Journal of Artificial Intelligence and Software Engineering
Vol. 5, No. 1, Maret 2025, pp. 205-214
E-ISSN: 2777-001X, DOI: 10.30811/jaise.v5i1.6518

Implementation of Cyber Threat Intelligence on Intrusion Detection System using STIX Framework

Yesta Medya Mahardhika[1*], Ferry Astika Saputra[2], Iwan Syarif[3], Prasetyo Wibowo[4], Misbahul Ardani[5]

[1] Teknik Informatika, Teknik Informatika dan Komputer, Politeknik Elektronika Negeri Surabya, Surabaya, 60111, Indonesia
[2,3,4,5] Teknik Informatika, Teknik Informatika dan Komputer, Politeknik Elektronika Negeri Surabya, Surabaya, 60111, Indonesia

Article Information

Accepted : 22 February 2025
Revised : 26 February 2025
Published : 20 March 2025

Keywords:

Cyber Threat Intelligence
Graph Vizualization
Intrusion Detection

ABSTRAK

Ancaman siber merupakan permasalahan kompleks dan beraneka ragam, seiring berkembangnya teknologi menjadi lebih banyak jenis ancaman yang terjadi. Pembuatan platform Cyber Threat Intelligence adalah solusi untuk menghadapi tantangan keamanan jaringan, dengan memanfaatkan Snort sebagai alat untuk mendeteksi ancaman pada lalu lintas jaringan dan STIX sebagai format serialisai serta standarisasi dari data *Cyber Threat Intelligence*. Hasil implementasi rancangan yang dibuat berupa platform *Cyber Threat Intelligence* berbasis Snort dengan menggunakan *Apache Spark* sebagai *processing engine*, MongoDB sebagai basis data, dan STIX sebagai format serialisasi serta standarisasi data. Pengujian dilakukan menggunakan 2 sumber data yaitu dataset CIC-IDS2017 dan real traffic. Graph memberikan informasi lalu lintas jaringanyang mencurigakan, negara asal penyerang, atribut dan pola serangan. Proses penerapan rancangan dengan mengubah data ke format STIX dan divisualisasikan ke bentuk graf. Hasil experiment menunjukkan bahwa konversi data *Snort* ke STIX membutuhkan waktu cukup lama jika jumlah data semakin besar, Real traffic membutuhkan waktu pemrosesan data selama 16 detik dan waktu konversi 3 menit, sedangkan PCAP membutuhkan waktu pemrosesan sebanyak 35 detik dan waktu konversi 13 menit.

ABSTRACT

Cyber threats are complex and diverse issues. Various types of threats emerge daily on the internet. In this research, we proposed a new Cyber Threat Intelligence platform to deal with the challenges above, using Snort as a tool for detecting anonymous network traffic and STIX as a serialization format and standardization of Cyber Threat Intelligence data. As a result, a Cyber Threat Intelligence based on Snort contains Apache Spark as the processing engine, MongoDB as the database, and STIX as the serialization format and data standardization. We test our platform by using two data sources, the CIC-IDS2017 dataset, and the real traffic. We successfully converted the snort alerts to STIX format and visualized them into graph. The graph shows indication of network traffic suspicious, the country of attacker come from, attribute information and attack pattern. The experiment shows that converting Snort data to STIX requires considerable time if the amount of data processed is getting bigger, Real Traffic needs 16 seconds of data preprocessing and 3 minutes of conversion time, while PCAP needs 35 seconds of preprocessing time and 13 minutes of conversion time.

Correspondence Author

Email: yesta@pens.ac.id

How to Cite IEEE :

Y. M. Mahardhika1, F. A. Saputra, I. Syarif, P. Wibowo, M. Ardani, "Implementation of Cyber Threat Intelligence on Intrusion Detection System using STIX Framework," *Journal of Artificial Intelligence and Software Engineering (J-AISE)*, vol. 5, no. 1, pp. 205-214, Maret 2025. doi:10.30811/jaise.v5i1.6518

1. INTRODUCTION

STIX (Structured Threat Information Expression) is the standard cyber threats are complex and diverse issues. According to recent Symantec threat reports, various types of threats emerge daily, such as malware, DDoS, phishing, spamming, keylogging, click fraud, and theft of personal information [1]. Human reliance on internet technology is susceptible to vulnerabilities, and cybercriminals can exploit numerous security gaps. The growing threat landscape, including cloud computing and distributed systems, also introduces more complicated attack scenarios [2]. Attackers disrupt systems by executing Distributed Denial of Service (DDoS) attacks against their targets. Considering the perspectives of attacker and system perspectives, there is a need for effective defense strategies to maintain a balance of information that reveals the attacker's true intentions, allowing for intelligent defensive decisions [3]. An Intrusion Detection System is commonly used to identify intrusions and vulnerabilities within a network. Snort is an example of a signature-based Intrusion Detection System. The effectiveness of Snort's detection relies on the rules set by the network administrator; however, the visualization provided by Snort lacks sufficient implementation and primarily focuses on presenting statistical data captured by the system. Consequently, understanding the attacker's behavior, capabilities, and intentions is challenging due to these limitations [4].

STIX (Structured Threat Information Expression) is the standard format utilized by cybersecurity analysts. STIX facilitates the modeling, visualization, and sharing of Cyber Threat Intelligence (CTI) models, offering valuable insights and contextual intelligence to construct more comprehensive narratives of cyber-attacks and to grasp ongoing threats better [5], [6]. STIX allows organizations to share one CTI each other in a consistent and machine-readable manner, enabling the security community to understand better he computer-based attacks they are most likely to see and to anticipate and/or respond to those attacks faster and more effectively. This data is an essential tool for security professionals to act swiftly in mitigating cyber-attacks or preventing further threats. However, despite its extensive documentation, STIX presents significant challenges due to its inherent complexity, as creating or updating models requires managing numerous parameters [7], [8].

Several methods have been developed to build standardized STIX using various threat modeling approaches. Sadique et al. [7] proposed a novel, privacy-preserving mechanism for automatically representing raw cyber threat data in STIX format.The results indicate that this method is suitable for large-scale deployments. Ahmad et al. [9] applied organizational learning theory to develop a conceptual framework, improving Information Security Management and Incident Response functions. The strong integration of ISM and IR functions enhances organizational awareness of security risks, compilation of threat intelligence, elimination of security flaws, evaluation of defensive strategies, and bolstered security response. Czekster et al. [10] created a smart grid cyberaCTIve tool to enhance STIX-based modeling tasks, capable of graphing the microgrid and more broadly. Tatam et al.[11] conducted a review of APT-style attacks in threat modeling by combining multiple modeling approaches in a journal, focusing on identifying limitations, strengths, and potential enhancements to improve TM performance and efficiency in the modeling process. Lallie et al. [12] reviewed a different scope using attack graphs and attack tree visuals, while Ainsle et al. [13] examined how organizations can implement cyber threat intelligence through operational scheduling practices, different strategic risks, and levels of exploration. Pratama et al. [14] proposed a modelling attack by behavior of intrusion data. To measure this method, they used real-time and non-real-time event testing.

According to this related work and problem, we applied a new method and modelling cyber threat intelligence framework for analyzing cyber threats using STIX (Structured Threat Information Expression) and Snort IDS as dataset resources. This system can convert alerts that Snort generates into STIX form and visualize

it to form a graph that can display the relationship between cybercriminals, targets, and the attack methods used.

2. METHODS

This section explains how the system is built and works coherently in each part, starting from the stage of getting the data until visualizing the data. Figure 1 shows the system design of the proposed cyber threat intelligence system. The system is built from Snort IDS as the data source, Apache Spark as a processing engine, MongoDB as the database used, and STIX as the serialization format and data standardization. The system can provide data in STIX format and visualize it into a graph.

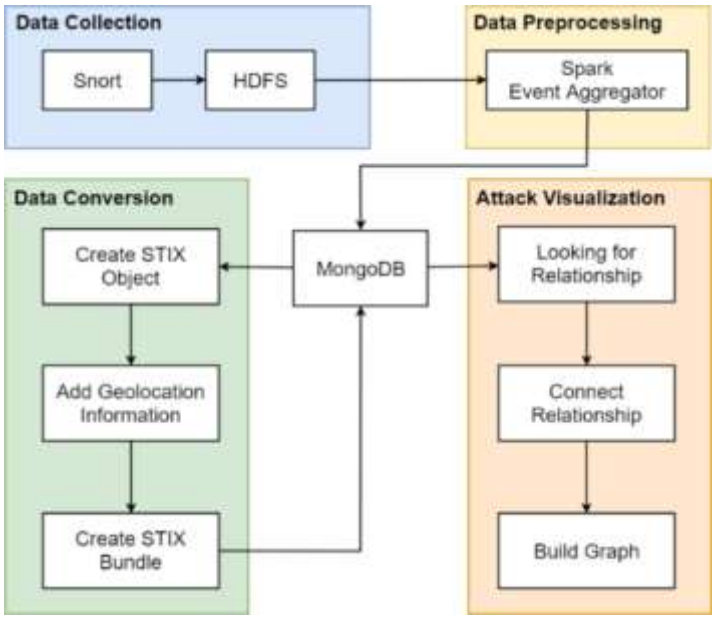

Figure 1. System Design

2.1. Data Collection

On protected subnets, we install Snort as an IDS sensor. In the data collection stage, we collect alerts from the set of IDS. Our IDS Snort sensor uses the subscribed community rule to detect suspicious packages. Our design makes it possible to install IDS Snort Sensors at different subnets with the Defense Center as the processing unit. The diagram of the sensor is shown in Figure 2. We use the UNIX socket logging to capture alerts from Snort IDS. Then, we use the Python Unsock library to deserialize and unpack UNIX sockets. The Python Unsock Parser then publishes log in the JSON format to the MQTT Server.

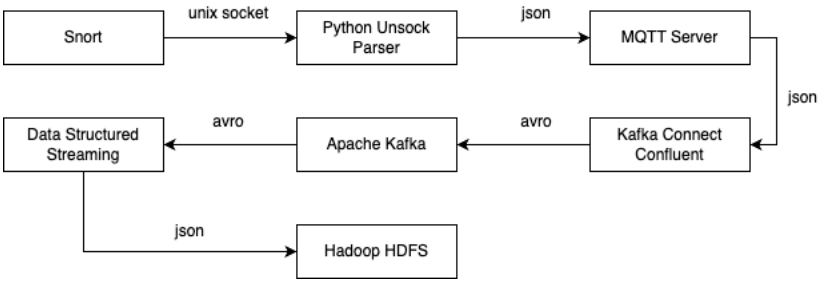

Figure 2. Data Type Flow

The study uses Apache Kafka as a message broker that bridges between the Snort sensor and the Spark engine. The MQTT package does not solely accept the means by Kafka brokers. The MQTT server will receive the MQTT package sent from the sensor. Then, by using the Kafka-connect feature from Confluent. Kafka brokers will not actively publish topics to other destinations but will only serve nodes that subscribe to specific topics on brokers. In this paper, Apache Spark that subscribes to brokers, using spark structured streaming data

will be collected and transformed / query/mapping / reduce or other processes in a stream, and the results are stored in HDFS in JSON format.

2.2. Data Preprocessing

Preprocessing data is the stage where the collected data is filtered and aggregated using Apache Spark batch processing. Apache Spark that subscribes to brokers, using spark structured streaming data will be collected and transformed / query/mapping / reduce or other processes in a stream, and the results are stored in HDFS in JSON format. The flow of data preprocessing is stored data from HDFS converted into data frames and mapped according to the defined schema. Then, we perform a query to select fields from data captured by IDS. We called **Event (E)** as the set of data containing timestamp, source IP, destination IP, destination port, protocol, and alert messages, as shown in equation 1.

$$CCE = \{\text{ts}, \text{src ip}, \text{dst ip}, \text{dst port}, \text{protocol}, \text{alert\_msg}\} \quad (1)$$

Description of ts is timestamp of data, src-ip is source ip, dst_ip is destination ip, dst_port is destination port, protocol is TCP/UDP protocol, and alert_msg is Alert message from snort.

We define $\boldsymbol{E_T}$ as a set of *E* that occurs in time window ***T.***

$$ET = \{ E1, E2, E3, E4, \ldots, Ei \} \quad (2)$$

where:

$$Ei = \{ tfirst, tlast, src\ ip, dst\ ip, dst\ port, protocol, alert\ message, number\ observed \} \quad (3)$$

In equation 2. ET is a collection of events aggregated at a certain time unit. T is a time window, which is a fixed time interval when data is processed or aggregated. After the aggregation process is done, there are additional fields such as first observed, last observed, and the number observed mentioned in equation 3.

2.3. Data Conversion

Data Conversion is the process of aggregated previous event data stored in MongoDB from Snort converted to STIX.

Pseudocode 1: Generate STIX Objects.

```
1:  procedure EventToSTIX(events)
2:  initialize bundles
3:  for event ← events do
4:  observedData ← createObservedData(event)
5:  indicator ← createIndicator(event)
6:  identity ← createIdentity(event)
7:  identityTarget ← createIdentityTarget(event)
8:  threatActor ← createThreatActor(event)
9:  attackPattern ← createAttackPattern(event)
10: relationships ← createRelationships(event)
11: bundle ← createBundle([observedData, indicator,
    identity, identityTarget, threatActor, attackPattern,
    relationships])
12: bundles.insert(bundle)
13: return bundles
```

Figure 3. Generate STIX object

Figure 3 explains that to make STIX objects, a collection of events is needed to be converted to the observed data object, indicator, identity, identity target, threat actor, attack pattern, and relationships. Then each object will be bundled into one bundle, which will be stored to MongoDB. STIX objects like Identity and Threat Actor need Geo-IP information. Figure 4 explains how to obtain geographic information on the IP address of an event.

Pseudocode 2: Get Geo–IP Information.

```
1:  procedure LookupIP(ip)
2:    reader ← geoipDatabase(ip)
3:    try
4:      country ← reader.getCountry()
5:      city ← reader.getCity()
6:    catch Address Not Found
7:      if is private IP address then
8:        country ← "private"
9:        city ← "private"
10:     else
11:       country ← "undefined"
12:       city ← "undefined"
13:   end try
14:   return [country, city]
```

Figure 4. Pseudocode of Get Geo IP Information

Figure 4. explains how to create a STIX object. Observing data is an object that has the purpose of telling information found on a system or network, such as an IP address.

$$Obs = \{ \text{first observed}, \text{last observed}, \text{src ip}, \text{dst ip}, \text{dst port}, \text{protocol}, \text{number observed} \} \quad (4)$$

Description of first observed is he first time the event is observed, last observed is the last time the event is observed, and the number observed is the same number of events.

Equation 4 explains how the structure of the Observed Data object. At Observed Data created using the src ip, etc ip, etc. port, protocol, the observed number obtained from the event. The indicator is an object that contains a pattern that can be used to detect suspicious or malicious cyber activity.

$$Ind = \{ \text{pattern} \} \quad (5)$$
$$pattern = \{ \text{src type}, \text{dst type}, \text{src value}, \text{dst value}, \text{qualifier} \} \quad (6)$$

Description of pattern is a suspicious pattern captured by IDS, src type is type of pattern source, dst type is type of destination pattern, src value is value of the source, dst value is value of the destination type, for example ip address, qualifier is limitation of patterns obtained, examples of attacks carried out 3 times, or in a span of 1 minute. Equations 5 and 6 explains how the structure of the object Indicator. The Indicator is created using the src ip attribute, etc. ip, the observed number obtained from the event. An attack pattern is a type of Tactics, Techniques, and Procedures (TTPs) that describe the ways in which actors attempt to attack a target. The Attack Pattern is created using the alert message attribute obtained from the event.

Identity is an object that describes individuals, organizations, or groups, as well as classes of individuals, organizations, or groups. The Identity is created using the src ip or etc ip attribute obtained from the event.Threat actor is individuals, groups, or organizations that are believed to do evil intentions (actors). The Threat Actor is created using the src ip attribute obtained from the event. Relationship is used to connect two STIX objects and to describe how they are related to each other. The Relationship is made using STIX objects that have been created before. Bundle is a collection of STIX objects bundled together in one container. Bundles are made using STIX objects that have been created before.

2.4. Attack Visualization

Attack Visualization is a stage where a collection of STIX objects is visualized into a graph form so that it is interactive and easy to understand. Graphs consist of nodes and links; nodes/vertices/vertices are usually represented by the form of links with labels, and a link/edge is represented by a line or arrow that extends from one node to another node.

Pseudocode 3: Visualize Attack.

```
1:  procedure CreateGraph(bundle)
2:    initialize nodes
3:    initialize links
4:    for object ← bundle do
5:      if object type is relationship
6:        link.source ← object.source_ref
7:        link.destionation ← object.destination_ref
8:        link.label ← object.relationship_type
9:        links.insert(link)
10:     else
11:       nodes.insert(object)
12:   DrawGraph(nodes, links)
```

Figure 5. Pseudocode of Attack Visualization

Figure 5. Explain how to visualize STIX objects into graphs. First, initialize the nodes, and links will be used to draw each STIX object. Then, on each object that is in the bundle, the relationship of each node is searched, such as the source, destination, and label of the relationship of each node. After all data is collected, the final step is to draw the graph.

3. EXPERIMENT

The experiment in this study is divided into two, using real traffic from Snort IDS running on VPN Cloud and simulation using the PCAP dataset.

3.1. Real Traffic

The test is done with real traffic, which aims to test whether the system can collect, process, and get genuine attacks that occur on the internet. The topology used is shown in Figure 6.

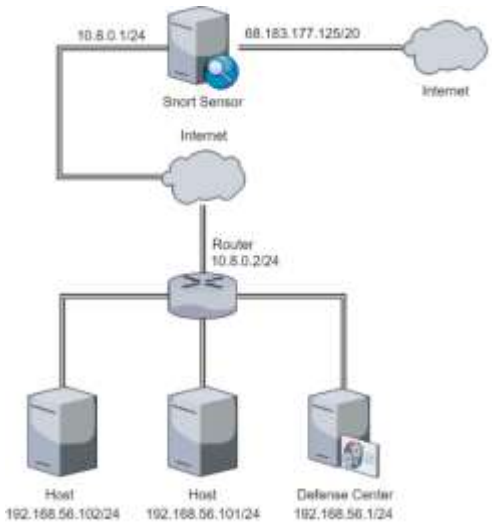

Figure 6. Real Traffic Topology

The following are the environments and devices used in the test:

- There is a Snort IDS sensor that runs on VPS Cloud with IP addresses 68.183.177.125/20 and 10.8.0.1/24.
- The defense center, host, and router are on the same network.

- A defense center is a node that functions as a data collection, data preprocessing, data conversion, and attack visualization as described in the system design. The Defense Center has an IP address of 192.168.56.1/24.

3.2. Simulation using PCAP

Simulation using CIC-IDS 2017 / PCAP dataset that was reimplemented by Snort IDS, the network topology in Figure 7 is as described by the related research.

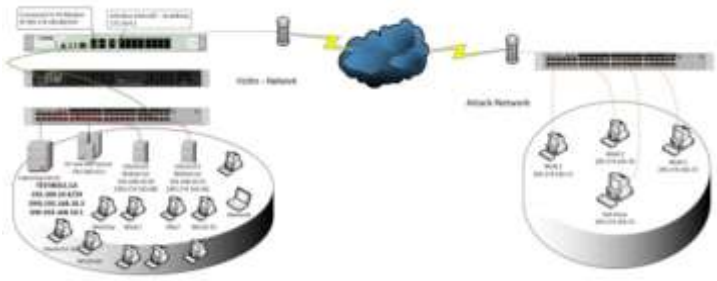

Figure 7. Topology CIC-IDS 2017

Figure 7 is the topology of the dataset made by researchers [7]. In this paper, use the Wednesday dataset, which contains DoS and Heartbleed Attacks slow loris, Slowhttptest, Hulk, and Golden Eye attacks.

3.3. Data Analysis

The testing scenario to analyze this study is using Real Traffic Alert and shows five alert messages and the number of each alert.

Table 1. Real Traffic Alert

No	*Alert Message*	Count
1	Consecutive TCP small segments exceeding threshold	363.412
2	Reset outside window	10.709
3	(spp_ssh) Protocol mismatch	896
4	Bad segment, adjusted size <= 0	240
5	(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE	206

The total data obtained is 375,888. In Table 1, there are five samples of data captured by the Snort IDS sensor. The biggest attack was "consecutive TCP small segments exceeding a threshold," with 363,412 attacks. The smallest alert message is HTTP inspect, with a total number of 206.Other testing scenario use PCAP Traffic Alert and the total data obtained is 5,703,781 from five highest alert message.

Table 2. PCAP Traffic Alert

No	Alert Message	Count
1	Reset outside window	1.703.700
2	(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE	1.472.494
3	Consecutive TCP small segments exceeding threshold	69.067
4	(hhtp_inspect) LONG HEADER	24.211
5	Bad segment, adjusted size <= 0	8.340

In table 2, there are 5 samples of data captured by the Snort IDS sensor. The biggest attack was "Reset outside window," with 1,703,700 attacks. The smallest attack was the Bad segment, with 8.340.

3.4. Data Preprocessing

Based on the results of the system test using real traffic data and PCAP. From the data that was successfully obtained, the data will be processed from preprocessing data to become STIX objects obtained. As shown in

Table 3, the time needed to preprocess the data is relatively short, and the difference in time needed is not too long when processing large amounts of data. This result is obtained because it uses Apache Spark as a processing engine.

Table 3. Data Preprocessing Duration

Type	Preprocessing Time	Conversion Time
Real Traffic	16 seconds	3 minutes
PCAP	35 seconds	13 minutes

The amount of data processed can be seen in Table 4. It shows that the alerts made by Snort are very large; for PCAP data, there are almost 3 million alerts made. The Processing time of real traffic is 16 seconds with 3 minutes conversion time, while PCAP's processing time is 35 seconds and 13 minutes conversion time.

Table 4. Amount of Data Before and After

Type	Snort Alert	STIX objects
Real Traffic	375.888	6.232
PCAP	3.303.697	27.351

Converting data from preprocessing results to STIX objects takes a long time if there is a lot of data to be processed. It is seen that fewer STIX Objects are successfully created from this amount of data. Table 4. shows that Real Traffic has 375.888 of Snort Alerts and 6.232 STIX objects, while PCAP has 3.303.697 Snort Alerts and 27.351 STIX objects.

3.5. Traffic Analysis for Real Traffic

The test resulted in the top 3 ranking in several categories, such as IP Address Activity, Attacker's Country Activity, Target IP Address Activity, and Target Country Activity.

Table 4. Top 3 Attacker's IP Address Activity

Number	IP Address	Count
1	202.9.85.34	212.473
2	36.82.97.33	90.076
3	202.9.85.38	8.001

Table 5. Top 3 Attacker's Country Activity

Number	Country	Count
1	Indonesia	352.146
2	China	33.686
3	United States	10.950

Table 5 shows the top 3 attacker IP activity, the most attacking IP is 202.9.85.34 with a total number of 212.473, then followed by IP 36.82.97.33 with 90.076, and the last is IP 202.9.85.38 with 8.001. Table 6 shows the real traffic, the most attacking country is Indonesia, with a total number of 352.146, then followed by China with 33.686, and the smallest is the United States, with 10.950.

Table 6. Top 3 Target IP Address Activity

Number	IP Address	Count
1	68.183.177.125	36.8833
2	218.92.0.157	1.646
3	218.92.0.169	1.374

Table 7 shows the highest target is IP 68.183.177.125 with 36.883, then followed by IP 218.92.0.157 with 1.646, and the last is IP 218.92.0.157 with 1.374. Table 8 shows the target statistics of the attacks, with the most targeting the United States, with a total number of attacks of 371.504, followed by China, with 3.853, and Singapore, with 186.

Table 7. Top 3 Target's Country Activity

Number	Country	Count
1	United States	371.504
2	China	3.853
3	Singapore	186

3.6. Graph Testing

In this test, the STIX object was successfully visualized into a graph. The graph in Figure 6, shows that the Actor from the United States uses the "Reset outside window" attack pattern to attack targets originating from the United States (because the sensor has a United States public IP address).

Figure 6. Topology CIC-IDS 2017

3.7. DDOS Graph Vizualisation

In this test, the DDoS attacks were successfully visualized in a graph. The graph can be seen in Figure 7, Actors come from more than one country, such as the United States, China, Germany, India, Vietnam, Singapore, Hong Kong, Korea, and Venezuela Korea, and use the "Reset outside window" attack pattern to attack targets originating from the United States (because the sensor has a United States public IP address).

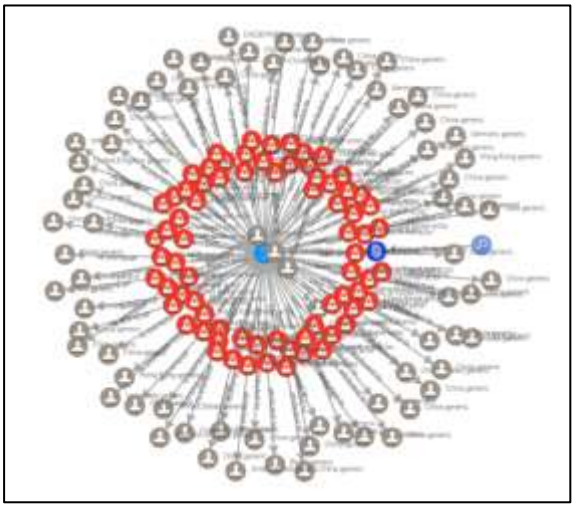

Figure 7. DDoS Graph Visualization

4. CONCLUSION

In this paper, we present a new system of cyber threat intelligence using Snort as a data source, Apache Spark as a processing engine, MongoDB as a database, and STIX as a format for serialization and standardization of data. The Analyse Scenario shows that the Real Traffic Alert is fewer, with a total of number 375,888 than the PCAP Traffic Alert, with 1,703,700. Both Real Traffic Alert and PCAP Alert have different total ranks of Alert Messages, the first rank of Real Traffic Alert is Consecutive TCP small, while PCAP traffic is Reset Outside Window. The time to change from data to STIX format takes a long time if the data processed is getting bigger, it means that the amount of data is proportional to the Pre-processing and Conversion Time. Real Traffic needs 16 seconds of data preprocessing and 3 minutes of conversion time, while PCAP needs 35 seconds of preprocessing time and 13 minutes of conversion time. The test resulted in the top 3 ranking of IP Address Activity, showing the most attacking IP is 202.9.85.34, then followed by IP 36.82.97.33, and the last is IP 202.9.85.38. The top 3 Attacker's Country Activity is Indonesia, then followed by China and the United States. The Top 3 Target IP Address is 68.183.177.125, followed by 218.92.0.157 and 218.92.0.169. The Top 3 Target Countries are the United States with a total of number 371.504, followed by China with 3.853, and

Singapore with 186. The system shows the data was successfully standardized to STIX format and can be visualized into a graph to facilitate analysis of the attacks. The Graph testing show an indication of network traffic suspicious, the country of attacker come from, attribute information and attack pattern.

REFERENCE

[1] Y. Medya Mahardhika, A. Sudarsono, and A. Barakbah, "Botnet Detection Using On-line Clustering with Pursuit Reinforcement Competitive Learning (PRCL)," *EMITTER International Journal of Engineering Technology*, vol. 6, no. 1, 2018.

[2] S. F. Astika, M. Jauhari, N. Isbatuzzin, M. Salman, and K. Ramli, "Building a Dynamic Scalable Parallel Cloud-Based Snort Nids Using Containers and Big Data," *Journal of Southwest Jiaotong University*, vol. 56, no. 5, pp. 317–326, Oct. 2021, doi: 10.35741/issn.0258-2724.56.5.27.

[3] S. E. Jasper, "U.S. Cyber Threat Intelligence Sharing Frameworks," *International Journal of Intelligence and CounterIntelligence*, vol. 30, no. 1, pp. 53–65, Jan. 2017, doi: 10.1080/08850607.2016.1230701.

[4] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *ICISSP 2018 - Proceedings of the 4th International Conference on Information Systems Security and Privacy*, SciTePress, 2018, pp. 108–116. doi: 10.5220/0006639801080116.

[5] S. Barnum, "Standardizing Cyber Threat Intelligence Information with the Structured Threat Information eXpression (STIX™)," 2012. [Online]. Available: http://www.lockheedmartin.com/content/dam/lockheed/data/corporate/documents/LM-White-Paper-Intel-Driven-Defense.pd

[6] Jun. Guo, *Proceedings of 2016 5th IEEE International Conference on Network Infrastructure and Digital Content : IEEE IC-NIDC 2016 : September 23-25, 2016, Beijing, China*. IEEE, 2016.

[7] F. Sadique, S. Cheung, I. Vakilinia, S. Badsha, and S. Sengupta, "Automated Structured Threat Information Expression (STIX) Document Generation with Privacy Preservation," in *2018 9th IEEE Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2018*, Institute of Electrical and Electronics Engineers Inc., Nov. 2018, pp. 847–853. doi: 10.1109/UEMCON.2018.8796822.

[8] R. Kwon, T. Ashley, J. Castleberry, P. McKenzie, and S. N. Gupta Gourisetti, "Cyber threat dictionary using MITRE ATTCK matrix and NIST cybersecurity framework mapping," in *2020 Resilience Week, RWS 2020*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020, pp. 106–112. doi: 10.1109/RWS50334.2020.9241271.

[9] A. Ahmad, K. C. Desouza, S. B. Maynard, H. Naseer, and R. L. Baskerville, "How integration of cyber security management and incident response enables organizational learning," *J Assoc Inf Sci Technol*, vol. 71, no. 8, pp. 939–953, Aug. 2020, doi: 10.1002/asi.24311.

[10] R. M. Czekster, R. Metere, and C. Morisset, "cyberaCTIve: a STIX-based Tool for Cyber Threat Intelligence in Complex Models," Apr. 2022, [Online]. Available: http://arxiv.org/abs/2204.03676

[11] M. Tatam, B. Shanmugam, S. Azam, and K. Kannoorpatti, "A review of threat modelling approaches for APT-style attacks," Jan. 01, 2021, *Elsevier Ltd*. doi: 10.1016/j.heliyon.2021.e05969.

[12] H. S. Lallie, K. Debattista, and J. Bal, "A review of attack graph and attack tree visual syntax in cyber security," Feb. 01, 2020, *Elsevier Ireland Ltd*. doi: 10.1016/j.cosrev.2019.100219.

[13] S. Ainslie, D. Thompson, S. Maynard, and A. Ahmad, "Cyber-threat intelligence for security decision-making: A review and research agenda for practice," *Comput Secur*, vol. 132, Sep. 2023, doi: 10.1016/j.cose.2023.103352.

[14] R. Y. Pratama, "The Development of Cyber Threat Intelligence Visualization on The Mata Elang Platform using STIX Framework," Surabya, 2022.