document, another matrix represents the terms contained in the topic and the third matrix is a diagonal matrix. The LSA implementation also comes from the Gensim library and allows for parameter tuning, namely allowing for the number of topics to be specified.

2.3 Dash

Dash is a framework that allows for the development of web applications in Python. [7] Instead of having to develop a website using HTML by using the Dash HTML Components, it becomes possible to develop a web application with HTML elements in python. The Dash Core Components would allow for stylization of the Dash application in python. Callbacks in Dash allow for the user to interact with the web page by taking in user input and producing an output depending on what the function is calling for. The user input could come from a drop- down menu or numeric value and this would update what is being displayed on the web application.

2.4 Render

Render is an online site where users can host their web applications on an online server. [10] This service was selected as it offers a free plan in the form of an individual plan for hosting. This feature was beneficial for testing on a small scale as the free plan offers 100GB of bandwidth per month. The render platform does offer scalable pricing based on what computer resources are needed for a web application. In addition to the free plan, render does offer a team plan with two highlights being the ability to add up to ten team members and 500GB of bandwidth. This is the hosting service that is currently being utilized for the visualization tool because of the free plan options and the ability to scale needed.

3 RELATED WORK

Vanamala et al. [8] and Stephens et al. [1] used the LDA topic model for CAPEC attack pattern recommendation. These work does use topic modeling for providing attack pattern recommendations through the use of LDA topic model. With the work by Vanamala et al. [8] utilized SRS documents for generating recommendations, the work by Stephens et al. [1] did not have such documentation available instead utilized other documentation. While the work by Vanamala et. Al. [8] does not visualize the recommended CAPEC attack patterns, a visualization of the LDA models is created through the python library PyLDAvis. This visualization shows when topics of the LDA model are overlapping with other topics, which can be helpful in fine tuning the LDA model and determining the optimum number of topics.

There has been work done towards making an interactive dashboard for CAPEC Attack Patterns. [9] This dashboard displayed CAPEC attack patterns by using a tree map and a network graph. This visualization dashboard helps users understand and interact with the content and relationships in CAPEC. This research utilized the python library Dash to create the web application and the cloud platform Heroku for hosting the application.

Another work utilized the CAPEC Attack Patterns for creating visualizations in the form of dendrograms to group CAPEC attack patterns based on the characteristics of the attack patterns. [6] The authors showcased that the visualization based on CAPEC characteristics provides a new way for understanding the relation of CAPEC Attack Patterns. While this work specifically created dendrograms based on the characteristics contained within CAPEC Attack Patterns, it highlights the potential that dendrograms can be used for the purposes of attack pattern recommendations. Such a system would be useful in helping users wanting to know what the correlation is between attack patterns recommendations.

Focusing on work related to word clouds and intractability, a work proposed a solution called VisGets. VisGets is a selection of widgets that organize information and can be used to build more complex ways of organizing data to suit the needs of the user. [4] The VisGets widgets were created around different information dimensions namely time, location, and keywords. These information dimensions provide different ways to process data and visualizations. Focusing specifically on the VisGet that worked with keywords, the authors utilized a version of the word cloud that sorted words in alphabetical order and the ability for the user to select specific words that would serve as a filter for the dataset. The ability to sort words featured in a word cloud is useful for cases where there are a multitude of words contained in a word cloud as sorting can help speed up a user looking up a specific term. The second feature is useful for narrowing down a selection of information. For the purpose of attack pattern recommendation, the benefit of having a word cloud with a filter can help in narrowing attacks, as it would allow for a user to select words that are common features across multiple attack patterns. For the purposes of text analysis, a word cloud is a component that can help in analyzing text by showing what words appear often in a selection of text by the size of word in the word cloud. However, there are other features of word clouds besides text size that can be utilized to create more informative. In the work by Heimerl et al. [5] the benefits of word

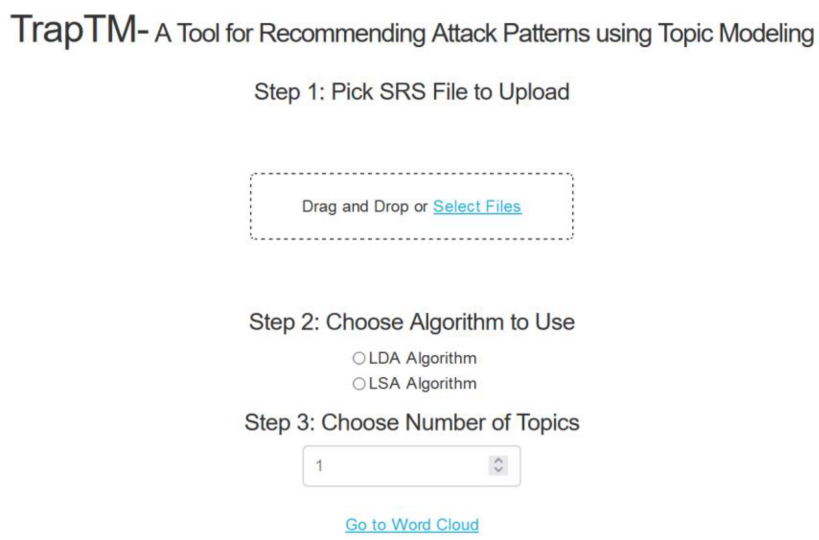

Figure 1: User interface for TrapTM web application

clouds used for the task of text analysis were shown. For this work, a system was developed to make an interactive word cloud and it was found that the participants of the study found the tool useful for the purpose of text analysis. The authors noted that participants found that such a system would better with tools that complement the functionality of the interactive word cloud. The features of this interactive word cloud utilize color, word size, and allowing the user to provide input in the form maximum number of terms and maximum font size.

4 VISUALIZATION DASHBOARD FOR RECOMMENDING CAPEC ATTACK PATTERNS

Discussed will be all the pages of the TrapTM application. The discussion will begin with section 4.1 discussing the landing page and a description of the different components utilized on the page will be given. Section 4.2 will discuss the background process of how the attack pattern recommendations are generated. Section 4.3 will discuss the recommendations page and the significance of the components displayed on that page. Section 4.4 ends with discussion of an interactive word cloud and its contribution to the TrapTM application.

4.1 Landing Page for TrapTM

Figure 1 shows the user interface of the landing page for the web application. This page breaks down the process of constructing recommendations into several steps. For the first step, the user will select the SRS document to be uploaded. The current implementation requires the SRS document provided to be in the portable document format (PDF). Upon uploading the document, the user will proceed to step two. For step two, the user will then select either the LDA or LSA algorithm as the topic model that will be used to generate the attack pattern recommendations. The third step will allow the user to specify the number of topics that the model will generate. Once the user has completed steps one through three, the user can click on the hyperlink below step three to proceed to the next page. Once the user has clicked to proceed to the next page, the application will begin to process the document submitted. The next section will discuss how the recommendations are generated.

4.2 Recommending CAPEC Attack Patterns using LDA and LSA

For both LDA and LSA algorithms, the following process performed is shown in Figure 2. After the document has been uploaded, the text preprocessing phase begins. For this phase, the text is extracted from the given document and data cleaning is performed on the text. This process involves converting text to lower case, and removing punctuation, excess white spaces, and numbers.

Next, the removal of stop words is performed using the Python library Natural Language Toolkit (NLTK). These words are referred to as stop words as they are words that appear often in sentences and these words do not contribute significantly to sentences. Stop word removal helps in reducing the size of the data set. This is useful as the removal of words such as ‘if’, ‘and’, and ‘the’ being removed from a document can help in ensuring that significant terms appear as top terms of a topic. The next step is to create bigrams and trigrams. After that lemmatization is performed which will reduce words down to their base forms. This text is then used as the corpus for the topic model. With the generated, the generation of the topic model phase begins.

For this phase, depending on the algorithm selected by the user, the corpus is given as input to either the LDA or LSA topic model along with the number of topics that the user had specified when uploading their documents. In addition to the user generated topic models, topic models are also created for each attack pattern in CAPEC where the selected number of topics is set to four topics.

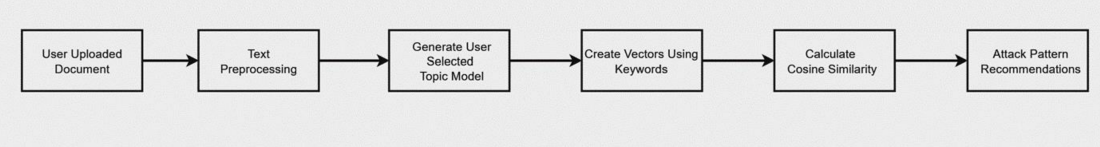

Figure 2: Attack pattern recommendation generation process overview

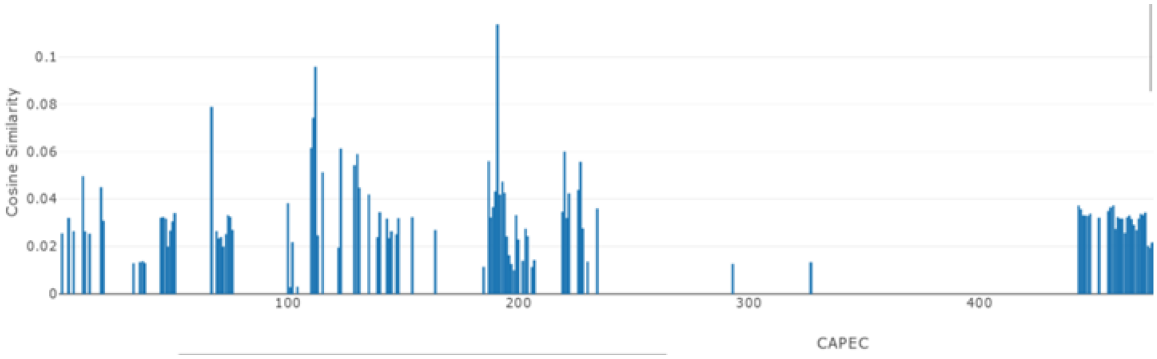

Figure 3: Bar Chart of CAPEC ID and their cosine similarity

Once the models are created, the next phase begins by obtaining top 30 keywords from each topic of the generated topic model and these keywords are used to create a vector. The same process is done for each attack pattern in CAPEC, the top 30 keywords from each of the four topics are put into a vector as well. Once these vectors are created, the next phase is to perform cosine similarity comparing the vector of the user generated topic model against all vectors of the attack Patterns in CAPEC. These attack patterns are then sorted from highest degree of similarity to lowest degree of similarity and are returned to the user. With the results sorted, the results page of the web application is ready to be displayed to the user.

4.3 Recommendations Page

On the page with the attack pattern recommendations, the first element displayed on screen is a bar graph displaying the relationship between the CAPEC ID's, and their cosine similarity scores, and this can be seen in Figure 3. The bar graph provides a general overview in the form of a visualization on how attack patterns compared to each other. The subsequent element displayed on screen is a table with the top ten most relevant attack pattern recommendations as shown in Figure 4

In addition to the bar graph, a table is provided that provides the details of the top ten CAPEC attack patterns recommended. As shown in Figure 4, the table shows the CAPEC ID, name of the attack pattern, description of the ID, the cosine similarity score of the attack, the severity of the attack, and the prerequisite for the attack pattern in relation to the SRS document that was uploaded.

This information was selected to be included as these fields provide developers with insight on what a given attack pattern is and what conditions need to be met for an attacker to perform the exploit. The CAPEC ID is provided as it can be used to retrieve more information about the attack pattern such as mitigations and related weaknesses on the CAPEC website. These two components provide an overview and detailed results such that a developer can use the recommendations to conduct further information gathering to develop more detailed abuse cases.

4.4 The Interactive Word Cloud

A word cloud is a helpful visualization that can provide a succinct summary of text. However, the Python library of word cloud is limited in functionality. The word cloud provided by the Python library provides a word cloud as images meaning that they are not interactable. Additionally, the only feature of the python word cloud library that portrays a characteristic of the dataset is the size of the words of word cloud. Where the size denotes the frequency of the word in the word cloud. Our implementation of word cloud uses the features of size, color, and making the word cloud interactable in form of mouse hover and mouse click.

The implementation of word cloud used for the application is portrayed in Figure 5. As shown in the figure, a scatter plot is used to plot the top recommendations. One feature of this word cloud is the numeric CAPEC IDs serve as points on the graph. Another feature is that the size of the CAPEC ID is used to denote the attack patterns likelihood to use an attack pattern. Where the larger the CAPEC ID is the more likely adversary is to take advantage of the attack pattern. Another feature utilized for this word cloud is the use of color. The color of the CAPEC ID is used to denote the severity of an attack pattern. As shown in Figure 5, a legend is given to the users providing specific meaning behind each color.

Another feature that is provided by this word cloud deals with making the word cloud interactable. One way this was done by utilizing mouse hovering over the CAPEC IDs. Doing so provides the user with the name of the attack pattern. Another interaction comes in the form of clicking on a given CAPEC ID. When clicking on a CAPEC ID, the user is given more information about an attack

CAPEC ID	Description	Cosine Similarity	Severity	Prerequisites
90	Reflection Attack in Authentication Pro…	0.129	High	::The attacker must have direct access to the target server in order to successfully …
97	Cryptanalysis	0.121	Very H…	::The target software utilizes some sort of cryptographic algorithm.::An underlying w…
77	Manipulating User-Controlled Variables	0.118	Very H…	::A variable consumed by the application server is exposed to the client.::A variable…
9	Buffer Overflow in Local Command-Line U…	0.113	High	::The target host exposes a command-line utility to the user.::The command-line utili…
112	Brute Force	0.111	High	::The attacker must be able to determine when they have successfully guessed the secr…
644	Use of Captured Hashes (Pass The Hash)	0.107	High	::The system/application is connected to the Windows domain.::The system/application …
111	JSON Hijacking (aka JavaScript Hijackin…	0.107	High	::JSON is used as a transport mechanism between the client and the server::The target…
94	Man in the Middle Attack	0.106	Very H…	::There are two components communicating with each other.::An attacker is able to ide…

Figure 4: Table containing top attack pattern recommendations

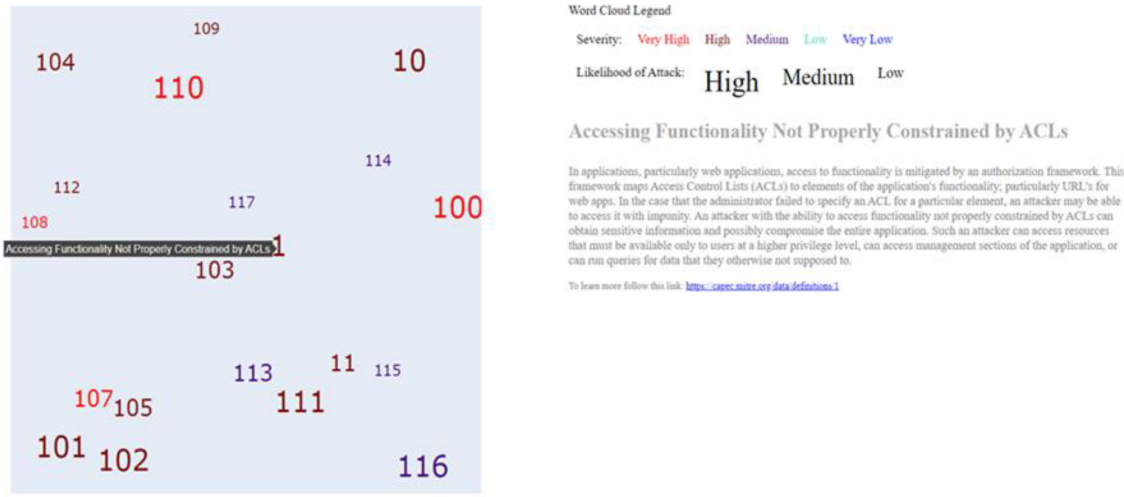

Figure 5: Word cloud web page

pattern. The attack pattern's description is provided along with a hyperlink to the CAPEC website that the user can click on to get more information regarding the specific attack pattern.

5 CONCLUSION AND FUTURE WORK

Discussed was a tool called TrapTM a tool for visualizing recommended attack patterns. The task of helping developers develop abuse cases was accomplished by this tool, a web application that utilizes a user's uploaded documents, a user selected topic model, and number of topics. Other pivotal points discussed were technologies utilized to create the application, the reasoning behind the selected technologies, description of the visualization dashboard's different screens and its components, and the significance the components play in the dashboard.

The future work consists of making improvements to this tool to further help developers. These plans include providing the ability for CAPEC attack patterns to be returned based on specific criteria, for example, only relevant attack patterns with high likelihoods or attack patterns with high severities are returned. Another potential improvement would be to have configurations tailored for specific tasks. This improvement would exist in the form of another step listing a question and a field for user input. The question being what a user's reason is for using the tool, and the user input in the form of a drop-down list with its elements being specific activities. Such activities could include performing risk assessment, wanting to develop mitigation strategies, and wanting to know how an attack could occur. Based on what option a user has selected, the table on the results screen would be adjusted to suit the specific activity the user had selected when uploading the SRS document.

For mitigations, the table would include the CAPEC attack pattern name and CAPEC ID, description, along with mitigations for the recommended attack pattern. If the option selected is to know how these attacks occur, the table would have the CAPEC attack pattern name and CAPEC ID, description, and prerequisites for the attack to occur. Tailoring the results in this manner would allow developers to look at and have access to only the information they determined to be relevant.

ACKNOWLEDGMENTS

This work is partially supported by NCAE 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] Stephen Adams, Bryan Carter, Cody Fleming, and Peter A Beling. 2018. Selecting System Specific Cybersecurity Attack Patterns Using Topic Modeling. In (2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)), 490–497. DOI:https://doi.org/10.1109/TrustCom/BigDataSE.2018.00076

[2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. 3, (2003), 993–1022.

[3] Scott Deerwester, Sustan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by Latent Semantic Analysis. 41, (1990), 391–407. DOI:https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9

[4] Marian Dörk, Sheelagh Carpendale, Christopher Collins, and Carey Williamson. 2008. VisGets: Coordinated Visualizations for Web-based Information Exploration and Discovery. 14, 6 (2008), 1205–1212. DOI:https://doi.org/10.1109/TVCG.2008.175

[5] Florian Heimerl, Steffen Lohmann, Simon Lange, and Thomas Ertl. 2014. Word Cloud Explorer: Text Analytics Based on Word Clouds. 1833–1842. DOI:https://doi.org/10.1109/HICSS.2014.231

[6] Steven Noel. 2015. Interactive visualization and text mining for the CAPEC cyber attack catalog. In (Proceedings of the ACM Intelligent User Interfaces Workshop on Visual Text Analytics), 1–8.

[7] Plotly. 2017. Introducing Dash. Retrieved from https://medium.com/plotly/introducing-dash-5ecf7191b503

[8] M. Vanamala, J. Gilmore, X. Yuan, and K. Roy. 2020. Recommending Attack Patterns for Software Requirements Document. In (2020 International Conference on Computational Science and Computational Intelligence (CSCI)), 1813–1818. DOI:https://doi.org/10.1109/CSCI51800.2020.00334

[9] Mounika Vanamala, Walter Smith, Xiaohong Yuan, Joi Bennett, and Kaushik Roy. Interactive Visualization Dashboard for Common Attack Pattern Enumeration Classification. In (The Seventeenth International Conference on Software Engineering Advances), 69–74.

[10] About Render. Retrieved from https://render.com/

[11] The Common Attack Pattern Enumeration and Classification. Retrieved from https://capec.mitre.org/about/index.html