Bridging the Gap: From CWEs to TTPs in Cybersecurity Attack Kill Chains

Max Wijnbergen, University of Twente, The Netherlands

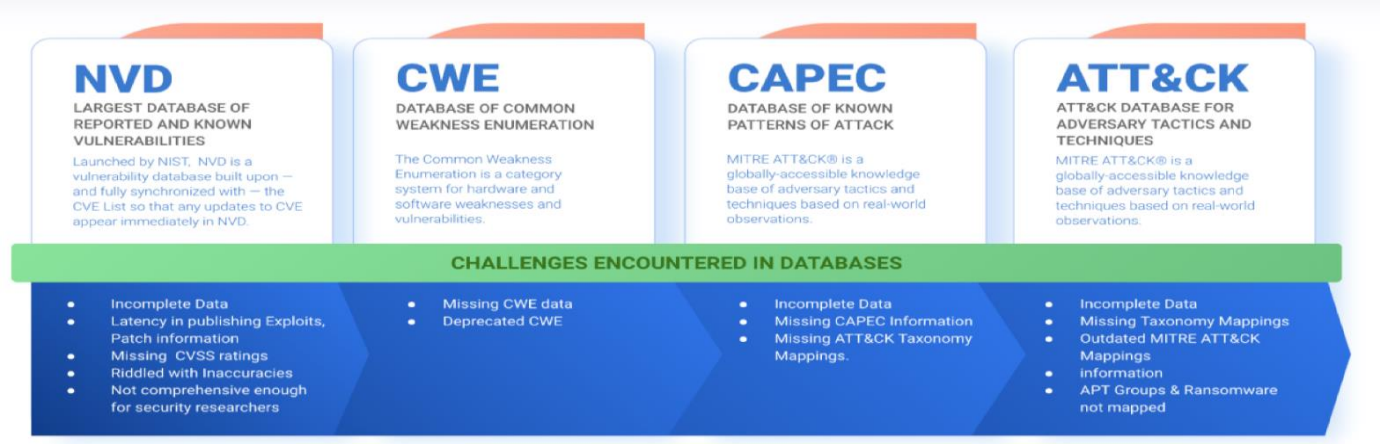

Fig. 1. Gaps and inconsistencies in the NVD, CWE, CAPEC , and ATT&CK databases [6]

ABSTRACT

In cybersecurity, the main aim is to understand and mitigate threats to keep digital assets and private information secure. With technology advancing as fast as it does it is very difficult to keep everything secure. This research paper aims to bridge the gap between Common Weakness Enumerations (CWEs) and Tactics, Techniques, and Procedures (TTPs) by exploring the possibility of finding correlations between them through a comprehensive analysis of historical data in the form of a Cyber Threat Intelligence (CTI) map containing historical incident data. By systematically identifying the CWEs and TTPs contained within each file in the CTI map, correlations can be established to help bridge the gap between CWEs and TTPs. This is done by preprocessing the data in the files of the CTI map. The findings help provide insights into the correlation between CWEs and TTPs. This has the potential to improve the effectiveness of threat detection and mitigation. The methodology used in this study exists out of preprocessing data, and the application of the Jaccard index to identify correlations between identified CWEs and TTPs. Through this research, we aim to create a progression by offering a new perspective on the integration of CWE and TTP data.

TScIT 41, July 5, 2024, Enschede, The Netherlands

KEYWORDS

CWE, TTP, CVE, Jaccard index, MITRE ATT&CK, cyber threat intelligence

1 INTRODUCTION

Cyberattackers keep discovering new vulnerabilities as the world of cybersecurity keeps developing. It is therefore critical to understand and mitigate these vulnerabilities to protect everyone's data. Three of the most important concepts in cybersecurity are CWEs, TTPs, and CVEs. CWEs pinpoint specific software vulnerabilities, whereas TTPs outline the techniques, tactics, and strategies employed by attacks to exploit these vulnerabilities, and CVEs catalog publicly known cybersecurity vulnerabilities. How these entities relate to each other can be seen in figure 2. Despite the importance of all three concepts, the correlation between CWEs and TTPs is hard to deduce, which complicates the development of defense strategies to mitigate the threats. The complications regarding the databases of these entities can be seen in figure 1. Current cybersecurity practices have the urge to often classify CWEs and TTPs as separate entities. Which neglects the opportunity to make use of the correlation between them to mitigate and detect threats. This study acknowledges the potential development of cybersecurity by focusing on the correlation between CWEs and TTPs. By exploring the correlations through a detailed analysis of a CTI map containing historical incident data. With the use of Python codes to identify the CWEs and

TTPs in the CTI map and applying the Jaccard index formula to extract the correlation between these entities.

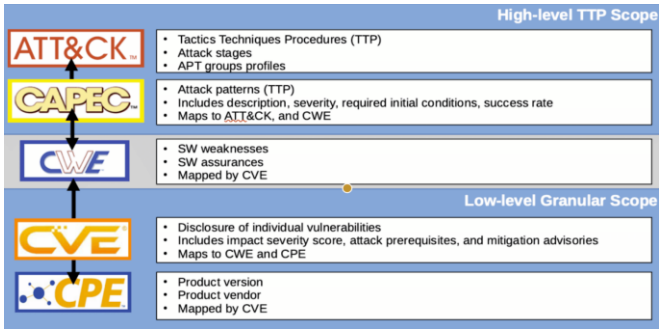

Fig. 2. From high-level TTP scope to low-level granular scope [6]

The main objective of this research is to bridge the gap between CWEs and TTPs, by providing insights that can improve threat detection and mitigate threats. This will provide a better understanding of the vulnerabilities' concatenation. The acquired data and results gathered from this research will help future researchers to understand the world of cybersecurity even more and prepare cybersecurity analysts against cyber criminals. The research question needs to focus on identifying the correlations between CWEs and TTPs and understanding how these correlations can be used to enhance cybersecurity frameworks.

The problem is tackled by answering the following research question:

How can correlations between
CWEs and TTPs be identified and utilized to effectively mitigate cyberattack vulnerabilities?

To answer the research question, consider the following sub-questions:

- What are the key challenges in correlating CWEs with TTPs?
- How can existing datasets and frameworks be most effective for mapping CWEs to TTPs?
- What methodologies can be used to systematically identify correlations between CWEs and TTPs?

To address the first sub-question, the research aims to understand the complexities of finding correlations between CWEs and TTPs. Understanding what makes it so hard to assign a TTP to every CWE gives more insight into the entities and the direction the research needs to go in to find useful data. The second question helps uncover new possibilities with the use of already-existing data. And finally, the third question aims to identify directions that potentially can mitigate the gap even further.

The methodology involves a CTI map containing comprehensive data that has been pre-processed, followed by the application of the Jaccard index to measure the similarity between CWEs and TTPs. Applying this approach provides a quantitative measure of the similarity between CWEs and TTPs. To determine whether a Jaccard index is high enough to conclude that a CWE and TTP are correlated, a threshold is established by using the average Jaccard index so the indices are compared to each other. In doing so, the overall distribution of similarity scores within the dataset is reflected and a stable measure of correlation strength across the dataset is obtained.

The structure of the research paper is divided into 6 sections including the introduction as the first section. The second section contains related work that reviews existing literature and relevant research. Which provides a foundation of knowledge on the CWEs, TTPs, and the Jaccard index. The third section contains the methodology. Here the research design is described, as the process of extracting CWEs and TTPs, and the process of choosing to use the Jaccard index. The fourth section displays the results, which are the Jaccard indices. The fifth section analyzes the findings and discusses the results. It explores the possibilities of bridging the gap with the identified correlations between the CWEs and TTPs. How these findings can improve threat detection and mitigation will be discussed. The sixth section is the conclusion, where the key findings will be summarized, answering the research question and providing knowledge on new possible methods of identifying correlations between CWEs and TTPs. The final section discusses potential future research that builds on this study.

2 RELATED MATERAILS

The related materials reviewed in this section create a foundation for understanding the CWEs and TTPs better and creating correlations between them. These materials exist out of related research studies, frameworks, and datasets that support the findings of this research.

Finding correlations between CWEs and TTPs is crucial for the development of comprehensive cybersecurity strategies. By understanding how the weaknesses in software are exploited through attack methods, the threats can be mitigated and detected. There is not much research about bridging the gap between CWEs and TTPs, but there are many who indirectly relate to the research by focusing on other aspects and entities. The work of the researchers at Securin [11] aims to map KEVs with the corresponding CWEs, with use of the MITRE ATT&CK framework. The research highlights the challenges, including the inconsistency in how vulnerabilities and threats are documented, which complicate establishing correlations between CWEs and TTPs.

The article by Adam et al., [1] also focuses on mitigating cyber threats. To do so, they take the same approach, but search for different information. Instead of using CVEs to map CWEs to TTPs, they use CWEs to map CVEs to TTPs. Unlike this research, they aim to correlate CVEs to TTPs. The taken approach is the same as they first correlate CVEs to CWEs and then correlate the CWEs to TTPs. This turned out to be very challenging due to the limited data available and the differences in descriptions between CVEs and TTPs as they exist out of two very different vocabularies. What relates this study to this research is that they also used similarity search, with the use of the cosine

similarity. The cosine similarity measures the similarity between vectors, in this case between the vulnerability and attacks. However, due to the difference in vocabulary, this was not a success.

The article by Hemberg et al., [7] is another article that tries to tackle cyber threats by introducing a graph model called BRON (Behavioral Relation Network). The study aims to systematically connect different aspects of cybersecurity. Again is the focus here on the correlation between CVEs and TTPs. In figure 3 the schematic of BRON's graph of combined sources can be seen.

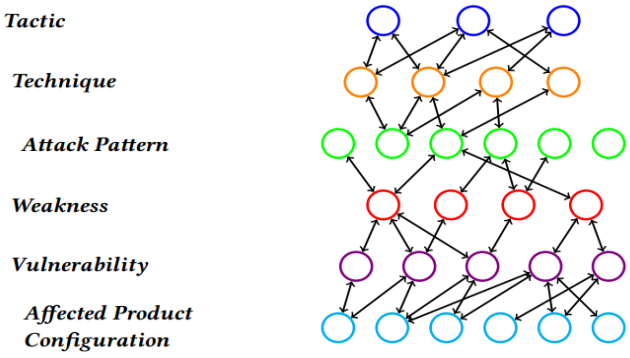

Fig. 3. Schematic of BRON's graph of the combined sources. [7]

All discussed research makes use of the MITRE ATT&CK framework [10]. The MITRE ATT&CK framework contains a base of adversary TTPs. It is a resource used by cybersecurity professionals to understand the behavior of cybercriminals. The framework exists out of multiple matrices, every matrix depicts a different stage of the attack. It is seen as a common language in cybersecurity and therefore also adopted by many research studies. Figure 4 shows the process of mapping done at MITRE ATT&CK. Not only the MITRE ATT&CK framework is very commonly used, but also the MITRE CWE database [2]. The research is mostly focused on mitigating threats and strategies by identifying and categorizing vulnerabilities.

Those vulnerabilities are the CVEs, which are found in the National Vulnerability Database (NVD) [9]. The NVD is maintained by the NIST, also known as the National Institute of Standards and Technology. Every CVE entry is stored there and all known information such as the correlated CWE, which is used in many studies.

3 METHODOLOGY

To extract the relevant data and get the desired results, multiple steps need to be taken. The data needs to be extracted, and a suitable formula needs to be applied to find the similarity between two sets.

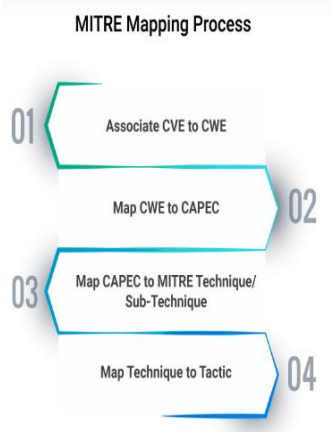

Fig. 4. Mapping from CVE to MITRE ATT&CK [6]

3.1 Collecting and extracting data

The data that is used in this study is collected from multiple sources. The CTI map contains historical incident data derived from the Chainsmith dataset detailed in the paper by Zhu and Dumitras (2018) [12]. This dataset exists out of threat intelligence reports that were crucial to this research. In the files, no CWEs are directly mentioned but the vulnerabilities are. This means that the CVEs are contained within the files, to get the CWEs, the correlation between the CVEs and the CWEs must be found. To do so, a file is used containing found correlations between CVEs and CWEs. The file is extracted from the NVD website [8]. To obtain the TTPs a different process needed to take place. Since the CTI files do not mention the TTPs, they need to be retrieved from the text, by making use of MITRE ATT&CK [10].

3.2 Jaccard index

To find the correlation between the found CWEs and TTPs the Jaccard index is used [3]. The Jaccard index is a formula to measure similarity between two sets, it is defined as the intersection of two sets divided by the union of the two sets. Which gives the following formula:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

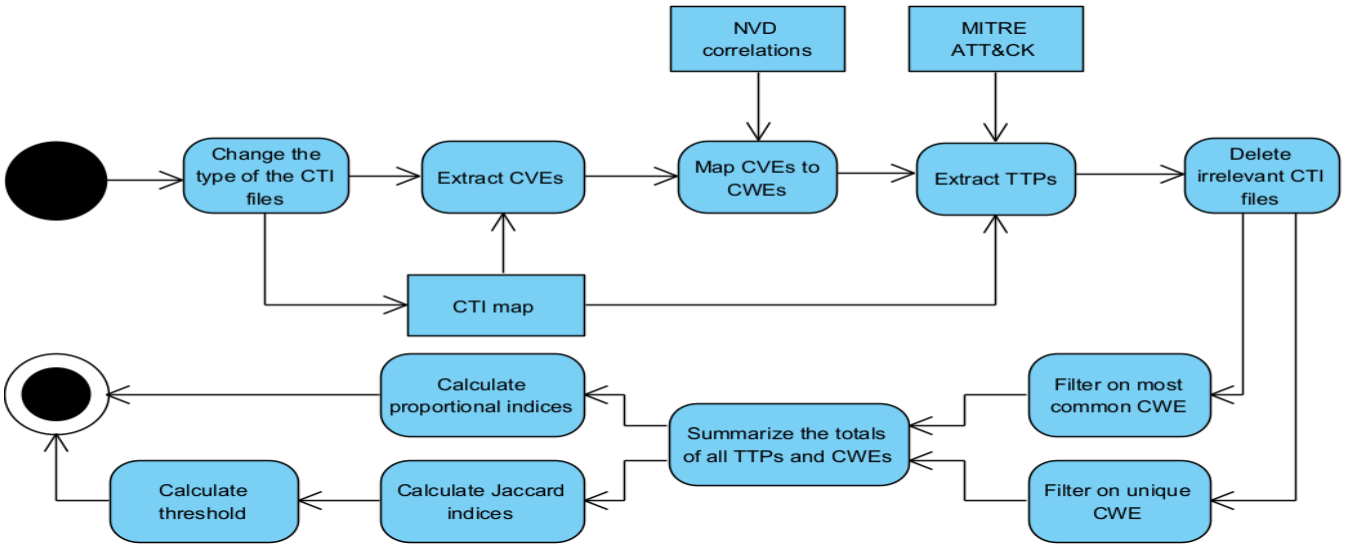

Fig. 5. The process of calculating the indices from the CTI map created with the use of Visual Paradigm.

This index is suitable for this research as it provides a clear metric to evaluate the overlap between CWEs and TTPs. The decision to use the Jaccard index was motivated by its simplicity and scalability. Which provides a clear understanding of the overlap between the two sets.

Other similarity coefficients were considered but deemed not as suitable as the Jaccard index. For example, the Overlap coefficient [3] with the following formula:

$$O(A,B) = \frac{|A \cap B|}{\min(|A|,|B|)} \quad (2)$$

The formula is very similar but it divides the intersection by the smallest of the two sets. This is not appropriate since it will give not give a suitable ratio for this research. For example, if CWE119 would have been found 11 times and TA0040 would have been found 15 times and the intersection is 9 times, the Overlap coefficient returns a ratio of 9/11 which is the ratio for CWE-119 being found in the same file as TA0040 but it does not take into consideration the ratio for TA0040 being in the same file as CWE-119 because that would be 9/15. The Jaccard index does take this into account and would return a ratio of 9/(11+15-9) = 9/17.

3.3 Walk-through

The codes used are in the GitHub repository which can be found in Appendix A and the codes have been established with the help of Appendix B. The data that has been used all came from the map CTI, which contains all the files containing the CVEs and TTPs. To make the CTI file useful, first the information needs to be accessed correclty. This meant that the files needed to be saved as text files so that no errors would arise. This is done by creating a Python code that would change the types of all the files contained in the CTI map to text files. In figure 5, an activity diagram is shown that explains a simplified version of the process.

3.3.1 CWE extraction and mapping

Now that the text in the files can be correctly accessed, a code is created that scans the text files and retrieves the relevant data by going through all files in the CTI map and retrieves all the data containing CVE IDs.

In this research CWEs need to be found, not CVEs. So to do that the CWEs that correlate to the extracted CVEs need to be found. The existing known correlations between CWEs and CVEs are extracted from the NVD . A code is used to find matching CVEs, for each matching CVE it takes the CWE that correlates to that CVE and it creates a new file where the matched CWEs are stored.

3.3.2 TTP extraction and mapping

The next step to take is to find the correlating TTPs for every file. This is harder to do because the CVE IDs are all listed in each file but the TTP is not. So to find the TTP, attack.mitre.org need to be used. On this website all TTP IDs have been listed and their names. A code is created that scans each text file for TTP titles. The code goes through the tactics listed on the MITRE ATT&CK framework tactics page [4] and the techniques listed on MITRE ATT&CK framework techniques page [5]. Every time a TTP title from MITRE ATT&CK has been found, the code parses the TTP name and ID into a new file that now contains the TTPs and CWEs for each text file. Now all the relevant data needed is available.

3.3.3 Information filtering

To help use the data more efficiently the data needs to be cleaned a bit more. Starting with removing any file that does not contain either a TTP or CWE, this is because it would decrease the accuracy of the indices. If there exist multiple files with no TTP but with multiple CWEs then the calculation of the Jaccard index would suffer from it. For example, if there is a CWE that has been found in 6 different occasions, 4 times the CWE has been correlated to, for example, TA0002 and the other 2 times it has not been correlated to a single TTP. This would decrease the accuracy of the Jaccard index since in 100% of the cases the

CWE has been correlated to TA0002 which would imply that they are correlated, but because 2 times the CWE has not been correlated with a single TTP, the Jaccard index will return that they are only correlated 66% of the time which is a big difference. To prevent this a Python code has been introduced to get rid of files that do not contain a single CWE or TTP.

3.3.4 Summarizing relevant data

The next step is to make a summary of the amount of every CWE found and every TTP that has been found. There are multiple ways on how to calculate the Jaccard index, these ways differ concerning what data is used. To calculate the Jaccard index a summarized file is needed that contains a summary of the number of times each TTP has been found and the amount of times every CWE has been found in total and how many times every CWE has been found in the same file as every specific TTP. The complications regarding the total CWEs found in each text file to calculate the Jaccard index arose since every text file can contain only one instance of every TTP but multiple instances of the same CWE. An example of a complication that can arise is found between the TA0002 and CWE-787. There are 85 instances of TA0002 in the entire CTI map and CWE-787 has appeared 144 times in the entire CTI map. The amount of times they have appeared in the same file is 142 times. This would give a Jaccard index of 142 / (85 + 144 - 142) = 1.632 which is not possible since the Jaccard index is a number between 0 and 1. This outcome occurs because in many files multiple unique CVEs have been correlated to CWE-787 which results in the total intersection between TA0002 and CWE-787 being bigger than the total amount TA0002 appeared. To solve this problem there are multiple ways to go.

One of the solutions is to filter the text file and delete all duplicate CWEs for every text file. This is done with a Python code where a new file is created, where all the duplicate CWEs are deleted. So now, if a text file in the CTI map contains multiple CVEs correlating to the same CWE, that CWE will only be counted once. The next step is to summarize the file to be able to calculate the Jaccard index. Now all relevant data is gathered to calculate the Jaccard index and determine the similarity between the CWEs and the TTPs of the used CTI map.

A second option for the calculation of the Jaccard index is by only taking the most common CWE into account for every text file. With the use a Python code, the file containing the matched CWEs and TTPs is filtered so that only the most common CWE of every text file is taken into account. If multiple CWEs have been encountered most often in a text file, then both are considered the most common CWE. So the Python code checks for every article in which CWE has been mentioned the most times.

3.3.5 Calculating Jaccard indices

To finally calculate the Jaccard index a Python code is created. The code checks for every TTP it goes through every CWE that is listed in its map, and it checks how many they have in common which is the intersection. This is then divided by the union of the TTP and the CWE. And finally, the intersection will be subtracted to calculate the union of the TTP and the CWE. This concludes the calculations to find the Jaccard index. The calculations of the first option have now been finalized, but to give a more thorough and accurate calculation, the Jaccard index has been calculated in multiple ways to gain more insight on the correlations. To calculate the Jaccard index for the second, the same code is used as the previous calculated Jaccard index but with a different input file and output file.

3.3.6 Calculating proportional indices and thresholds

The Jaccard index calculations are now complete, and the correlations have been established based on the methods used. But to get even more insight into how these numbers came to place another index has been calculated. This time the index is calculated based on either the TTP or the CWE. So the impact of both factors can be determined and if the total amount of intersections is similar for both factors compared to the total amount of the specific TTP or CWE. This means that the amount of intersections has been divided by the total amount of that specific CWE to see what percentage of that CWE has been correlated with the TTP in question. This has been done by creating a Python code that is slightly different from the other Jaccard calculation Python codes since it does not divide the intersection by the union but only by the total amount of instances of that CWE. This code those not calculate the Jaccard index but the proportion of the CWE that is correlated to a specific TTP. The same thing has been done concerning the TTPs, where again the intersection was not divided by the union but by the total amount of instances of that TTP. The next step is to calculate the thresholds, this is done by taking the average of the calculated Jaccard indices. This resulted in two thresholds, one for the Jaccard indices based on unique CWEs and for the Jaccard indices based on the most common CWEs. The average is taken for the Jaccard indices and resulted in the following thresholds: 0.1044 for the most common CWEs, and 0.0881 for the unique CWEs.

4 RESULTS

The purpose of this study was to identify correlations between CWEs and TTPs. The approach taken to do this is by use of the Jaccard index. By extracting the CWEs and TTPs through numerous Python codes from a CTI map, it was possible to apply the Jaccard index.

A total of 4 files containing different measurements have been created and some of the key findings will be presented here. A high Jaccard index indicates a high similarity and therefore a high correlation between the CWE and the TTP.

The correlations shown in the tables only contain CWEs and TTPs that have both been encountered more than 4 times. This is because there are a lot of CWEs and TTPs that have only been encountered once which results in a Jaccard index of 1, but that is not be accurate due to the low amount of times they have been encountered. Multiple indices result in 1 because only one instance of a certain CWE and one instance of a certain TTP has been encountered which indicates a Jaccard index of 1. This is not reliable because if they are in the same file 1 time does not mean they are correlated 100% if the time. So the table only

shows Jaccard Indices that exist out of CWEs and TTPs that are both encountered more than 4 times to make sure the Jaccard index is more reliable and accurate.

4.1 Jaccard index most common CWE

Table 1: Top Jaccard indices based on the most common CWE per CTI file

TTP (total)	CWE (total)	Jaccard index
TA0002 (85)	CWE-119 (37)	0.2577
TA0002 (85)	CWE-787 (22)	0.2299
T1176 (5)	CWE-200 (7)	0.2000
TA0003 (9)	CWE-94 (10)	0.1875
TA0010 (7)	CWE-416 (6)	0.1818
TA0002 (85)	CWE-20 (19)	0.1667

The table displays some of the highest Jaccard indices that are acquired by only assigning the most common CWE per file. For example, if CWE-119 has been encountered 5 times in a certain file, and CWE-787 has been encountered 3 times in that same file, only CWE-119 will be counted for that file. As displayed in the table, TA0002 has been encountered 85 times which is very high in comparison with other TTPs. The Jaccard index indicates that the correlations found between the CWEs and TTPs are not correlated with each other. The Jaccard index shows that the CWEs and TTPs in Table 1 have a similarity ranging from around 15% to 25%. All results in the table are higher than the threshold of 0.1044 which indicates that the highlighted results are moderate correlations within the dataset.

4.2 Jaccard index unique CWEs

Table 2 calculates the Jaccard index by taking all the unique CWEs from every incident file.

Table 2: Top Jaccard indices based on unique CWEs per CTI file

TTP (total)	CWE (total)	Jaccard index
TA0002 (85)	CWE-20 (33)	0.3258
TA0002 (85)	CWE-119 (42)	0.3093
TA0010 (7)	CWE-22 (6)	0.3000
TA0040 (34)	CWE-20 (33)	0.2642
TA0002 (85)	CWE-787 (23)	0.2414
TA0001 (5)	CWE-22 (6)	0.2222

Table 2 again displays some of the highest Jaccard indices, this time acquired by using unique CWEs, this means that all CWEs are extracted from every file but if, for example, CWE-119 was encountered more than once in a file, the duplicates were discarded. If we compare Table 2 with Table 1, it can be determined that the total of the CWEs is different. CWE-119 has been encountered 37 times in Table 1, and 42 times in Table 2. This means that CWE-119 has been encountered in 42 different files, and has been the most common CWE in 37 different files. The Jaccard indices are higher than Table 1 but still conclude that there is no correlation between the CWEs and TTPs. The highlighted Jaccard indices are again higher than the threshold of 0.0881 in this case. Which indicates that these indices represent moderate correlations within the dataset.

4.3 Proportional Correlations

These tables show the proportion of how many instances of a specific CWE are correlated to each TTP, and how many instances of a specific TTP are correlated to each CWE. This research aims to get more insight into the factors that result in the Jaccard Indices.

Table 3: Top CWE proportional correlations and corresponding TTP proportional correlations

TTP (total)	CWE (total)	CWE proportional	TTP proportional
TA0002 (85)	CWE-416 (14)	1.0000	0.1647
TA0002 (85)	CWE-843 (4)	1.0000	0.0471
TA0002 (85)	CWE-787 (23)	0.9130	0.2471
TA0002 (85)	CWE-20 (33)	0.8788	0.3412
TA0040 (34)	CWE-125 (5)	0.6000	0.0882
TA0040 (34)	CWE-200 (12)	0.5000	0.1765

Table 3 takes the same approach as Table 2 but calculate a different index. The index calculated in Table 3 is not the Jaccard index, instead of dividing by the union of the CWE and the TTP, the intersection is divided by the CWE for the CWE proportional index and for the TTP proportional index it is divided by the total of that TTP. This is because the research wants to find out where the Jaccard indices are based on, to see what makes the Jaccard indices the index that they are . By using this calculation it can be seen that the percentage of the amount of times a certain CWE is correlated with a certain TTP. For example, the huge differences between the CWE proportions and the TTP proportions immediately can be seen, which indicates that there is a huge difference in the correlation between each other. CWE-416 is in all cases correlated with TA0002, which means that all 14 encountered instances of CWE-416 have been in files where TA0002 was also encountered, but TA0002 is only correlated with CWE-416 in 16.47% of the cases.

Table 4 is displayed below and shows the same calculation but this time it displays the top TTP proportional correlations. So the differences between how CWEs and TTPs are correlated to each other can be found. Again, very big differences between the indices apart from two can be seen, TA0010 together with CWE-22, and TA0040 together with CWE-20,

these two are also both in Table 2, which displays some of the highest correlations found.

Table 4: Top TTP proportional correlations and corresponding CWE proportional correlations

TTP (total)	CWE (total)	CWE proportional	TTP proportional
T1014 (6)	CWE-119 (42)	0.1190	0.8333
TA0009 (10)	CWE-119 (42)	0.1429	0.6000
TA0003 (9)	CWE-119 (42)	0.0952	0.4444
TA0010 (7)	CWE-22 (6)	0.5000	0.4286
TA0040 (34)	CWE-20 (33)	0.4242	0.4118
TA0001 (5)	CWE-20 (33)	0.0606	0.4000

5 DISCUSSION

The results present the correlations between CWEs and TTPs, which bring new insight into the understanding of the relationships for developing new strategies and frameworks to mitigate cyber threats. The most common CWE correlations table aims to highlight which TTPs are most often associated with the most common weaknesses.

The results of Table 1 show a range of Jaccard indices that are high compared to the established threshold, which suggest that the CWEs and TTPs found are correlated. The thresholds measure the average Jaccard indices found in the CTI dataset and based on the dataset it determines when a Jaccard index is high enough to be considered a correlation between a CWE and TTP. A few outliers that indicate a high similarity between certain CWEs and TTPs have been acknowledged but not displayed in the tables. Even though these outliers show good results, the Jaccard index might not be accurate due to the low amount of encounters. The highest, more accurate, correlating CWEs and TTPs that have been encountered the most have been displayed in the tables. The results are low, but still give us important insight to mitigate threats. Table 2 presents the correlations between CWEs and TTPs based on unique CWEs. By presenting all unique found CWEs in the dataset, the table highlights the span of weaknesses that can be exploited by various TTPs. It brings new insight to help understand the variety of weaknesses and the different attack methods that are associated with them. The results again indicate that there are correlations between certain CWEs and TTPs in the dataset. Compared to Table 1 the Jaccard indices are higher which also indicates a higher threshold. The highest index is 0.3258, which indicates that in 32.58% of the cases, TA0002 and CWE-119 are correlated, which is considered a moderate correlation according to the established threshold of 0.1044. Table 2 also shows multiple moderate correlations based on the threshold of 0.0881.

The results from Table 1 and Table 2 indicate that there are multiple correlations found between the CWEs and TTPs within the dataset. To gain more insight and a better understanding of the results, Table 3 and Table 4 were introduced.

Table 3 and Table 4 show the proportion of how many instances of a specific CWE are associated with each TTP and vice versa. The purpose of calculating these ratios is to identify patterns that might not be noticeable in the found Jaccard indices. By examining the results, we can see which TTPs tend to target specific types of weaknesses more consistently. In doing so, it might be possible to identify certain TTPs that are more likely to exploit certain weaknesses.

The results from Table 3 and Table 4 give contradicting CWE and TTP proportions. Some CWEs from Table 3 even have a perfect correlation with certain TTPs, which indicates that all encountered instances of that CWE are found in the same files as the correlated TTP. But the TTP is not correlated with that CWE in many cases. Many others show a considerably high correlation with certain TTPs. This indicates that the number of encountered TTPs might cause the Jaccard indices in Tables 1 and 2 to be considerably lower.

The way that the TTPs are extracted from the dataset is by extracting the names of the TTP IDs. Even though this is the most consistent way of extracting TTPs, it might have resulted in wrongly identifying some TTPs from the text files. This is because some tactics have very common words as their name, for example, TA0002 has been encountered the most by far, but it name is ‘execution’, which can be very common to find in CTI files without it being referenced to the TTP. Overall, the approach correctly calculates the similarities found between CWEs and TTPs, but to continue with this approach, more time needs to be invested in extracting TTPs to enhance the accuracy and reliability of the correlations.

6 CONCLUSION

This research aimed to bridge the gap between CWEs and TTPs. There are a lot of challenges in finding correlations between CWEs and TTPs. One challenge that stood out was the lack of data and research. To avoid this problem another approach was taken, a dataset with historical data was used to extract CWEs and TTPs and correlate them. Another key challenge is the extraction of TTPs. Extracting CWEs was made possible because every incident file listed corresponding CVEs to each incident. With the use of existing correlations between CWEs and CVEs, CWEs were extracted from the incident files. This was not the case for the TTPs, the TTPs had to be extracted directly from the text in the incident files. This process was more complex and prone to inconsistencies due to the way TTPs are documented on the MITRE ATT&CK framework. To extract the TTPs a Python code was created that extracted the TTP IDs by scanning through the dataset for the TTP names and finding the corresponding IDs on the MITRE ATT&CK framework. This remained challenging due to some TTP names being common

words. This challenge underscores the need for improved data collection and standardized practices to extract TTPs.

Identifying the most effective datasets and frameworks is crucial for mapping CWEs to TTPs. The MITRE ATT&CK framework exists out of a base of TTPs. It aids cybersecurity professionals in understanding the behavior of cybercriminals. With the use of the ATT&CK MITRE framework, it was possible to extract the TTPs from the used dataset. Another very important resource is the National Vulnerability Database (NVD), it holds every CVE entry and their relevant information, such as their correlated CWE. And finally, the MITRE CWE database holds all information regarding CWE. With the use of these databases and frameworks, identifying the CWEs and TTPs in the dataset by extracting them from the text was made possible.

To identify correlations between CWEs and TTPs multiple methodologies can be applied. Some studies make use of data mining, and some apply machine-learning techniques to be able to analyze large datasets to find patterns and relationships. In our case, we made use of natural language processing (NLP) to extract information from textual data in incident reports. Another methodology used to identify correlation is the use of statistical analysis, which can include the use of the Jaccard index to measure similarity. The use of a framework like MITRE ATT&CK is also considered a very useful methodology. It provides a structured way to map TTPs and to find correlations between CWEs and TTPs. By combining these methodologies, a well-structured approach to identify correlations between CWEs and TTPs is established.

Identifying correlations between CWEs and TTPs requires overcoming multiple key challenges, which include the limited available data and the complexity of extracting the relevant information. Extracting the relevant information is difficult due to the varying documentation standards. That is why NLP and machine learning are crucial for identifying TTPs. Existing frameworks, such as the MITRE ATT&CK framework, are essential to map entities like CVEs, CWEs, and TTPs to each other. With the use of methodologies for systematically identifying correlation, such as data mining and statistical analysis, patterns within large datasets can be uncovered. Statistical analysis allows us to measure the similarity between CWEs and TTPs with, for example, the Jaccard index.

By integrating these approaches, cybersecurity professionals can identify and utilize correlations between CWEs and TTPs to effectively mitigate cyberattack vulnerabilities.

7 FUTURE RESEARCH

In this study, multiple areas are required for further research to enhance the correlation between CWEs and TTPs. The main area that needs a different approach to improve the accuracy of the results is the TTP extraction. Currently, the process of extracting TTP IDs relies on extracting TTP names from incident files. This approach has shown potential inaccuracies due to commonly used words being possibly mistaken for TTPs. For example, TA0002 has been encountered 85 times, compared to TA0040 which has been encountered the second most amount of times with a total of 34 times. For example, 'execution' and 'impact'. This is a very big difference and it potentially results from the fact that the names from these tactics are common words in the field of cybersecurity. Future research must develop a more reliable method of extracting TTP IDs. One approach used in related research was the use of Doc2Vec, it is a machine-learning model designed to create a numeric representation of text documents, by generating a unique vector for each document if trained correctly [1]. Including more historical data could also help make the Jaccard indices more accurate. This expansion would help due to the limited amount of data that has been used in this study.

REFERENCES

[1] Adam, C., Bulut, M. F., Sow, D., Ocepek, S., Bedell, C., & Ngweta, L. (2022, June 22). Attack Techniques and Threat Identification for Vulnerabilities. arXiv.org https://arxiv.org/abs/2206.11171

[2] Common weakness enumeration. CWE. (n.d.). https://cwe.mitre.org/

[3] Digitate. (2023, March 22). Similarity coefficients: A beginner's guide to measuring string similarity. Medium. https://medium.com/@igniobydigitate/similarity-coefficients-a-beginners-guide-to-measuring-string-similarity-d84da77e8c5a

[4] Enterprise techniques. Tactics – Enterprise — MITRE ATT&CK®. (n.d.). https://atta ck.mitre.org/tactics/enterprise/

[5] Enterprise techniques. Techniques – Enterprise — MITRE ATT&CK®. (n.d.). https://atta ck.mitre.org/techniques/enterprise/

[6] fnCyber. (2023, 9 maart). *Understand common attack patterns*. fnCyber. https://www.fncyber.com/web-of-trust-article/understand-common-attack-patterns/

[7] Hemberg, E., Kelly, J., ShlapentokhRothman, M., Reinstadler, B., Xu, K., Rutar, N., & O'Reilly, U. M. (2021). Linking Threat Tactics, Techniques, and Patterns with Defensive Weaknesses, Vulnerabilities and Affected Platform Configurations for Cyber Hunting. arXiv. https://arxiv.org/pd f/2010.00533

[8] NVD - Data Feeds. NVD. (n.d.). https: //nvd.nist.gov/vuln/data-feeds

[9] NVD - Home. NVD. (n.d.). https://nvd.ni st.gov/

[10] Mitre ATT&CK®. MITRE ATT&CK®. (n.d.). https://attack.mitre.org/

[11] *Mitre mapping of Cisa Kevs and its challenges*. Securin. (2023, February 1). https://www.securin.io/mitre-mapping-of-cisa-kevs-and-its-challenges/

[12] Zhu, Z., & Dumitras, T. (2018). ChainSmith: Automatically Learning the Semantics of Malicious Campaigns by Mining Threat Intelligence Reports. Proceedings of the 2018 IEEE European Symposium on Security and Privacy (EuroS&P), 458-474. https://users.umiacs.umd.edu/~tdumitra/papers/EuroSP-2018.pdf

APPENDIX

Appendix A: Code Repository

The source code used for this research can be found at the following GitHub repository:
https://github.com/MaxWijnbergen/Thesis_BIT_MOD12

Appendix B: Use of AI

During the preparation of this work the author(s) used Python, an AI-based language model developed by OpenAI and available in ChatGPT, in order to assist with debugging, and refining Python code. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the content of the work.