ISSN:2321-1156
International Journal of Innovative Research in Technology & Science(IJIRTS)

# DETECTION OF SIMILARITIES IN CYBER THREATS THROUGH OSINT

Seulgi Lee, Korea Internet & Security Agency; NakHyun Kim, Korea Internet & Security Agency; Hyeisun Cho, Korea Internet & Security Agency; Byung-ik Kim, Korea Internet & Security Agency; Jun-hyung Park, Korea Internet & Security Agency

## Abstract

The real condition is that infringement incidents are automated and intelligent with the rapidly growing number of cyber threats. There are double-sided characteristics that attack tools used for infringement incidents share, but the characteristics of the tool are different for each attacker. As a result, demand on cyber threats information sharing increases, so that the whole world can respond to cyber threats instead of local respond to infringement incidents. This paper proposes a method of calculating a level of similarity by graph kernels among cyber threat information that is shared. And we also propose several techniques for facilitating development based on the proposed method. For example, in selecting targets for comparison, we have limited the input values to execute the delta function in original graph kernels. In order to make similarity more reliable, we adopt a pre-processing procedures by types of threat information shared. This system provides some insight into planning that cyber threats information can be grouped according to characteristics.

## Introduction

Nowadays, tools that threaten cyberspace are created automatically, and concerns about the AI-based infringement incident occurrence are on the rise. A global response system is required to respond to rapidly-increasing infringement incidents strategically, and the knowledge database information that has been regarded as an asset is shared in various ways and has multiple meaning. Anomaly detection based on internal network security events have frequently been studied together with the trend of big data analysis. However, more studies are needed on the integration and analysis of data featuring various types and meanings both on the inside and outside of the network and, in particular, in the cybersecurity area. In particular, it is imperative that different analytical methods differ depending on the context of data. These problems cause the low reliability of the analysis result. Issues arise about how to save and analyze cyber threats information with various types, sources, and mean

This motivates us to build a data warehouse based on the graph database was developed by refining and loading cyber threat information collected from 10 channels. For the reasons mentioned above, we thought that we should solve the problem in the direction of comparing groups, pre-processing information, and applying general graph analysis algorithms in cybersecurity. In this paper, the level of similarity between the vertices is calculated based on the vertex type and value of the graph loaded into the data warehouse, and the level of similarity among graphs is created using graph kernels. Lastly, we also share some techniques to implement result for contributing directly to threat information cluster that results using the modified graph kernels.

## Related research

Open-source intelligence (OSINT) refers to information that can be collected from an open-source but, in this paper, it is limited to the information that can be collected from computers only [1]. Table 1 shows several OSINT channels that provide information about cyber threats. Channels can be classified into IP, domain, and malicious code according to the type of data provided, or into the blacklist, distribution site, waypoint, and victim according to the meaning of the data provided.

**Table 1. OSINT channels that can be collected**

| Channel | Description |
|---|---|
| Exploit DB | Vulnerability, Exploit |
| NVD | Vulnerability |
| OTX | OSINT of AlienVault |
| MISP | Open source sharing platform |
| C-Share | Infringement incident information sharing system of the KISA |
| DNSBL | Blacklist information |
| Bambenek's C&C | OSINT of Bambeneck Consulting |
| Zone-H | Victim |
| VirusShare | Malware samples |
| VirusTotal | Malware analysis |

Open threat information is shared using different formats, provision methods, and service models. The framework that integrates those items is composed differently in each related study. Studies have not been performed to integrate those items because those items are hard to share due to a different saving structure, and a structured Threat Information Expression (STIX) is the first one proposed by MITRE [2]. STIX is a structured language for cyber threat intelligence, in the interest of brevity, and can be used to create cyber

threats storage structure with the same type. However, it can contain the information that is not used or cannot be collected, because it should be able to handle every category of cyber threats. Therefore, that type of information causes overload, and frameworks become different for each study because they are optimized for the data that can be collected. Likewise, the data warehouse was established in previous studies in such a way that it was optimized for the data that can be collected [3]. As a reference to the standard and previous work, data model for storage in graph form like Figure 1.

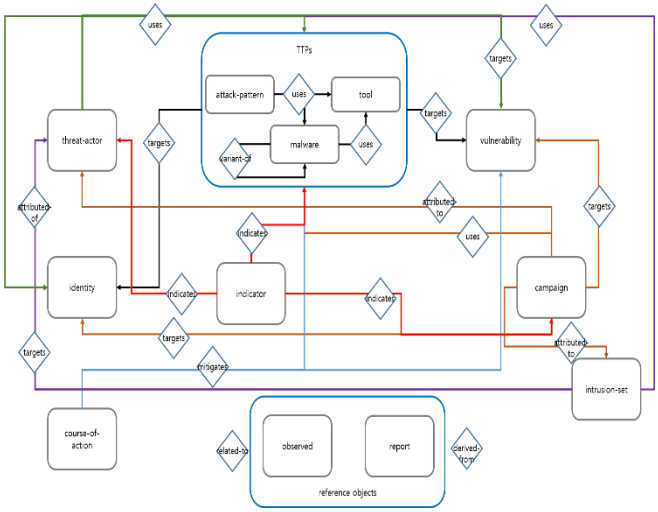


**Figure 1. Data model diagram in a graph database**

The algorithm used to calculate the level of similarity among cyber threat information graphs is same with that of normally used graphs. There are various methods of comparing simple graph connection patterns such as Jaccard Similarity, Cosine Similarity, and Graph Kernels. This paper uses the graph kernels algorithm, which is a comparison method that was previously used for cyber threat analysis and can reflect the connection pattern of a graph that has various types of vertices and meanings of values in the cybersecurity area [4]. Shortest-Path Graph Kernels (SPGK) can be defined as follows: When graph $G_1$ and $G_2$ are present, the graph type is modified as all vertices are connected by applying the Floyd-Warshall algorithm, and every edge of connecting the vertex is saved after applying the weighted value to the shortest distance. The SPGK function $K_{SP}$ for $G_{1SP}$ and $G_{2SP}$ calculated in this way can be defined as follows:

$$K_{SP}(G_1, G_2) = \sum_{e_1 \in E_1'} \sum_{e_1 \in E_2'} k_{walk}(e_1, e_2) \qquad (1)$$

Here, $G_{SP}$ is composed of $\{V, E'\}$ and $e_1$ connects $v_1$ and $w_1$, and $e_2$ connects $v_2$ and $w_2$. The delta($\delta$) function is defined as comparing type $v_1$ and $v_2$. The constant c is decided by the data type [5]. In the expression, $k_{type}(x, x')$ compares the vertex type. If the types are same, 1 is returned. Otherwise, 0 is returned. $k_{length}(x, x')$ is used to calculate the difference between original weight and weight after applying the Floyd-Warshall algorithm (called length).

# Proposed method

## Calculating similarity in threat information

The graph kernel algorithm is primarily designed to analyze information using the graph composition type. However, it needs to be modified before application, because the meaning of the value held by the vertex plays an essential role in calculating the level of similarity in cyber threat information graphs. Based on the previous research, we suggest a method in the environment where data is stored in a graph database as shown in Figure 2.

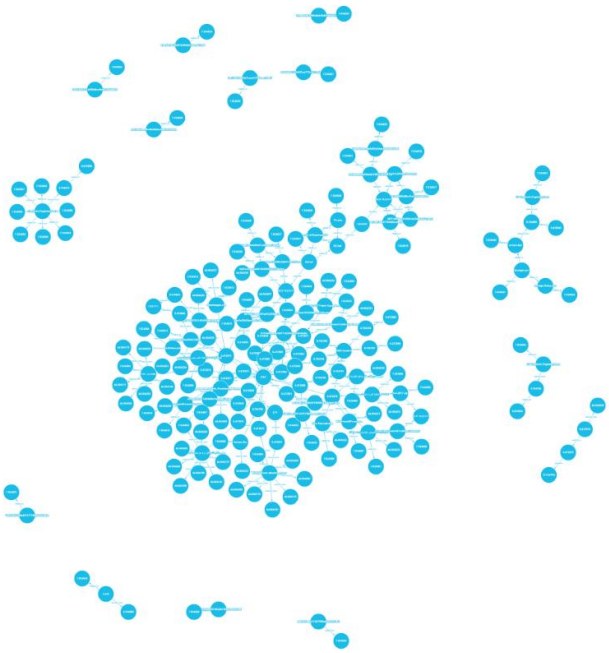

**Figure 2. Threat information stored in graph form**

Types of individual cyber threat information are classified into IP, domain, malicious code, and meta-data, and meta-data is defined as the "information that describes the information of remaining cyber threats." The values that represent each threat information type are different. Some types compare values only, and some information (e.g., hash value) is meaningless when compared. Therefore, we need a

method that pre-processes the level of similarity depending on the type of collected cyber threat information. For example, when we compare URLs that distribute vulnerable files, we can take the domain string, web path, and vulnerable file name into account. Levenshtein, the string comparison function provided by PostgreSQL that is adopted by the system with the proposing method, is applied to those items, and the maximum value of similarity for those items can be defined as the level of similarity of two threat information. It can be defined in such a way because the level of similarity among all information is not compared when deciding a similarity between two infringement incidents by profiling infringement incidents. Instead, if there is some similar information, it is determined that those infringement incidents are similar. As a rule, the same type of information should be compared, except when the threat information value is included in other threat information values like the domain↔URL.

**Table 2. Pre-processing methods by individual cyber threat information type**

| Data type | Pre-processing information |
|---|---|
| Domain | Sub-domain, domain string, SLD, TLD |
| Malicious code | Comparison with a focus on the file name, among collected malicious code information |
| Vulnerability | Comparison with a focus on the URL including a vulnerable file. It can be separated into the sub-domain, domain string, SLD, TLD, path, file name |
| IP | Checking whether the IP falls within the IP address range allocated to the ISP. If there is no address allocation information, determine based on whether the C-Class is same or not |
| Other meta-data | Executing a general string comparison algorithm without pre-processing |

Table 2 shows the comparative classification of similarity by type that is managed by the system. Table 2 shows the pre-processing process that should be handled respectively when compared only threat information is classified into five types based on the string. If the cyber threat information type belongs to a domain, the information is separated into four string types, and only the domain string is compared. Then, the result is calculated as the similarity value. The reason is that the domain string is the information that represents a domain, and other string (sub-domain, SLD/TLD) value generally appears. The final individual level of similarity can be calculated after defining and merging the methods of similarity calculation by cyber threat information type.

## Calculating comparison group's similarity

A similarity relationship can be generated among all vertices based on the level of threat information similarity. As the final objective is to calculate the level of similarity among information groups having similarity, we should be able to calculate the level of similarity among all vertices.
As existing algorithms perform analysis using graph types only, the algorithm needs to be modified so that the vertex value can be added for analysis. If the modified algorithm is called Modified Shortest-Path Graph Kernels (MSPGK), it can be defined as follows:

$$K_{MSP}(G_1, G_2) = \frac{1}{cM_1M_2} \sum_{e_1 \in E_1'} \sum_{e_1 \in E_2'} k_{walk}(e_1, e_2) \times S(v_1, v_2) \times S(w_1, w_2) \quad (2)$$

$M_1$ in the above definition refers to the number of edges included in $G_{1SP}$ where $1/cM_1\,M_2$ is a normalized term. When comparing types, existing algorithms use the result of 1 or 0. If there is similarity even though the vertex type is different, it cannot be used dichotomously. For example, if the vertex to compare is the URL and domain address, there is room for comparison even though their types are different. However, as this paper uses the original algorithm dichotomously, applying the standard when comparing different type vertices will be postponed to the next study. In addition, $S(v_1, v_2)$ in a similarity indicator of the vertex value can be defined as follows:

$$S(v_1, v_2) = 1 - \frac{Editdistance(v_1, v_2)}{\max(length(v_1), length(v_2))} \quad (3)$$

**Table 3. Example of re-classification by information's type**

| Type-1 | Type-2 | Value | Result preprocessed | Similarity target |
|---|---|---|---|---|
| IP | IP | 132.*.*.241 | 132.*.*.241(IP) | All |
| Domain | Domain | www.mor***fer.co.kr | www(sub-domain), mor***fer(domainstring), co(SLD), kr(TLD) | domainstring |
| Attack Type | Attack Type | RIG | RIG(attack type) | All |
| Malware | URL | hxxp://ko.ith***v.or.au/l/me/eal.jar | ko(sub-domain), ith***v(domainstring), or(SLD), kr(TLD), l/me/(path), eal.jar(filename) | domainstring, path, filename |
| | Filename | mal_testconfdko.exe | mal_testconfdko.exe(filename) | filename |
| Vulnerability | URL | hxxp://spoc***ac.com/img/front-page/11-001.gif | Spoc***ac(domainstring), com(TLD), img/front-page/(path), 11-001.gif(filename) | domainstring, path, filename |
| | CVE | CVE-2016-0723 | CVE-2016-0723(CVE) | None |
| File | Filename | 11-001-attach232 | 11-001-attach232(filename) | All |
| | Hash | 21a5ef699a0dc1f36…… | 21a5ef699a0dc1f36……(hash) | None |

This definition is also modified to limit the similarity indicator of the value to a value between 0 and 1. The result calculated using the algorithm described above can be used as a similarity indicator that considers the connection pattern value and similarity at the same time. However, as the volume of cyber threat information collected when calculating the level of similarity among groups increases numerously, the generated similarity relationship also increases geometrically. Therefore, the target of the prototype is to compare two threat information groups that are received from the collection channel after grouping. When the level of internal similarity for compared information is calculated, the following result can be obtained.

The proposed method is suitable for being used in a system which analyzes cyber threat intelligence. Typically in a live environment, collected threat intelligence has not weight between two infringement incidents. Because, most OSINT channels provide Indicators of Compromise (IoCs) that is composed of hash, domain, and IP. Therefore, we can decrease the priority of this value, weight. Even we also exclude the effectiveness of the weight, given that its value is one. If length$(v_1)$ and length$(v_2)$ are set one, max $(0, \mathrm{c} - |length(e_1) - length(e_2)|)$ is to be constant c. Moreover, delta($\delta$) function has a large amount of task in original SPGK. In order to avoid useless tasks, we can pick appropriate dataset consist of two vertices, one edge with the same type. If we choose suitable datasets, the system will not have to work for comparing data separated. In this respect, the abbreviated formulas are as follows:

$$K_{MSP}(G_1, G_2) = \frac{1}{\mathrm{cM_1M_2}} \sum_{\mathrm{e_1 \in E_1}} \sum_{\mathrm{e_1 \in E_2}} (\mathrm{S(v_1,v_2) \times S(w_1,w_2)}) \quad (4)$$

In the similarity calculated in the present paper, so-called "$K_{MSP}(G_1, G_2)$" serve as a factor to infer campaign from operations. With Supposition that there are two groups to perform comparison, the following is shows the procedures of extracting similarities.

1) Select two groups, you wish to compare
2) Extract individual information inside each group
3) Classify each information according to its type such as IP, domain, hash, etc.
4) Re-classify the information according to pattern
5) Apply the proposed method

Table 3 shows a detailed example of procedural three and four. The reason why the twice classification work is to increase reliability. The vulnerability is commonly shared by the data pattern of the URL and CVE. There is no point in trying to utilize the similarity calculated from CVE number. In others, malicious codes are shared in the form of hash value. However, the similarity between hash values does not mean anything. If we test Levenshtein algorithm to two hash values, 3ff60c100b67697163291690e0c2c2b7 and 617ba99be8a7d0771628344d209e9d8a, the result will be 0.15. Nobody would believe it was useful. Lastly, comparing particular text is better than doing full text. Likewise, when we analyzed this information shared from OSINT, we were able to organize Table 3.

An example of practical application is shown in Figure 3. At first, we choose information group 'Gaza Cybergang' and 'GRIZZLY STEPPE.' The developed system only start comparing information that is comparable inside the groups. The comparison of individual comparison operations is performed 996 times, and a similarity is calculated as 0.006. The result should be taken to mean group 'Gaza Cybergang' and 'GRIZZLY STEPPE' are not really similar.

0.0068113906

| 순번 | 타입 | g0001 | g0002 | 유사도 |
|---|---|---|---|---|
| 0 | domain_name | mo[redacted]fe | private.dire[redacted]ing.com | 0.14814814814814814 |
| 1 | domain_name | mo[redacted]fe | cd[redacted]h.com | 0.142857142857142905 |
| 2 | domain_name | mo[redacted]fe | editprod.w[redacted]lter.in.ua | 0.230769230769230727 |
| 3 | domain_name | mo[redacted]fe | insta.re[redacted]t.ru | 0.0666666666666666519 |
| 4 | domain_name | mo[redacted]fe | one[redacted]ee.com | 0.199999999999999956 |
| 5 | domain_name | mo[redacted]fe | rits[redacted]i.ru | 0.214285714285714302 |
| 6 | domain_name | mo[redacted]fe | littj[redacted]iap.ru | 0.0588235294117647189 |
| 7 | domain_name | mo[redacted]fe | wilc[redacted]be.com | 0.0714285714285713969 |
| 8 | domain_name | mo[redacted]fe | cde[redacted]rn.com | 0.142857142857142905 |
| 9 | domain_name | mo[redacted]fe | mymodu[redacted]lter.in.ua | 0.269230769230769273 |

**Figure 3. Example of similarity calculated proposed method**

# Conclusion and Future work

This paper presents a method of calculating similarities by modified graph kernels and pre-processing information shared in cyber threats through OSINT. More specifically, in order to calculate similarity in cyber threats, we proposed modified algorithms reflected values of vertices. It is vital to reflect values and optimize method because of the characteristics of cybersecurity area. This optimization plans on decreasing its workload to make the system even faster. That's all there is to be applying general graph analysis algorithm to cybersecurity.

We also showed how to make similarity more reliable through pre-processing procedures. Within this process, data cleansing and selecting are applied to bring the higher reliability of similarity.

Classification after calculating the level of similarity uses different algorithms depending on the data type. The representative algorithms used according to the data characteristics include k-means clustering and support vector machines for the quantified data, and modularity method and hierarchical clustering are used when the data is not structured, but quantified similarity exists among data. In addition, if the data is not structured and the similarity indicator cannot be quantified, machine learning is used. When comparing the efficiency of the produced outcome with the complexity of algorithm implementation, it is essential to use a proper algorithm according to the data type. As a result, we expect that it is more efficient to use the modularity method or hierarchical clustering according to the characteristics of the holding data. In addition, as it was mathematically proven that the modularity method and hierarchical clustering generate the same value, the modularity method that seems to be a graph methodology is going to be adopted [6].

As modularity optimization is primarily a quality function, approximation algorithms that optimize this value were thoroughly studied [7]. It was known that the Girvan-Newman method is generally the best-performing algorithm for small size graphs (less than 10K vertices), and the Louvain method that modifies the Greedy Algorithm is the fastest and best-performing algorithm for graphs larger than 10K [8, 9, 10]. The proposed measures will be promoted, and studies of extracting similar group will be conducted.

# Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT)(2017-0-00158, Development of Cyber Threat Intelligence(CTI) analysis and information sharing technology for national cyber incident response.)

# References

[1] Wikipedia, Open-source intelligence, https://en.wikipedia.org/wiki/Open-source_intelligence, retrieved October 17, 2017.

[2] OASIS Open, STIX, https://oasis-open.github.io/cti-documentation/stix/intro.

[3] Cho Heiysun, "Analysis of Cyber Threat Level based on Indicator of Compromise," 2017 Spring KIPS Conference on, pp.291-pp.294, 2017.

[4] Boukhtouta, Amine, Djedjiga Mouheb, Mourad Debbabi, Omar Alfandi, Farkhund Iqbal, and May El Barachi. "Graph-theoretic characterization of cyber-threat infrastructures," Digital Investigation 2015, 14, S3-S15.

[5] Borgwardt, Karsten M., and Hans-Peter Kriegel. "Shortest-path kernels on graphs," In Data Mining, Fifth IEEE International Conference on, pp. 8-pp. IEEE, 2005.

[6] Newman, M. E. J. "Equivalence between modularity optimization and maximum likelihood methods for community detection," Physical Review E 94, no. 5, 2016.

[7] Fortunato, Santo. "Community detection in graphs," Physics reports 486, 2010, no. 3, pp.75-pp.174.

[8] Newman, Mark EJ. "Fast algorithm for detecting community structure in networks." Physical review E 69, no. 6, 2004.

[9] Newman, Mark EJ. "Modularity and community structure in networks." Proceedings of the national academy of sciences 103, 2006, no. 23, pp.8577-pp.8582.

[10] Blondel, Vincent D., Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. "Fast unfolding of communities in large networks." Journal of statistical mechanics: theory and experiment, 2008, no. 10, P10008.

# Biographies

**SEULGI LEE** received the B.S. degree in Computer Science from the University of Chungnam, Korea, in 2013. Currently, He is a Researcher of Security Technology R&D Team 1 at Korea Internet & Security Agency. His research areas include cyber threat analysis, cyber attack related data correlation and machine learning algorithm. Seulgi Lee may be reached at sglee@kisa.or.kr

**NAKHYUN KIM** received the B.S. degree in Computer Science from the University of Seoil, Korea, in 2008, the M.S. degree in Network Security from the University of SoongSil, Korea, in 2011, respectively. Currently, He is a Deputy General Researcher of Security Technology R&D Team 1 at Korea Internet & Security Agency. His research areas include cyber threat analysis, cyber attack related data correlation, and Sensor Network Security. Nakhyun Kim may be reached at knh@kisa.or.kr

**HYEISUN CHO** received the B.S. degree in Computer Science from the University of Sejong, Korea, in 2013, the M.S. degree in Information Security from the University of SungKyunKwan, Korea, in 2017, respectively. Currently, She is a Researcher of Security Technology R&D Team 1 at Korea Internet & Security Agency. His research areas include cyber threat analysis, cyber attack related data correlation and cyber threat reputation analysis. Hye-sun Cho may be reached at hscho@kisa.or.kr

**BYUNG-IK KIM** received the B.S. degree in Computer Science from the University of Ajou, Korea, in 2010. Currently, He is a Deputy General Researcher of Security Technology R&D Team 1 at Korea Internet & Security Agency. His research areas include cyber threat analysis, cyber attack related data correlation, and Sensor. Byung-Ik Kim may be reached at kbi1983@kisa.or.kr

**JUN-HYUNG PARK** received the B.S. degree in Computer Science from the University of Ajou, Korea, in 1999, the M.S. degree in Multimedia from the University of Chonnam, Korea, in 2002, the PhD degree in Information Security from University of Chonnam, Korea, in 2004, respectively. Currently, He is a Manager of Security Technology R&D Team 1 at Korea Internet & Security Agency. His research areas include cyber threat analysis, malware analysis and mobile billing fraud detection. Junhyung Park may be reached at junpark@kisa.or.kr