Quantifying the severity of cyber attack patterns using complex networks

Ahmed Salih Hasan[1], Yasir F. Mohammed[2], Basim Mahmood[3]
[1]Department of Computer Science, College of Computer Scince and Mathematics, University of Mosul, Mosul, Iraq
[2]Cybersecurity Unit, Computer Center, University of Mosul, Mosul, Iraq
[3] ICT Research Unit, Computer Center, University of Mosul, Mosul, Iraq

Article Info

Article history:

Received Nov 24, 2024
Revised Jun 17, 2025
Accepted Jul 1, 2025

Keywords:

Complex networks
Cybersecurity
Data analysis
Graph theory
Network science

ABSTRACT

This work quantifies the severity and likelihood of cyberattacks using complex network modelling. A dataset from common attack pattern enumerations and classifications (CAPEC) is collected and formalized as nodes and edges aiming at creating a network model. In this model, each attack pattern is represented as a node, and an edge is created between two nodes when there is a relation between them. The dataset includes 559 attack patterns and 1921 relations among them. Network metrics are used to perform the analysis on the network level and node level. Moreover, a rank of the CAPECs based on a complex network perspective is generated. This rank is compared with the CAPEC ranking system and deeply discussed based on cybersecurity perspective. The findings show interesting facts about the likelihood and severity of attacks. It is found that the network perspective should be given attention by the CAPEC ranking system. Finally, the results of this work can be of high interest to security architects.

Corresponding Author:

Ahmed Salih Hassan
Department of Computer Science, College of Computer Science and Mathematics, University of Mosul
University Street, Mosul, Iraq
Email: ahmed_salih_h@uomosul.edu.iq

1. INTRODUCTION

With new technologies being released every day, it is difficult to categorize the top weaknesses and vulnerabilities. This is due to a variety of factors that reflect different facts about them [1]-[3]. However, there are many online databases that exist that aim at identifying and categorizing vulnerabilities based on platform, severity, and other factors [4]. The focus of this work is on discussing vulnerabilities and weaknesses that occur in the CAPEC and common vulnerabilities and exposures (CVE) databases [5]-[7]. These databases rate and classify vulnerabilities and weaknesses based on platform, likelihood and severity. In 2007, the U.S. Department of Homeland Security initially established CAPEC to identify, collect, refine, and share attack patterns among the cybersecurity community. CAPEC establishes relationships between attack patterns and weaknesses, demonstrating how they can be exploited based on information found in CVE and common weakness enumeration (CWE) [8]. CAPEC helps professionals and newcomers in cybersecurity understand attack patterns and how adversaries exploit weaknesses in network protocols and applications to carry out attacks. Thus, CAPEC focuses on assisting individuals and enterprises in designing and implementing systems in a secure manner. Furthermore, analysing CAPECs using traditional statistical analysis approaches has been frequently performed in the literature [9]. However, this kind of approach is efficient but does not dig deeply into the relations among CAPECs. On the other hand, when having data objects and relations among them, the most efficient analysis approach is using concepts inspired by complex networks, which have yet to be utilized in this work. A network is represented as a Graph (G) with nodes (N) and edges (E), where the nodes represent the data object and an edge is created between two nodes when there is a relation between both [10], [11]. The network model can be investigated and analysed using two different kinds of metrics: network-level metrics, which measure phenomena in the whole network structure [12]. While node-level metrics measure the performance of an individual node within the network structure [13]. Using complex network metrics, many deep facts in data can be extracted since these metrics reflect different perspectives in the data.

The literature includes a lot of works that consider CAPECs analysis using a graph-based approach. The reason behind using this kind of approach is its ability to extract facts about data that is difficult to extract using traditional approaches. Miyata *et al.* [14] used a graph-based method to model the CAPECs aiming to investigate the relations among attack patterns. They found that several relations are missed among CAPECs and these relations are important to be utilized by security architects. This kind of study is useful when investigating security attack patterns since it demonstrates the analysis from different perspectives. Another study that used a graph-based approach for analyzing attack patterns was performed in [15]. They suggested an automated method for modelling attacks on a computer network. The graph was used to analyze distinctive features of the attacks against the mobile components of the network. The analysis was performed considering hardware and software vulnerabilities, and the weaknesses of mobile channels. Their work also suggested a metric for evaluating security risks in the network. In the same context, the authors in [16], [17] suggested a graph-based model for analyzing hardware security weaknesses and vulnerabilities. Their generated model was analyzed using network metrics aiming to show the severity of weaknesses considering network perspective. The findings showed that graph-based modeling of security risks is considered a powerful tool in assessing security weaknesses and their corresponding vulnerabilities. Similar studies considered CWE, CVE, and CAPEC to assess attack patterns. Grigoriadis in [18] developed a search engine that connects CVEs and CAPECs using their corresponding CWEs. The approach was useful for the security experts to understand risks and threats. Many graph-based approaches are suggested in the literature such as the studies [19]-[23].

According to the literature, there exists a lack in providing analysis approaches that look at attack patterns from different angles considering their relations to each other. This is important insofar as they can contribute to improve the CAPEC ranking system. For instance, CAPEC categorizes attack patterns mainly based on likelihood and severity, which are calculated using their own rating formula. However, adding more dimensions to the rating formula will definitely improve the accuracy of CAPEC's ranking. Hence, the contribution of this work is to show the powerful of complex networks modeling and metrics in analyzing CAPECs considering the relations among them. This approach can be a complementary approach to the existing scoring system. To this end, a network model that includes all attack patterns in CAPEC dataset is generated. Then, several network metrics are used to extract facts about the CAPECs that can be integrated into the current scoring system, which is of interest to security architects. The organization of this document is as follows: The method followed in this research is described in section 2 including the dataset collection and the methodology followed. Section 3 presents the obtained results and discussion about the results. The work is concluded in section 4.

2. RESEARCH METHOD

2.1. Dataset collection

The dataset used in this work is extracted from MITRE/CAPEC official website. This source is considered one of the most accredited sources of attack patterns. The dataset includes 559 well-defined attack patterns alongside information about each attack. Table 1 presents a description of each entry in the dataset. The dataset is then formalized to be in two files as follows:

- *Nodes File*: includes 559 attack patterns and each one is represented as a node alongside its attributes.
- *Edges File*: includes 1921 relations that exist among each pair of nodes in the dataset. The relations are extracted from the dataset for each attack pattern.

These two files are further used to generate the network model as illustrated in the next section.

2.2. Network generation

The nodes file and edges file are used to generate the network model using a visualization software called Gephi. To illustrate how the network model is generated, consider the following example:
Assume there are 4 CAPECs (CAPEC-1, CAPEC-2, CAPEC-3, and CAPEC-4 and the relations among them are described as follows: CAPEC-1 has relations to all other CAPECs, and CAPEC-2 and CAPEC-3 have a relation between them. Now, a Graph (G) is generated as follows:

Nodes Set: G(N) ={CAPEC-1, CAPEC-2, CAPEC-3, CAPEC-4}

Edges Set: G(E)={(CAPEC-1 → CAPEC-2), (CAPEC-1 → CAPEC-3), (CAPEC-1 → CAPEC-4), (CAPEC-2 → CAPEC-3)}. Figure 1 demonstrates the generated graph of the abovementioned example.

Table 1. CAPEC dataset description

Entry	Description
ID	The identifiers of the attack patterns
Label	The labels (known names) of the attack patterns
Abstraction	The level of details provided to each attack pattern and can be “Meta”, which represents the highest level of description that provides general information, “Standard” means a mid-level of details, and “Detailed”, which is the lowest level and provides highly specific information.
Status	Represents the current status of an attack pattern in the dataset. It can be “Draft”, which means the initial stage, “Stable” means that the attack pattern is reviewed and validated, “Deprecated” means it no longer exists or is outdated, and “Incomplete” means that the attack pattern is identified but not yet validated.
Likelihood of Attack	It can be “High”, “Medium”, or “Low”.
Typical Severity	It can be “High”, “Very High” “Medium”, “Very Low” or “Low”.
Category	The category of a particular attack pattern (e.g., Engage in Deceptive Interactions)
Relationships	The relation of the current attack pattern to other attack patterns in the dataset.

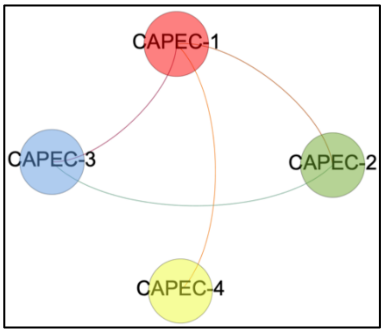


Figure 1. Example of how the CAPEC network model is generated

2.3. Analysis metrics

The analysis metrics that are used in this work can be summarized as follows [10], [12]: *Diameter*, represents the longest distance between network nodes. *Density*, it is the ratio of the number of edges to the number of all potential edges in a network model. *Average Path Length*, reflects the average shortest paths between nodes in a graph. *Clustering Coefficient*, reflects the tendency of network nodes to cluster together. *Degree Centrality*, represents the number of connections that a node has within the network model. And finally, *Betweenness Centrality*, it is the number of shortest paths that pass through a node.

3. Results and Discussion

This section describes the obtained results and discussions. The results are illustrated from the perspective of network metrics, while the discussion is described from cybersecurity perspective and integrated with the network facts.

3.1. Results

The first step of this work is to visualize the attack patterns (AP) network model that includes all 559 attack patterns in CAPEC as shown in Figure 2. The figure shows that the majority of attack patterns are disconnected and do not have relations to other attack patterns according to CAPEC database. However, the figure also depicts that many attack patterns are densely connected to each other, and they are considered influential. The general characteristics of the AP model is presented in Table 2. The average degree of the network is considered low considering the majority of attack patterns are disconnected with a degree of 0. The diameter is considered short; however, the disconnected attack patterns cannot be accessed in the network structure. On the other hand, the average clustering coefficient is considered high due to the fact that the maximum value is 1 meaning that the attack patterns have a significant tendency to cluster together. Similarly, the average path length is also considered short, which means that to access an attack pattern in the network, the distance needed is 1.242 steps from any position within the network. However, these values are counted for only the connected attack patterns since the disconnected ones cannot be accessed. Therefore, Figure 3 visualizes the AP network by discarding the disconnected attack patterns and keeping only the connected ones. The number of connected attack patterns is 63 and the number of relations is 1921, which is the same as the AP network model. The figure demonstrates the connected attack patterns and shows how dense the relations among them. Node size reflects the number of the relations that an attack pattern has, and the label size is proportional to node size.

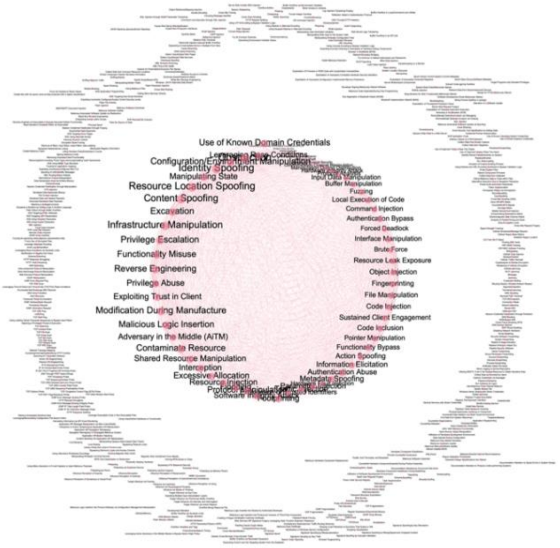


Figure 2. Visualization of Ap network model, node size reflects the degree of the nodes (high degree means bigger nodes)

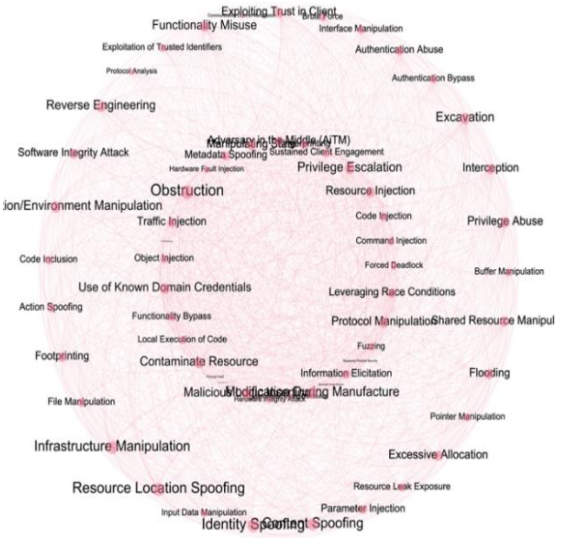


Figure 3. Connected attack patterns in AP network model

Table 2. Characteristics of AP network model

# of attack patterns	# of relations	Average degree	Diameter	Density	Average clustering coefficient	Average path length
559	1921	6.873	3	0.012	0.937	1.242

As mentioned in the previous section, the clustering coefficient of a node (attack pattern) is a measure of the tendency of a node to be clustered with other nodes. This means it assessed an attack pattern whether it has a strong or weak tendency to be associated with another attack pattern(s), which is an important factor important to be considered by security architects. The top 10 attack patterns in AP model are presented in Table 3. The table presents the *status* of the attack patterns, it can be observed that 5 of the top 10 attack patterns are *draft* and the other 5 are *stable*. This means, the stable attack patterns should not be the only interest of security architects. Another observation, the *likelihood of attack* in the top 10 attack patterns is not always *high*, some of them are *low* while the other are *medium*. Furthermore, the *medium* severity in the table appears twice, which is interesting. Finally, the number of relations of an attack pattern does not express its importance in the community of attack patterns, therefore; it can be seen that highest clustering coefficient values are obtained by the lowest degree attack patterns in the table. Hence, the power of complex network methods can be considered useful to security architects because it provides more dimensions of analysis as well as more insights and perspectives about the importance of the relations among attack patterns, which is useful in strengthening the security strategies and defenses against attacks. Moreover, other visualizations are performed to reveal more knowledge about the relations among attack patterns. Figure 4 depicts the relations between the *stable* and *draft* statuses of attack patterns. The red nodes represent the *stable* attack patterns, and the green nodes represent the draft attack patterns. As can be seen, the visualization shows a dense connection between both statuses. Another visualization about *the likelihood of attack* is performed as shown in Figure 5. The red nodes denote the attack patterns with *high* likelihood of attack, the green for *low*, the orange for *medium*, and the other nodes do not show values according to CAPEC. The visualization demonstrates dense relations among the different values of likelihood of attacks. Furthermore, the typical severity of attack patterns is also visualized as shown in Figure 6. The dark green nodes in the figure denote the *very high* severity of the attack patterns, the light green denotes the *high* severity, the orange nodes for *low* severity, the red nodes for *very low* severity, the purple nodes for the medium severity, and the other nodes don't typical severity according to CAPEC. As seen, there are also dense relations among the different typical severity levels.

Table 3. Top 10 attack patterns in the AP network model alongside other indicators

Rank	Attack pattern	Status	Likelihood of attack	Typical severity	Degree	Clustering coefficient
1st	Manipulate Human Behavior	Stable	Medium	Medium	9	1.0
2nd	IP Address Blocking	Draft	Low	High	4	1.0
3rd	Brute Force	Draft	N/A	High	52	0.996
4th	Forced Deadlock	Stable	Low	High	51	0.996
5th	Command Injection	Stable	Medium	High	55	0.996
6th	Fuzzing	Draft	High	Medium	53	0.996
7th	Buffer Manipulation	Draft	High	Very High	54	0.996
8th	Input Data Manipulation	Draft	N/A	Medium	54	0.996
9th	Local Execution of Code	Stable	Medium	High	55	0.983
10th	Code Inclusion	Stable	Medium	Very High	59	0.982

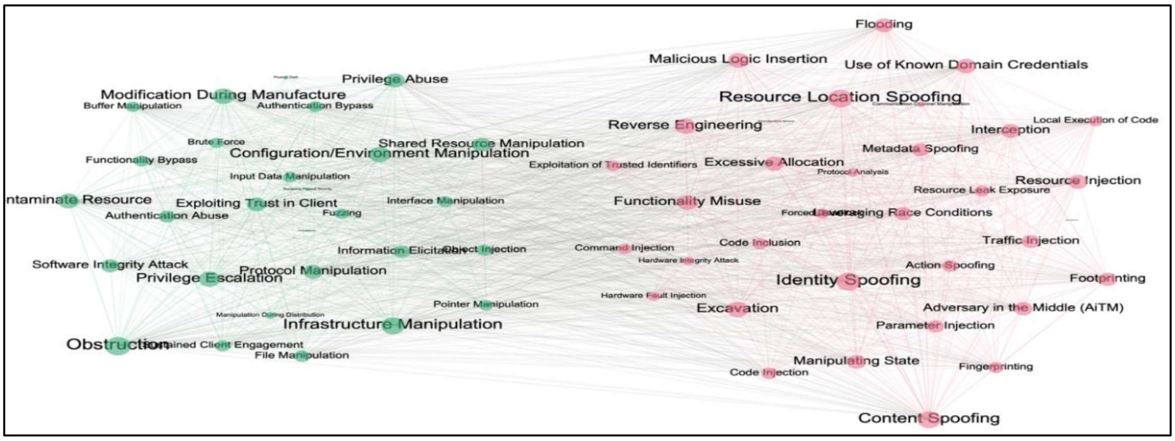
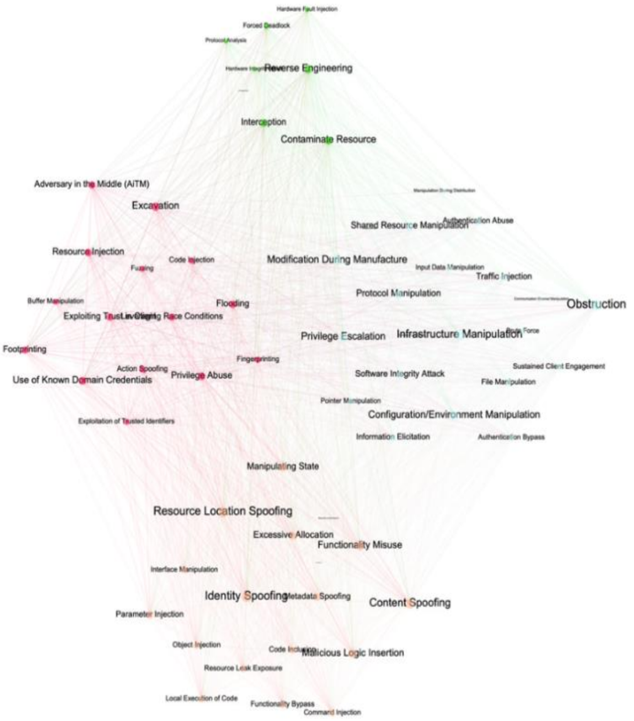


Figure 4. Relations between stable status (red nodes) and draft status (green nodes) in the AP network model of the connected attack patterns



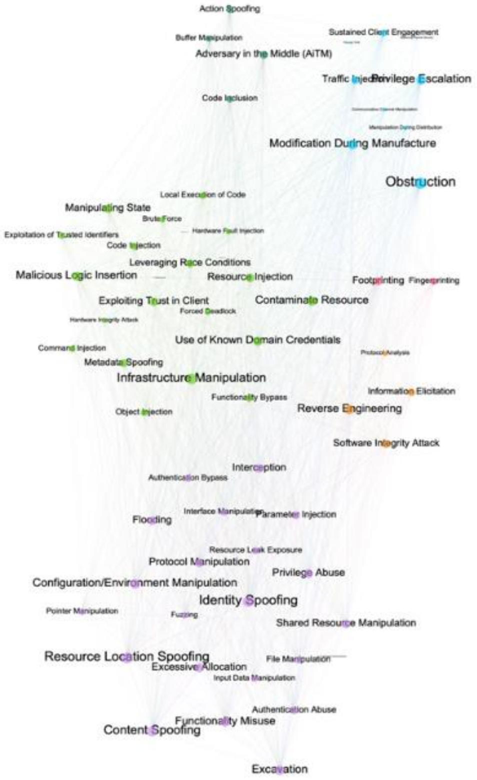


Figure 5. Visualization of attack patterns with their likelihood of attack

Figure 6. Visualization of attack patterns with their typical severity

In addition to the aforementioned description, the weaknesses that are associated with the attack patterns are extracted from CAPEC and CWE databases. The goal of this step is to deeply analyze the attack patterns and investigate their relations to the weaknesses. The network model that combines all attack patterns (559) and their associated weaknesses is visualized in Figure 7. However, the visualization is updated by discarding the disconnected nodes as demonstrated in Figure 8. The characteristics of this network model are presented in Table 4. The average degree is significantly high, this density in connections causes the density of the network to be also high. The diameter is 4, which means that the network has gaps in the connections and therefore; the average path length is also high. Moreover, the average clustering coefficient is 0.881 (close to 1), which is also high meaning that the network has a strong tendency to cluster with the nodes in the network. On the other hand, the visualization of the network in Figure 8 shows attack patterns that are relatively influential in terms of their positions. In complex networks, the well-positioned nodes that play as bridges in the network have high values of betweenness centrality. Therefore, the top 10 highest values of betweenness centrality of the attack patterns are presented in Table 5. According to the values of betweenness centrality of the top 10, it is observed that most of the statuses appear in the table are *stable*, and two attack patterns with N/A likelihood of attack are *draft*, which is reasonable. This result reflects the fact that the betweenness centrality plays a significant role in distinguishing the influential attack patterns in this work. Moreover, the clustering coefficient is investigated to show the attack patterns that expose a strong tendency to associate with other attack patterns in the attack pattern and weaknesses network model. Table 6 shows the top 10 attack patterns that have the highest clustering coefficient.

According to the table, the top two attack patterns are similar to what has been presented in Table 3. This is because even with the existence of the weaknesses in the network, the clustering coefficient of these two attack patterns still the same, which reflects their importance when it comes to the tendency to cluster with other attack patterns in the network. Interestingly, the majority of the top 10 have *draft* status. Also, only one attack pattern with a *high* likelihood of attack is shown in the table. The table also presents three attack patterns with *medium* severity and one with *low* severity. These results are interesting since they provide the perspective of complex networks when analyzing attack patterns, which is useful for security architects. On the other hand, Table 7 presents the top 10 weaknesses based on the values of clustering coefficient. The table demonstrates the top 10 weaknesses with a value of clustering coefficient of 1. Weaknesses with medium or unknown *likelihood of exploit* are shown in the table. This is also an indicator of the fact that complex networks can provide insights that add another dimension to the analysis. Therefore, the weaknesses with high clustering coefficient should be given special attention because they represent the road to many attack patterns.

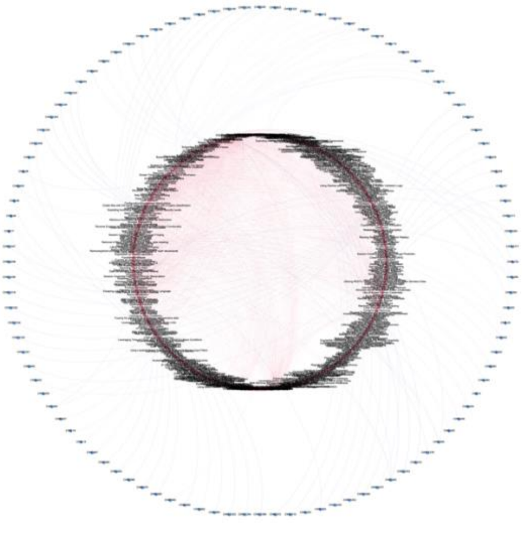


Figure 7. Visualization of all attack patterns and their associated weaknesses

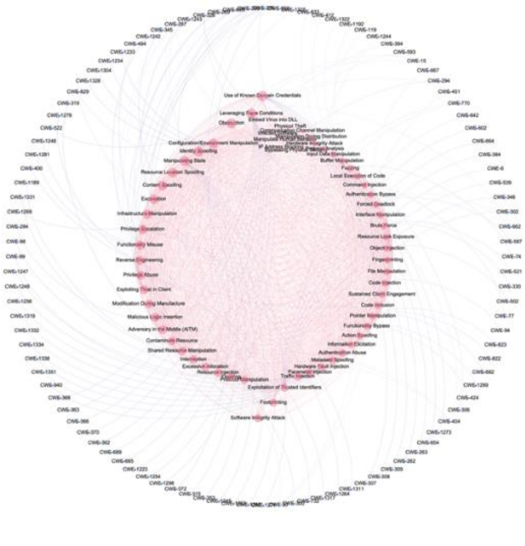


Figure 8. Visualization of only-connected attack patterns and their associated weaknesses

Table 4. Characteristics of AP network model

# of attack patterns and CWEs	# of relations	Average degree	Diameter	Density	Average clustering coefficient	Average path length
167	2223	26.623	4	0.16	0.881	2.344

Table 5. Top 10 attack patterns in CAPEC and CWE models according to the betweenness centrality

Rank	Attack pattern	Status	Likelihood of attack	Typical severity	Degree	Betweenness
1st	Leveraging Race Conditions	Stable	High	High	101	0.13145
2nd	Exploitation of Trusted Identifiers	Stable	High	High	73	0.09759
3rd	Use of Known Domain Credentials	Stable	High	High	103	0.09732
4th	Hardware Fault Injection	Stable	Low	High	69	0.09491
5th	Manipulating State	Stable	Medium	High	98	0.08985
6th	Configuration/Environment Mani.	Draft	N/A	Medium	100	0.06143
7th	Pointer Manipulation	Draft	N/A	Medium	61	0.05952
8th	Adversary in the Middle (AiTM)	Stable	High	Very High	82	0.05681
9th	Forced Deadlock	Stable	Low	High	58	0.05657
10th	Functionality Misuse	Stable	Medium	Medium	90	0.0382

Table 6. Top 10 attack patterns in CAPEC and CWE models according to the clustering coefficient

Rank	Attack pattern	Status	Likelihood of attack	Typical severity	Degree	Clustering coefficient
1st	Manipulate Human Behavior	Stable	Medium	Medium	9	1.0
2nd	IP Address Blocking	Draft	Low	High	4	1.0
3rd	Input Data Manipulation	Draft	N/A	Medium	55	0.9592760180995475
4th	Buffer Manipulation	Draft	High	Very High	55	0.9577677224736049
5th	Command Injection	Stable	Medium	High	57	0.9577677224736049
6th	File Manipulation	Draft	N/A	Medium	60	0.9562594268476622
7th	Bypassing Physical Security	Draft	N/A	N/A	14	0.9560439560439561
8th	Physical Theft	Draft	N/A	N/A	14	0.9560439560439561
9th	Information Elicitation	Draft	N/A	Low	65	0.9492017416545718
10th	Code Inclusion	Stable	Medium	Very High	61	0.9462989840348331

Table 7. Top 10 weaknesses in CAPEC and CWE models according to the clustering coefficient

Rank	Weakness ID	Weakness	Clustering coefficient	Likelihood to exploit
1st	CWE-287	Improper Authentication	1.0	High
2nd	CWE-20	Improper Input Validation	1.0	High
3rd	CWE-404	Improper Resource Shutdown or Release	1.0	Medium
4th	CWE-662	Improper Synchronization	1.0	N/A
5th	CWE-770	Allocation of Resources without Limits or Throttling	1.0	High
6th	CWE-451	User Interface Misrepresentation of Critical Info.	1.0	N/A
7th	CWE-667	Improper Locking	1.0	N/A
8th	CWE-200	Exposure of Sensitive Info. to an Unauthorized Actor	1.0	High
9th	CWE-290	Authentication Bypass by Spoofing	1.0	N/A
10th	CWE-829	Inclusion of Func. from Untrusted Control Sphere	1.0	N/A

4. DISCUSSION

Many of the CAPECs have different ratings based on likelihood and severity. Based on this work, some of these CAPECs are discussed based on cybersecurity perspective. CAPEC-416 (Manipulate Human Behavior), adversaries often rely on social engineering attacks when no weaknesses are found in systems and applications. People are considered one of the weak points in any enterprise. In this kind of attack, the adversary exploits points of interest in target individuals to prompt them to take a certain action, such as visiting a malicious website or downloading and running a Trojan, in order to gain access to the target computer or network. As shown in this work, the CAPEC database rated the likelihood as medium and the severity as medium as well. We agreed on the likelihood to be a medium as a Social Engineer attack is one of adversary's options to perform if they have enough information such as employee emails or other useful information. Despite CAPEC rating the severity as medium, we believe its rating could be higher, considering that if the objective of the attack is success, it could result in significant damage. For instance, let’s say that the attacker is attempting to breach an organization’s network. Once the target grants access to the attacker, the adversary gains control over the compromised employee's computer as well as to entire target network. This access level enables the attacker to execute further internal attacks. The work of [24] provides advantages of social engineering attacks and how adversaries can exploit targets using techniques that refer to this kind of attack as of serious threat.

CAPEC-590 (IP Address Blocking), the end-goal of an IP address blocking attack is to prevent access to services or applications hosted on a specific IP address by blocking incoming or outgoing traffic to that address. Adversaries conduct deep packet inspection and craft a network packet that causes the dropping of target traffic or connections. This kind of attack could be performed against various network protocols such as TCP, DNS, HTTP, and other protocols. The CAPEC database classifies the IP address blocking attack as highly severe, and we wholeheartedly concur with this assessment. Thus, if the attacker succeeds in this attack, any access from other clients will be dropped, resulting in disrupting or denying legitimate users' access to the services or applications hosted on the target IP address. In order to perform the IP address blocking, the attacker requires access to specific network devices, such as Firewall, Routers or gaining access to internal infrastructure network. Consequently, CAPEC rated the likelihood of this attack as low, and we totally agreed with assessment.

CAPEC-153 (Input Data Manipulation), many applications rely on user interaction or feedback, and that’s often facilitated through users’ input. Thus, designing an application that requires user input must be coded carefully. Developers can use various sanitization techniques to handle input validation from users, however, the input data manipulation still occurs. Input data manipulation emerges as one of the top ten weaknesses discussed in this paper. The reason for making it common in this paper is that input data manipulation weaknesses are connected with other weaknesses and vulnerabilities. The input data manipulation main cause of vulnerabilities such, command injection, code injection, and cross-site scripting [25], [26]. CAPEC classified the likelihood of the attack as none-applicable (N/A), and we are unable to determine the reason behind this classification. As previously mentioned, the input data manipulation could occur in numerous mobile, desktop, and web applications. We advocate for re-classifying the likelihood of input data manipulation weakness at least as medium. While CAPEC classification of input data manipulation as a medium severity is a reasonable assessment, we fully advocate.

CAPEC-248 (Command Injection), adversaries are executing command injection attacks by injecting arbitrary commands into a vulnerable application. The ultimate aim of the attack is to execute system or other types of commands to gain control of the host operating system. The command injection occurs due to the lack of proper input validation and sanitization. Once the vulnerable application receives the command sent by the attacker, it passes it to the operating system, database, or other components. Command injection vulnerabilities come in various types depending on the platform or applications, including direct execution of Windows or Linux commands, uploading of malicious files into the server's runtime environment, the exploitation of configuration file flaws like XML external entities (XXE), and others. Command injection attacks are often rated with high severity when they occur. According to the SANS Institute, a trusted resource for cybersecurity training and research, command injection vulnerabilities are listed among the top 25 vulnerabilities that need to be protected against. The CAPEC dataset indicates that command injection is classified as having a medium likelihood of being found due to the training and education available to protect against it. Additionally, CAPEC rates it with high severity. We fully support their scoring system and believe it could be rated as critical if it occurs since the attacker will have control over the system through the vulnerable application.

CAPEC-175 (Code Inclusion), similar to command injection attacks, adversaries inject malicious code into a vulnerable application to execute commands or retrieve local files on a host machine. The main distinction between command injection and code inclusion is that command injection involves executing operating system commands, while code inclusion involves injecting code that compiles, runs, and renders on the host machine.

5. CONCLUSION

This work evaluated the severity and likelihood of cyberattacks using concepts inspired by complex networks. The CAPECs dataset was converted into a network model, where each attack pattern is a node and edges represent their relationships. The dataset included 559 attack patterns and 1921 connections. Using network metrics, the overall network was assessed and a rank of CAPECs was generated based on a network perspective. This ranking was compared with the existing CAPEC ranking system, revealing significant insights about attack likelihood and severity. The findings suggest that the network perspective should be considered in CAPEC's ranking system, offering valuable insights for security architects. As future work, this study can be extended by combining the current ranking system of MITRE and network perspective. The result of this combination can be reviewed by cybersecurity experts aiming to validate the approach.

ACKNOWLEDGEMENTS

The authors would like to thank the support provided by the Computer Centre and the Department of Computer Science, College of Computer Science and Mathematics, University of Mosul.

FUNDING INFORMATION

There is no funding supporting this research.

AUTHOR CONTRIBUTIONS STATEMENT

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Ahmed Salih Hasan	✓		✓			✓	✓	✓	✓	✓			✓	
Yasir F. Mohammed	✓			✓	✓	✓		✓	✓	✓				
Basim Mahmood	✓	✓	✓	✓	✓	✓				✓	✓	✓	✓	

C :	**C**onceptualization	I :	**I**nvestigation	Vi :	**Vi**sualization
M :	**M**ethodology	R :	**R**esources	Su :	**Su**pervision
So :	**So**ftware	D :	**D**ata Curation	P :	**P**roject administration
Va :	**Va**lidation	O :	Writing - **O**riginal Draft	Fu :	**Fu**nding acquisition
Fo :	**Fo**rmal analysis	E :	Writing - Review & **E**diting		

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

INFORMED CONSENT

There is no informed consent associated with this research.

ETHICAL APPROVAL

This is not applicable in this research.

DATA AVAILABILITY

The data used in this research was obtained from publicly available sources, including the CWE, CVE, and CAPEC.

REFERENCES

[1] E. Zio, “Challenges in the vulnerability and risk analysis of critical infrastructures,” *Reliability Engineering and System Safety*, vol. 152, pp. 137–150, Aug. 2016, doi: 10.1016/j.ress.2016.02.009.

[2] Z. Mohamad Fadli, S. S. Yong, L. K. Kee, and G. H. Ching, “Cyber attack awareness and prevention in network security,” *International Journal of Informatics and Communication Technology (IJ-ICT)*, vol. 11, no. 2, p. 105, Aug. 2022, doi: 10.11591/ijict.v11i2.pp105-115.

[3] A. A. Ojugo, P. O. Ejeh, O. C. Christopher, A. O. Eboka, and F. U. Emordi, “Improved distribution and food safety for beef processing and management using a blockchain-tracer support framework,” *International Journal of Informatics and Communication Technology*, vol. 12, no. 3, pp. 205–213, Dec. 2023, doi: 10.11591/ijict.v12i3.pp205-213.

[4] G. Spanos and L. Angelis, “Impact metrics of security vulnerabilities: analysis and weighing,” *Information Security Journal*, vol. 24, no. 1–3, pp. 57–71, Jun. 2015, doi: 10.1080/19393555.2015.1051675.

[5] T. L. Nielsen, J. Abildskov, P. M. Harper, I. Papaeconomou, and R. Gani, “The CAPEC database,” *Journal of Chemical and Engineering Data*, vol. 46, no. 5, pp. 1041–1044, Mar. 2001, doi: 10.1021/je000244z.

[6] C. Vulnerabilities, “Common vulnerabilities and exposures,” *The MITRE Corporation*, 2005. https://cve.mitre.org/index.html (accessed Jan. 01, 2024).

[7] S. J. Mohammed and D. B. Taha, “From cloud computing security towards homomorphic encryption: a comprehensive review,” *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 9, no. 4, pp. 1–10, Aug. 2021, doi: 10.12928/telkomnika.v19i4.16875.

[8] B. Martin, M. Brown, A. Paller, D. Kirby, and S. Christey, “CWE/SANS top 25 most dangerous software errors,” *Common Weakness Enumeration*, 2011.

[9] T. S. Riera, J. R. B. Higuera, J. B. Higuera, J. J. M. Herraiz, and J. A. S. Montalvo, “A new multi-label dataset for Web attacks CAPEC classification using machine learning techniques,” *Computers and Security*, vol. 120, p. 102788, Sep. 2022, doi: 10.1016/j.cose.2022.102788.

[10] R. Albert and A. L. Barabási, “Statistical mechanics of complex networks,” *Reviews of Modern Physics*, vol. 74, no. 1, pp. 47–97, Jan. 2002, doi: 10.1103/RevModPhys.74.47.

[11] B. Mahmood and R. Menezes, “United states congress relations according to liberal and conservative newspapers,” in *Proceedings of the 2013 IEEE 2nd International Network Science Workshop, NSW 2013*, Apr. 2013, pp. 98–101, doi: 10.1109/NSW.2013.6609201.

[12] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. U. Hwang, “Complex networks: Structure and dynamics,” *Physics Reports*, vol. 424, no. 4–5, pp. 175–308, Feb. 2006, doi: 10.1016/j.physrep.2005.10.009.

[13] S. H. Strogatz, “Exploring complex networks,” *Nature*, vol. 410, no. 6825, pp. 268–276, Mar. 2001, doi: 10.1038/35065725.

[14] R. Miyata, H. Washizaki, K. Sumoto, N. Yoshioka, Y. Fukazawa, and T. Okubo, “Identifying missing relationships of CAPEC attack patterns by transformer models and graph structure,” in *Proceedings - 2023 IEEE/ACM 1st International Workshop on Software Vulnerability, SVM 2023*, May 2023, pp. 14–17, doi: 10.1109/SVM59160.2023.00008.

[15] E. Doynikova and I. Kotenko, “An automated graph based approach to risk assessment for computer networks with mobile components,” in *Communications in Computer and Information Science*, vol. 797, Springer Singapore, 2018, pp. 95–106.

[16] Z. Younis and B. Mahmood, “An in-depth vision to hardware design security vulnerabilities,” *Jordanian Journal of Computers and Information Technology*, vol. 8, no. 1, pp. 33–44, 2022, doi: 10.5455/jjcit.71-1635517841.

[17] B. Mahmood, “Prioritizing CWE/SANS and OWASP vulnerabilities: A network-based model,” *International Journal of Computing and Digital Systems*, vol. 10, no. 1, pp. 361–372, Feb. 2021, doi: 10.12785/ijcds/100137.

[18] C. Grigoriádis, “Identification and assessment of security attacks and vulnerabilities, utilizing CVE, CWE and CAPEC,” *Diss. University of Piraeus (Greece)*, 2019.

[19] K.-P. Grammatikakis and N. Kolokotronis, “Attack graph generation,” in *Cyber-Security Threats, Actors, and Dynamic Mitigation*, CRC Press, 2021, pp. 281–334.

[20] V. Pham and T. Dang, “CVExplorer: multidimensional visualization for common vulnerabilities and exposures,” in *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*, Dec. 2018, pp. 1296–1301, doi: 10.1109/BigData.2018.8622092.

[21] J. Yin, W. Hong, H. Wang, J. Cao, Y. Miao, and Y. Zhang, “A compact vulnerability knowledge graph for risk assessment,” *ACM Transactions on Knowledge Discovery from Data*, vol. 18, no. 8, pp. 1–17, Jul. 2024, doi: 10.1145/3671005.

[22] A. Sejfia and N. Medvidovic, “Strategies for pattern-based detection of architecturally-relevant software vulnerabilities,” in *Proceedings - IEEE 17th International Conference on Software Architecture, ICSA 2020*, Mar. 2020, pp. 92–102, doi: 10.1109/ICSA47634.2020.00017.

[23] W. Cai, J. Chen, J. Yu, and L. Gao, “A software vulnerability detection method based on deep learning with complex network analysis and subgraph partition,” *Information and Software Technology*, vol. 164, p. 107328, Dec. 2023, doi: 10.1016/j.infsof.2023.107328.

[24] K. Krombholz, H. Hobel, M. Huber, and E. Weippl, “Advanced social engineering attacks,” *Journal of Information Security and Applications*, vol. 22, pp. 113–122, Jun. 2015, doi: 10.1016/j.jisa.2014.09.005.

[25] V. B. Livshits and M. S. Lam, “Finding security vulnerabilities in Java applications with static analysis,” *USENIX Security Symposium*, 2005.

[26] T. Hamed, R. Dara, and S. C. Kremer, “Network intrusion detection system based on recursive feature addition and bigram technique,” *Computers and Security*, vol. 73, pp. 137–155, Mar. 2018, doi: 10.1016/j.cose.2017.10.011.

BIOGRAPHIES OF AUTHORS

Ahmed Salih Hasan received his first degree from the University of Mosul, Iraq, in 2007. He also has a Master's degree from the School of Computer Science in Universiti Sains Malaysia (USM), Malaysia in 2012. He is currently a lecturer at University of Mosul. His main research interests are computing networks. He can be contacted at email: ahmed_salih_h@uomosul.edu.iq.

Yasir F. Mohammed obtained his Ph.D. in Computer Science from the University of Arkansas, USA, in 2021. He currently works at the Cybersecurity Unit, Computer Center at the University of Mosul. His areas of interest are related to the cybersecurity field. He can be contacted at: yasirfaraj@uomosul.edu.iq.

Basim Mahmood obtained his Ph.D. degree in Computer Science from Florida Institute of Technology, Melbourne, USA, in 2015. He is currently a computer science associate professor at the ICT Research Unit, Computer Center, University of Mosul, Iraq. His main fields of interest are complex networks, data mining, and big data analysis. He can be contacted at: bmahmood@uomosul.edu.iq.