Important Member Discovery of Attribution Trace Based on Relevant Circle (Short Paper)

Jian Xu[1,2], Xiaochun Yun[1,2,3(✉)], Yongzheng Zhang[1,2], and Zhenyu Cheng[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
{xujian,zhangyongzheng,chengzhenyu}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100093, China
[3] National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100093, China
yunxiaochun@cert.org.cn

Abstract. Cyberspace attack is a persistent problem since the existing of internet. Among many attack defense measures, collecting information about the network attacker and his organization is a promising means to keep the cyberspace security. The exposing of attackers halts their further operation. To profile them, we combine these retrieved attack related information pieces to form a trace network. In this attributional trace network, distinguishing the importance of different trace information pieces will help in mining more unknown information pieces about the organizational community we care about. In this paper, we propose to adopt relevant circle to locate these more important vertices in the trace network. The algorithm first uses Depth-first search to traverse all vertices in the trace network. Then it discovers and refines relevant circles derived from this network tree, the rank score is calculated based on these relevant circles. Finally, we use the classical 911 covert network dataset to validate our approach.

Keywords: Importance rank · Network attribution · Relevance

1 Introduction

According to the report of cybersecurity and cyberwar [11], the cyberspace security is the first class network security problem. In order to expose attackers and their organization behind the scenes, relating these information pieces about the organization community to constitute a trace network for comprehensively profiling the attackers is a very promising means against modern cyberspace threats, such as APT (Advanced persistent threat) [17]. This new emerging threat is a set of stealthy and continuous computer intrusion processes. While generally, it is hard to directly halt these intrusion operations, it is possible to profile the

H. Gao et al. (Eds.): CollaborateCom 2018, LNICST 268, pp. 222–232, 2019.
https://doi.org/10.1007/978-3-030-12981-1_16

attackers as an organization community through continuously monitoring related information pieces about them and to detect their intrusion to certain internet devices by means of seeking for IOCs (Indicators of compromise). Monitoring and mining of the attack network is often a long and ongoing process resulting in a gradual accumulation of information. Over time, as more information is uncovered, new vertices and relations are added [15]. To better facilitate this process, it is significant to distinguish important vertices in the attributional trace network from other relatively less matter ones. Because important vertices in the network are strong relevant to other undiscovered information pieces about the attack organization, and the attributional trace network constituted of information pieces is large, thus processing all these information pieces is inefficient and may lead the investigation to trivial path.

Our paper proposes to rank vertices in an attributional trace network through detecting relevant circles. The effectiveness of this method is demonstrated by applying it to the classical 911 attack dataset. Overall, we make the following contributions.

- We propose relevant circle into ranking vertex importance, derive important measures, such as minimized relevant circles, and propose criterions to score the vertex importance. Relevant circles are effective measurement of relevance among information pieces.
- We implement and explain the key algorithms to rank vertices based on relevant circle, including construction of the network tree, regeneration of relevant circles, deduction of minimized relevant circles and the score algorithm.
- We evaluate the effectiveness of this approach using the 911 covert network dataset. The result shows the key members found by our proposed model are great investigation entry of the covert network. These members lead to more relevant information about the network.

The remainder of this paper is structured as follows: In Sect. 5, we introduce the related researches. Section 2 will present the concept of relevant circle and its derived measures. We implement the corresponding algorithms in Sect. 3. Experiments are conducted in Sect. 4. We finally draw our conclusion in Sect. 6.

2 Method

In this Section, we first introduce the relevance definition, followed by proposing the relevant circle. Minimized relevant circle derived from relevant circle is discussed, and the score rules of rank importance are presented finally.

During the profiling process, information pieces are connected by many means. They may appear in the same context or have been searched in Google by users as a combination. For example, if the information pieces are about people, they may be connected by attending the same conference, living in the same area or they might attend the same college in the past. Two information pieces are relevant when one information piece is being connected to another information piece in a way that makes it necessary to consider the second information piece when considering the first one.

Information pieces are extremely relevant to others especially when the inter relevant relationships form a circle, indicating a strong evidence to confirm the relevance among the vertices residing in this circle. For example, in Fig. 1, network A contains no relevant circle, all vertices in network A are relevant to others through cascading the relevant relationship. While network B contains relevant circles. Every vertex in the relevant circles re-confirms relevance of others.

Sometimes a vertex is located in multiple relevant circles. And one relevant circle is contained in another one. For example, in Fig. 2, circle B is a part of circle A, therefore circle A is not a minimized relevant circle when considering the existing of circle B. On this case, we break relevant circle A into smaller relevant circles B and C, and deduct circle A to circle C. We define that relevant circle B and C are minimized relevant circles. The minimized relevant circle is defined as a relevant circle that is not contained in other relevant circles. It is important to find the minimized relevant circle in the attributional trace network because they influence the rank score of vertices located in them.

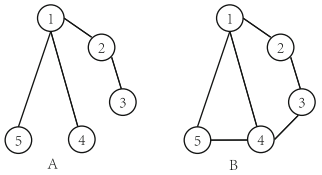

Fig. 1. Relevant circle

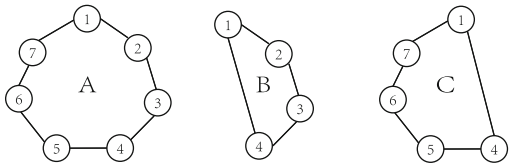

Fig. 2. Relevant circle deduction

Also there are situations when one vertex performs as the joint of multiple minimized relevant circles. These joint vertices are the central of the organization we are investigating. They may lead to more yet unknown but relevant information pieces if further researches are conducted on them. The more minimized relevant circles one vertex resides in, the more important this vertex becomes. For example, in Fig. 3, there exist three minimized relevant circles, they are 1–2–3–4, 1–4–5 and 1–6–7. Vertex 1 locates in three circles, and vertex 4 in two circles, the rest vertices are in one circle.

What's more, the rank score is also influenced by the size of the minimized relevant circle. Vertices in the smaller circle are assigned with bigger score. Because if the circle size is small, it denotes that these vertices are a compact community with tight connections. In Fig. 3, relevant circle 1–4–5 is smaller than 1–2–3–4, thus vertex 1, 4, 5 are more relevant to each other than vertex 1, 2, 3, 4. These two criterions are efficient to distinguish important vertex from other ones.

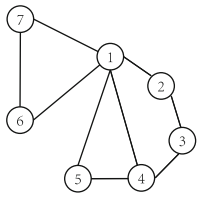

Fig. 3. Multiple circles

We formulate the rank score according to these rules for the importance ranking of vertices. The initial score of each vertex is zero. And the score piles up when a vertex is discovered locating in more relevant circles. The vertex score is calculated according to Formula 1.

$$Score = \sum_{i=1}^{n} \frac{1}{l_i} \qquad l_i \geq 3 \tag{1}$$

Where n is the number of minimized relevant circles that the vertex is found resided in. l_i is the length of the minimized relevant circle and the length is required to be greater than 3. For example, in Fig. 3, vertex 1 is in three minimized relevant circles. And the lengths of these circles are 3, 3 and 4 respectively. Therefore, according to Formula 1, the importance score of vertex 1 is 11/12. And it is the most important vertex in this network.

3 Implementation

In this Section, we present the key algorithms to implement our proposed method. The main process first constructs the network tree. Then, relevant circles are reconstructed from the network tree. Third, relevant circles are further deducted to minimized relevant circles. We finally calculate the rank score of vertices in these minimized relevant circles.

We construct the network tree by means of iteratively calling Algorithm 1. It employs Depth-first search (DFS), and records relevant circles along the searching process. The parent list stores the network tree structure. Algorithm 1 accepts the start vertex and its parent vertex as inputs. It first appends the start vertex into the visited vertex list, then processes every vertex of the start vertex's adjacent vertices excluding the parent vertex. For every vertex, if it is already included in the visited list, it means that the vertex pair (v_{start} and v_i) is contained in a relevant circle and we record this pair into backtrack list for later usage. Otherwise if this vertex is not visited yet, we record the parent-child relation into the parent list, and iteratively call Algorithm 1. This algorithm will guarantee that the search firstly handles vertices in the deepest layer, and then goes back layer by layer to the root vertex.

Algorithm 1. Construct network tree using DFS algorithm

Input: $G(V, E)$, v_{start}, v_{parent}
Output: *parent_list*, *visited*, *backtrack*

```
visited.append(v_start)
V ← v_start.adjacents exclude v_parent
if V = ∅ then
    return
else
    for all v_i ∈ V do
        if v_i ∈ visited then
            backtrack.append([v_start, v_i])
        else
            parent_list.append([v_start, v_i])
            find_tree_DFS(G(V, E), v_i, v_start)
        end if
    end for
end if
```

Algorithm 2. Regenerate relevant circles

Input: v_{root}, *parent_list*, *backtrack*
Output: *circle_list*

```
for all item ∈ backtrack do
    circle ← ∅
    circle.append(item[0])
    while item[0] ≠ item[1] do
        item[0] = parent_list(item[0])
        if item[0] ≠ v_root then
            circle.append(item[0])
        else
            circle.append(item[1])
            each[0] = each[1]
            each[1] = v_root
        end if
    end while
    circle_list.append(circle)
end for
```

After the network tree building, Algorithm 2 makes use of network tree information from v_{root}, *parent_list*, combined with the vertex pairs in backtrack list generated from Algorithm 1 to reform relevant circles. Each item in the backtrack list will be utilized to generate a relevant circle. For each vertex pair in backtrack, the first vertex in the item is pushed into the circle. If the first vertex is not the same as the second vertex, Algorithm 2 will find the parent vertex of the first vertex in the parent list. Then, it will check whether the process is encountered to v_{root}, and push the first vertex into circle if the parent vertex is not the root vertex. Otherwise, it pushes the second vertex, exchanges the first and second vertices, and replaces the second vertex with v_{root}. This loop will continue until the first vertex is the same as the second vertex. The circle stores the relevant circle at this time. Finally, this circle is pushed into the circle list.

Algorithm 3. Deduct circle list

Input: *circle_list*
Output: *circle_list*

```
for all circle pairs ∈ circle_list do
    if circleA ⊂ circleB then
        break circleB into two small ones
        circleB.update(small circle that is
        not circleB)
    end if
end for
```

Algorithm 4. Score

Input: *circle_list*
Output: *scores*

```
scores ← 0
for all circle ∈ circle_list do
    for all v_i ∈ circle do
        scores(v_i)+ = 1/length(circle)
    end for
end for
```

Algorithm 3 will deduct the circle list in order to find the minimized relevant circles. It checks every circle pair in the circle list. If one circle in the pair is a part of the other one, the bigger circle will be broken into two distinguished relevant circles, including the smaller relevant circle. Finally, the bigger circle is updated with the deducted relevant circle. This process will continue running until no circle contains another circle pair in the circle list.

Algorithm 4 calculates the rank score for vertices in the attributional trace network. It initiates the score of all vertices to 0, and then enumerates all the minimized relevant circles in the circle list. For each vertex in the circle, the vertex's score is added by $\frac{1}{length(circle)}$ according to Formula 1.

The variables we used during these implementations are described in Table 1.

Table 1. Variable summary

Module	Variable	Explanation	Type
Construct network tree using DFS	G(V, E)	Store the adjacent vertices of each vertex in the network	Network
	v_start	The start vertex of each running	Vertex
	v_parent	The parent vertex of *v_start*	Vertex
	visited	Keep track of visited vertices	List
Regenerate relevant circles	*parent_list*	Record the parent-child relations of the network tree	List
	backtrack	Record the vertex pairs that may contain relevant circles	List
	v_root	The root vertex where the DFS search begins	Vertex
	circle_list	Record the relevant circles	List

4 Experiment

In this Section, we validate the effectiveness of the method and implementation we proposed using the classic 911 covert network dataset. We first introduce the dataset and its statistical characters, and then preprocess the dataset for the input to be compatible with our implementation. Algorithms in Sect. 3 are experimented to search for the minimized relevant circles through the network, and rank scores of vertices are calculated. We finally demonstrate and discuss the result.

In order to demonstrate the method and its implementations we proposed, we employ the famous network dataset of the terrorists involved in the 911 bombing of the World Trade Centers in 2001. The vertex connection types of this dataset range from 'attend the same school' to 'on the same plane'. It is based on open source intelligence (OSI) such as news reports, and tidied by Krebs [7].

The dataset consists 61 vertices, representing the members who are believed associated to this operation. The whole network is not densely connected, and the density is 0.08, exhibiting the secrecy characteristic of a covert terrorist network. This mitigates the consequence brought when a member is captured or compromised. The statistical characters of this dataset are depicted in Table 2.

Despite the connection sparsity, the network diameter is 5, indicating this covert network is organized efficiently. The communication through this network only requires 5 relays in the worst situation. In common cases, the information path is 2.92, which is the average shortest path in Table 2. These characters profile a covert network that although quite invisible, maintains strong communications among its members.

In the original dataset, there exist some incompatible data. Therefore, we preprocess the connections to mitigate the incompatibility. There are 131 connections among these members, of which 64 connections are directed. Because the relevant circle is defined on undirected relations, we consider these directed connections as undirected, and make connections to ensure the relations are bidirection. In the dataset, vertex 32, Rayed Mohammed Abdullah, has connection to itself, and we remove this self loop.

After the dataset preprocess, it contains 61 members and 131 connections. Their relationships are demonstrated in Fig. 4.

Table 2. Dataset summary

Characters	Value
Density	0.08
Avg. degree	4.9
Diameter	5
Avg. shortest path	2.92

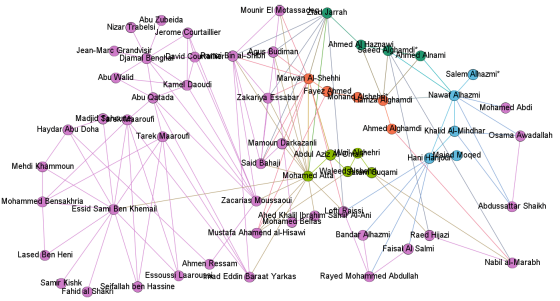

Fig. 4. 911 Covert network (Color figure online)

In Fig. 4, orange circles denote people who were in airplane UA #175 heading to WTC South, green circles denote people who were in airplane UA #93 heading to Pennsylvania, blue circles denote people who were in airplane AA #77 heading to Pentagon, chartreuse green circles denote people who were in airplane AA #11 heading to WTC North and pink circles represent other people who were associated in this event but were not in any hijacked planes.

For the purpose to guarantee the network tree built by Algorithm 1 maintains a shallow tree structure, we utilize the score rank implementations to build the network tree from the vertex of which the degree is the greatest. This vertex is Mohamed Atta with vertex degree as 15. we see him as our topic vertex to start the investigation. This also ensures that these found minimized relevant circles are most around our investigation topic. The rank of these members is depicted in Table 3.

Table 3. Rank result

1	Mohamed Atta	15	Tarek Maaroufi	29	Waleed Alshehri	43	Ahmed Alghamdi
2	Ziad Jarrah	16	Takek Maaroufi	30	Haydar Abu Doha	44	Mohand Alshehri
3	Marwan Al-Shehhi	17	Imad Eddin Baraat Yarkas	31	Mehdi Khammoun	45	Wail Alshehri
4	Ramzi Bin al-Shibh	18	Jerome Courtaillier	32	Ahmed Al Haznawi	46	Bandar Alhazmi
5	Essid Sami Ben Khemail	19	Kamel Daoudi	33	Fayez Ahmed	47	Faisal Al Salmi
6	Abu Qatada	20	Hani Hanjour	34	Ahmed Alnami	48	Lased Ben Heni
7	Said Bahaji	21	Lofti Raissi	35	Raed Hijazi	49	Madjid Sahoune
8	Djamal Benghal	22	Nabil al-Marabh	36	Mamoun Darkazanli	50	Ahed Khalil Ibrahim Samir Al-Ani
9	Zacarias Moussaoui	23	Khalid Al-Mihdhar	37	Osama Awadallah	51	Mohamed Belfas
10	Saeed Alghamdi	24	Agus Budiman	38	Abdussattar Shaikh	52	Abdul Aziz Al-Omari
11	Zakariya Essabar	25	Satam Suqami	39	Abu Walid	53	Ahmen Ressam
12	Mohammed Bensakhria	26	Mounir El Motassadeq	40	Seifallah ben Hassine		
13	Nawaf Alhazmi	27	Rayed Mohammed Abdullah	41	Mustafa Ahamend al-Hisawi		
14	Hamza Alghamdi	28	David Courtaillier	42	Essoussi Laaroussi		

This table includes 53 out of 61 members. 8 members are excluded because they are not found in any minimized relevant circle. From the table, we can figure out that Mohamed Atta is the most important member in this network. It resides in 14 minimized relevant circles, and its rank score is 4. This means the exposure of Mohamed Atta would lead to more relevant information about this covert network. Wikipedia shows that Mohamed Atta is the ringleaders of this attack. The top 3 members, Mohamed Atta, Ziad Jarrah and Marwan Al-Shehhi were 3 leaders separated in three hijacked planes. Although Ramzi Bin al-Shibh, as the fourth important member, did not directly involved in hijacking the planes, wikipedia shows he was a key facilitator of this attack, contributing great to the achievement of this operation. They are all the important members functioning as the perfect investigation entry points for this covert network.

5 Related Work

Vertex importance rank is a well-researched area in SNA (Social Network Analysis) [2,3]. Its optimization as sub research areas are also well established [6,10]. Most of these works are based on the vertex degree and the centricity. While some others are based on the PageRank and the connection importance [14]. Recently, epidemic models are also employed to measure vertex importance. We will introduce these categories respectively in the following.

5.1 Connection-Centric Approaches

Farley presented mathematical analysis of Al Qaeda organization. They used the order theory to quantify the degree to which the organization is still able to work, and determined these important vertices that are needed to be removed in order to neutralize the network [4]. They proposed the break the chains model as to break the connection for separating the important commanders from other vertices in the network.

Taha [12] presented a system called SIIMCO (System for Identifying the Influential Members of a Criminal Organization). It created network from Mobile Communication Data (MCD) and combined the vertex degree and its edge weight to rank the vertex importance. The result of their system showed improvement compared with CrimeNet Explorer [16] and LogAnalysis [5]. Taha also proposed to use the spanning tree of the network for identifying their leaders [13].

5.2 Vertex-Centric Approaches

Memon proposed a vertex centric measure that considers the number of connections incident to vertex along with connection weight. The importance of each vertex is determined by the overall vertex centrality [9].

Butt et al. employed hybrid framework to predict important vertices in the covert network. Their system calculates centrality measures as the features, and hybrid classifiers, such as k-Nearest Neighbors and Support Vector Machine are applied to figure out these key players [1].

5.3 Community-Centric Approaches

Langohr et al. proposed probabilistic similarity measure for vertices, and employed both k-medoids and hierarchical clustering methods to find the community. They regarded the representation of each community as the important vertex [8].

Our approach is different from theirs as we consider the connection structure of the attributional trace network and exploit the inter connection pattern to work as the foundation to rank vertices.

6 Conclusions

In this paper, we proposed a relevant circle-based approach to rank and discover the important vertices in the attributional trace network. The top rank vertices are most relevant to the investigation central, which lead the information trace-back to discover more still unknown relevant information. We also introduced the implementation of pivotal algorithms. Lastly, we demonstrate this method is valuable to mining these key players participating the 911 terrorist crime as a covert network. In further work, we would also like to research on trace networks which are featured by directed relation and connections with weight.

Acknowledgment. This work was supported by the National Natural Science Foundation of China (No. U1736218).

References

1. Butt, W.H., Akram, M.U., Khan, S.A., Javed, M.Y.: Covert network analysis for key player detection and event prediction using a hybrid classifier. Sci. World J. **2014**, 13 (2014). 615431
2. Chitrapura, K.P., Kashyap, S.R.: Node ranking in labeled directed graphs. In: Thirteenth ACM International Conference on Information and Knowledge Management, pp. 597–606 (2004)
3. Dasgupta, S., Prakash, C.: Intelligent detection of influential nodes in networks. In: International Conference on Electrical, Electronics, and Optimization Techniques (2016)
4. Farley, J.D.: Breaking Al Qaeda cells: a mathematical analysis of counterterrorism operations (a guide for risk assessment and decision making). Stud. Conflict Terrorism **26**(6), 399–411 (2003)
5. Ferrara, E., Meo, P.D., Catanese, S., Fiumara, G.: Detecting criminal organizations in mobile phone networks. Expert Syst. Appl. **41**(13), 5733–5750 (2014)
6. Halappanavar, M., Sathanur, A.V., Nandi, A.K.: Accelerating the mining of influential nodes in complex networks through community detection, pp. 64–71 (2016)
7. Krebs, V.E.: Mapping networks of terrorist cells, pp. 43–52 (2002)
8. Langohr, L., Toivonen, H.: Finding representative nodes in probabilistic graphs. In: Berthold, M.R. (ed.) Bisociative Knowledge Discovery. LNCS (LNAI), vol. 7250, pp. 218–229. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31830-6_15
9. Memon, B.R.: Identifying important nodes in weighted covert networks using generalized centrality measures. In: Intelligence and Security Informatics Conference, pp. 131–140 (2012)
10. Sheikhahmadi, A., Nematbakhsh, M.A., Shokrollahi, A.: Improving detection of influential nodes in complex networks. Physica A Stat. Mech. Appl. **436**, 833–845 (2015)
11. Singer, P.W.: Cybersecurity and Cyberwar: What Everyone Needs to Know. Oxford University Press, Oxford (2014)
12. Taha, K., Yoo, P.D.: SIIMCO: a forensic investigation tool for identifying the influential members of a criminal organization. IEEE Trans. Inf. Forensics Secur. **11**(4), 811–822 (2016)

13. Taha, K., Yoo, P.D.: Using the spanning tree of a criminal network for identifying its leaders. IEEE Trans. Inf. Forensics Secur. **PP**(99), 1 (2017)
14. Wiil, U.K., Gniadek, J., Memon, N.: Measuring link importance in terrorist networks. In: International Conference on Advances in Social Networks Analysis and Mining, pp. 225–232 (2010)
15. Xu, J., Yun, X., Zhang, Y., Sang, Y., Cheng, Z.: NetworkTrace: probabilistic relevant pattern recognition approach to attribution trace analysis. In: 2017 IEEE Trustcom/BigDataSE/ICESS, pp. 691–698, August 2017. https://doi.org/10.1109/Trustcom/BigDataSE/ICESS.2017.301
16. Xu, J.J., Chen, H.: Crimenet explorer: a framework for criminal network knowledge discovery. ACM Trans. Inf. Syst. **23**(2), 201–226 (2005)
17. Wei, Z., Yang, S., Wenwu, C.: A game model of APT attack for distributed network. In: Xhafa, F., Caballé, S., Barolli, L. (eds.) 3PGCIC 2017. LNDECT, vol. 13, pp. 224–234. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-69835-9_21