A
Project Report
on

Big Data Analytics

REAL-TIME NETWORK TRAFFIC ANALYSIS FOR THREAT DETECTION

Submitted by-

Name of the Student	Enrolment Number
Anant Kaul	*1800201C202*

Computer Science Core

Under the guidance of

Dr. Yogesh Gupta
Associate Professor

Department of Computer Science and Engineering
SCHOOL OF ENGINEERING AND TECHNOLOGY
BML MUNJAL UNIVERSITY GURGAON-122413, INDIA
MAY, 2021

Acknowledgement

In successfully completing this project, many people have helped us. We would like to thank all those who are related to this project.

Primarily, we would thank God for being able to complete this project with success. Then, we will thank our Vice Chancellor, Prof. Manoj K. Arora and Dean SOET, Prof. Maneek Kumar for providing us the golden opportunity to work on this project which helped us research and grow in knowledge. Also, we'd like to express special gratitude and thanks to our faculty mentor, Dr. Yogesh Gupta from BML Munjal University under whose guidance we learned a lot about this project and were able to implement our concepts in a logical way. His suggestions and directions have helped in making us capable and confident for completing this project magnificently.

Finally, we would like to thank our parents and friends who have helped us with their valuable suggestions and guidance and have been very helpful in various stages of project completion.

Thanking You
Anant Kaul

Table of Contents

Abstract

Wi-Fi is now ubiquitous in most populated areas, and the way the devices communicate leaves a lot of 'digital exhaust'. Usually, a computer will have a Wi-Fi device that's configured to connect to a given network, but often these devices can be configured instead to pick up the background Wi-Fi chatter of surrounding devices. There can always be good reasons as well as bad ones for the same, but the matter is all about the intensions. So, now imagine how many packets are flowing in a network and how harmful or useful they can be. Keeping the bad part aside, we can use this for ethical purpose, like in this report. Now, the analysis can be done by examining the flow of packets (which can add some reasonable meaning to it) that are being sent/received over the network (heavily). This report basically revolves around the analysis and hence, can be detected on a real-time basis.

Keywords: Wireshark CLI (tshark), Zookeeper, Kafka, Logstash, Elasticsearch, Kibana.

Motivation

Due to the colossally increasing flow of data packets (as network traffic analysis in [1]), it has become extremely important to dissect and examine a certain protocol inside the network traffic captured. So, the major reasons for analyzing the network traffic at real-time are:

(1) By collecting this type of traffic, it can be possible to track and analyses behavior in ways that we may or may not feel comfortable with.

(2) Improving public transport as TFL (Transport for London) as in [2] is already doing by collecting the Wi-Fi connection data to get a far better understanding of how customers move through stations. Moreover, the traffic captured is not used to identify any specific individuals or monitor anyone's personal browsing history. Everything done is just for analyses purpose and obviously a good cause.

(3) Not likely but tracking customers in-store as in [3] could be used to provide the customers with the best offers and hence, analysis is to be done in real-time, which we normally consider near-real-time.

1. INTRODUCTION

The messages sent or received in any network generally flow at a rate of 100k per hour (approximately 30 per second). Somewhat that can't be considered as "Big Data", but it isn't insignificant either. Likewise, it becomes important to monitor computer networks of majorly every network operator. The captured valuable information about utilization and status of a network infrastructure can go in wrong hands during the monitoring stage. But keeping network security in mind, safety of users and their respective data is the key to a safe and ethical environment. Due to the sheer volume of data that is transferred through modern network infrastructures, monitoring systems mainly collect traffic information in smaller flow logs. These traditionally consist of network addresses, ports, timestamps, channels, radio signals and volume information. This data can be used in accounting or in statistical analysis to improve any situational awareness or to detect anomalies.

Every such data packet, which contains such information can be extremely harmful (if gone in black hands). So, it becomes crucial to analyze such packets on a regular basis to trace out the real black hat. With rising threats, cybersecurity becomes very important in today's era. But when we talk about the amount of data that we get; it is also important to take the term "Big Data" into consideration. Moreover, streaming it in near-real-time becomes the tip of the iceberg. Cybersecurity itself has no meaning if not done in real-time. But for this research, let's keep "Cybersecurity" as the tip of the iceberg and emphasize more towards streaming real-time data and further analyze/visualize streamed (captured) data packets using the "Big Data Technologies".

This research follows certain steps to detect, analyze and then finally visualize the pattern of the network protocols or the data packets flowing. To get a clear view of the steps, this research will take you through the methodology, where all the terminologies and its respective usage will become clear with appropriate examples, pertaining to step-by-step demonstration (using appropriate figures/tables). Process flow diagram with appropriate notations (tools) are also used to make the process more clear and easily understandable.

The paper is organized as follows. Section 2 states the problems that motivate or require a solution provided by this proposed system. Section 3 lists the related work which helps to gain a deeper understanding of the problem in picture. Section 4 elaborates on the

methodology behind building the system, with analysis on the various aspects of the projects and how they integrate to give one state of the art solution for the problem. Section 5 discusses the proposed system and derives its corresponding results. Section 6 of the paper draws the conclusion and talks about the future work that will go into perfecting as well as updating the system with the constant feedback checking.

2. PROBLEM STATEMENT

2.1 Problem Motivation & Respective Solution

With the rising number of cyber threats and cybercrimes, it has become crucial to get an eye upon its numbers. Many other reasons may involve the analysis done with streams of Wi-Fi packet capture (pcap) data. As already mentioned earlier, these packets carry very critical information of the host. Such information can become vulnerable and can be captured to analyze the user behavior and predict the hacker (passive or active) in some or the other way. This research basically analyzes the user behavior and detects the black guy, who is trying to trace, harm or deny the victim from using the services. This denial of service is also referred to as DoS (Denial-of-Service) or DDoS (Distributed Denial-of-Service) attack. Further, many types of man-in-the-middle (MITM) attacks can be predicted with such an enhanced solution as provided in this report.

2.2 Solving Specific Problem

Solving a specific problem can never be the right way for problem solvers. Rather, making the solution that can be widely used must be the right approach. Keeping that thought, this research provides a vast knowledge base to the analysts for performing different types of analyses over the captured dataset and then finally predict (as in [4]) the untrustworthy users as per their convenience and analytical skills in real-time (precisely near-real-time).

2.3 Detailed Explanation

The main problem begins when the black hats try to hook or trace out a victim. Keeping the cybersecurity and all the protocols in mind, this research helps all types of the analysts to gather vast information accurately belonging to those black guys. This is easily done with the help of event streaming and further using the right

pipelining framework to analyze and predict different cyber-attacks, black hats, and middleware (man-in-the-middle attackers). Also, *Fig.1*, as given in section 4, describes the whole scenario from capturing the flowing data packets (in any network within range) to finally giving it a shot for perfect visualizations and future predictions. Performing such steps can have a good as well as a bad intention. Keeping all the evils aside, this research tends to take it in a direction where every single analysis is done to gain information only for a constructive resolution, that too only for educational and research purpose (like in [5]).

3. LITERATURE REVIEW

3.1 Existing State-of-the-Art

Various methods for the detection of the black hats are present like "Wireshark API." But taking it a step forward towards analyzation and visualization, makes it much more powerful and easily noticeable for analysts to perform different predictions and hence, trace out the black guy in real-time. A similar approach is discussed in [6], where the author intends to present a hybrid scenario for emulating networks on one or more PCs with the CORE demonstrations keeping the wired as well as wireless settings. Their system yielded good results but at the same time the process becomes very technical and require more of network based knowledge, whereas it should be simple and should be available and understandable by the data scientists as well to apply their knowledgebase and make it more powerful than ever.

In [7], the author states an innovative approach for real-time network traffic classification. The developed model adopts a new approach for the relaxation of the hypothesis of independence between the attributes of the Naive Bayes algorithm. As a result, the proposed system becomes a promising alternative to be applied in real-time scenarios. But at the same time, the researcher [7] do requires skills of finding the hypothesis of the dataset, whereas in this project, every analyst is welcomed and is provided with knowledge about the system in a much more easy and simple way that too from scratch. Similarly, [8] lists down the real-time diagnoses of network anomaly based on statistical traffic analysis by experimentally showing the developed mechanism to alert the DDoS attack schemes within short

respond time. The drawbacks as listed for [7] can be considered for [8] as well as the knowledge of statistical measurements gets included. Whereas for the analyzation, the researcher [8] needs to be analytical for diagnosing and detecting the attack on a pre-developed system (like ELK stack) to save time and become more efficient.

Another framework as anticipated by [9] and [10] for network traffic analysis focuses on handling high speed network traffic with links working at 100 GB/s. The author proposes a flow-based modular Network Measurements Analysis (NEMEA) system to overcome the situation. Their system also yielded good results but at the same time there are some drawbacks related to the server's resources. All the modules as proposed by [9] are required to run on a single server with 6 CPU cores and 12 GB of memory and consume approximately 40% of the resources whereas in this project, the process is purely done on a virtual machine provided with 4 CPU cores and 8GB of memory on a single server and is able to process all incoming data without any loss.

3.2 Overcoming Existing state-of-the-art

Following *Table 1* outlines the list of existing patents-directly or indirectly related to the project (with proper citation), the known ways about how others have tried to solve the same or similar problems, their respective drawbacks and how any prior art documentation or other material (including this research) overcomes it.

Table 1. Overcoming the difficulties as faced by the existing-state-of-the-art

S. No.	Existing state-of-the-art	Drawbacks in existing state-of-the-art	Overcome
1.	CORE: A real-time network emulator [6]	Very technical, complex and time taking network-based emulator	Using Kafka CLI for swift event streaming in a modest way
2.	An innovative approach for real-time network traffic classification [7]	Using complex theorems and algorithms for aggregation, more error-prone, time consuming and demanding large knowledgebase	Presenting trouble-free real-time visualization and analyzation with the art of Kibana API

3.	Real-Time Diagnosis of Network Anomaly Based on Statistical Traffic Analysis [8]	Statistical measurement process taking vital time and inviting slip-ups, therefore, increasing the possibility of getting incorrect results	Minimal possibility of errors with efficiency of Elasticsearch stash using well-organized Logstash
4.	NEMEA: A framework for network traffic analysis [9]	Requirement of minimum 6 CPU cores and 12 GB of memory to be able to process all incoming data without any loss	Combining the power and efficiency of open source Tshark, Kafka and the ELK Stack to provide effective and lossless fallouts of all event streams

4. METHODOLOGY

4.1 Explaining the Methodology

Providing a clear and simple explanation of how this research solves the problem(s) cannot be done without considering event streaming and pipelining framework. Considering *Fig.1* for describing the agenda and the elementary steps, starting from capturing the live data packets to finally analyzing it in various forms (as done by the analyst) for future predictions (at real-time), becomes the foremost step of entering the "Big Data."

The data is being immensely generated at every single moment and that too at a reckless pace, it becomes very important to collect that data. But the main procedure starts when the data cleaning takes place. It is the most critical step for every analyst as it is very important to filter out the needful data and trash the rest. After perfectly collecting and cleaning the recorded or live streamed data, it turns out to be the right time for entering the analyzation phase. Here, the researcher or the analyst, with the help of different graphs (vertical as well as horizontal bar or line graphs), pie charts, tree maps and various other helpful visualizations, analyses the captured data in a very effective way and then lastly, tries to make the most accurate predictions, based on his/her abilities and analytical skills. Moreover, implementing and visualizing the right data model becomes yet another important step for faultless predictions.

4.2   Technical Features & Elements

As implemented and featured in *Fig. 1* as well as in [11] , the process starts with capturing or tracing out the live data packets with "Tshark" (Wireshark CLI), a widely-used network protocol analyzer. It acts as a producer for "Kafka." Kafka is very powerful for event streaming and maintaining logs. To view the live packets being streamed in Kafka, Kafka-consumer provides a very efficient CLI as well as API. Now, the packets as consumed in Kafka are sent to "Elasticsearch" where the JSON formatted data is indexed and stored into very easy and reliable format for performing different analyzations in "Kibana" via "Logstash", which is yet another free and open source server-side data processing pipeline that ingests data from a multitude of sources (Kafka in our case), transforms it, and then sends it to your favorite "stash" (Elasticsearch in our case). Finally, exploring and analyzing of the data with stunning visualizations in Kibana makes the process complete and starts the job for the analysts.

4.3   Process Block Flow Diagram
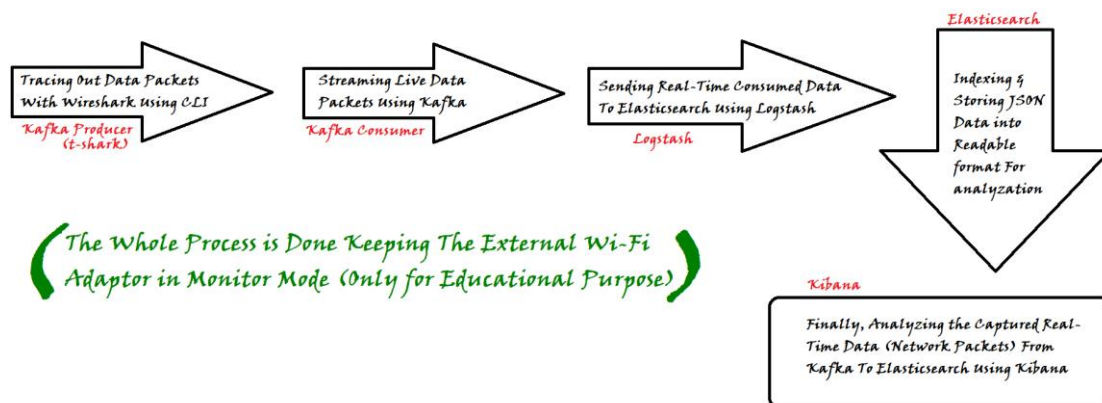
Fig. 1. Process Block Flow Diagram

4.4   Hardware & Software Components

Aiming for generating, collecting, processing, cleaning, visualizing/analyzing and then finally concluding with the utmost accurate predictions at real-time as shown in *Fig. 1*, becomes very important to have the perfect hardware as well as the software components. Basic requirements for the hardware comprise of a minimum of 4 CPU cores and 4GB of memory on a single server (recommended for linux users). As network packet capturing (pcap) comes under the umbrella of cyber security, nobody can deny the fact of recommending linux based operating system

for this purpose. Hence, making it a collaborative work for the "Ethical hackers" and "Data Scientists."

In view of the software requirements, there is no hard and fast rule of using a specific tool or a technology. But, combining the power of the perfect pack enhances its features. Considering null pricing and utmost enhancements, the combination of open-source CLIs like Tshark, Kafka and APIs like the ELK statck (Elasticsearch, Logstash and Kibana) is used in this research.

4.5 New Features Over the Closest Technology

Believing over the new technology has always been treated as an effective solution. But the trust develops once the innovation is implemented and gives the most successful results. Analyzations and predictions made over such effectual results, makes the technology utmost effective. Comparing the power of Tshark, Kafka streams and ELK stack with the closest technology, it is necessary to include the hardware requirements as well as their respective efficiencies to create a great difference. Keeping this concern in mind, the accuracy of Tshark packet capturing (pcap), the fleetness of Kafka streams (streaming different topics at the same time and inviting all sorts of developers with its wide acceptance) and with the precision of the ELK stack (the perfect pipelining, indexing and visualizing/analyzing framework), it becomes very difficult to compare it with other open-source tools or applications (either CLIs or APIs).

4.6 Alternative Ways

It is mostly not very difficult to implement an identical idea, which may be already known to another person. But compiling the right way of implementation matters. One should be smart enough to place the perfect set of applications or tools depending upon their respective usage and procedure. There are many such open-source applications or tools (like Clickhouse, Apache Superset, Confluent platform, KsqlDB) which can perform the similar approach (like in [12]). But then again, the right sequencing and perfect implementation should be done to get the most precise results with accurate predictions. Not knowing the operation, as explained in this report, could become henceforth a great drawback for such a doppelganger.

4.7 Status

Compiling and executing the real-time research as performed in this project has been successfully tested and implemented. Providing the particulars and demonstrations for the same would include the GitHub profile under the Big-Data-Analytics repository. All the evidences, in the form of screenshots (images) are present with all the respective commits including the documentary evidence of this event with exact timestamps.

5. RESULTS AND DISCUSSION

Using Ubuntu 20.04-LTS (Linux operating system) for the purpose of presenting the demonstration. Following steps are designed for the real-time network traffic analysis for threat detection.

5.1 Starting & Examining ELK Stack Daemons (Running)

Checking the state (up) of ELK stack by the following commands as shown in *Fig. 2*.

(1) sudo systemctl status logstash.services

(2) sudo systemctl status elasticsearch.services
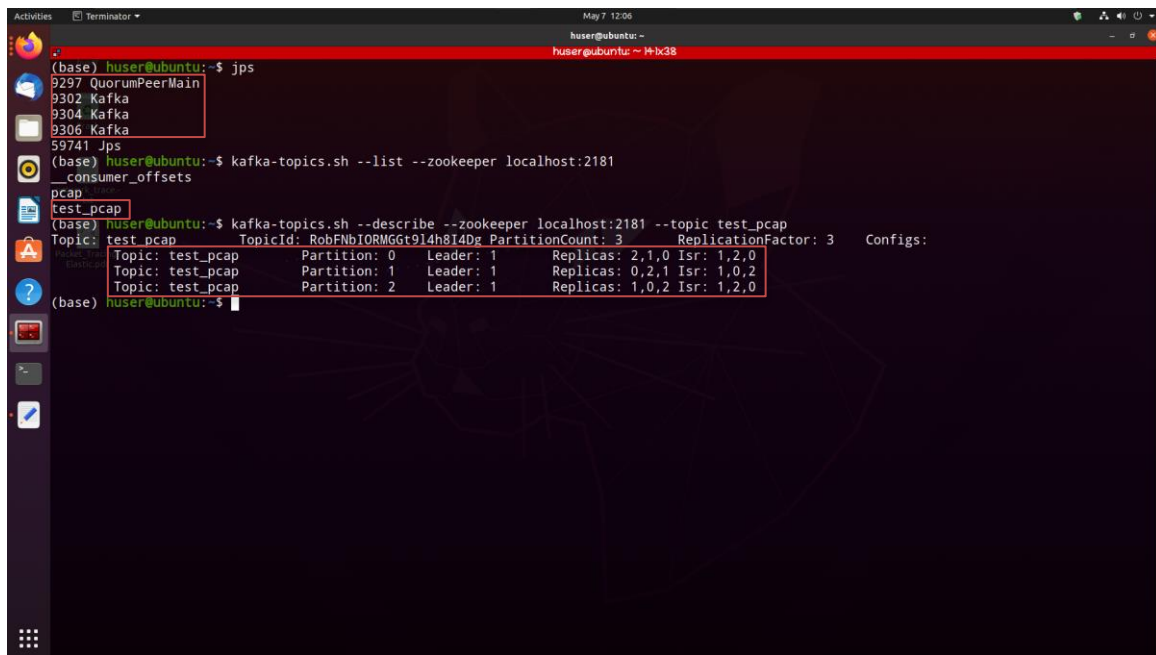
(3) sudo systemctl status kibana.services

Fig. 2. Status of ELK Stack Daemons (Running)

5.2 Starting & Examning Kafka Daemons (Running)

In *Fig. 3*, checking the kafka daemons by 'jps' command and ensuring that the state of zookeeper and kafka (3 brokers) is up with their respective 'ProcessIDs' (as in *Fig. 3*) and 'Port numbers' (as in *Fig. 4*).

(1) 'kafka-topics.sh --list --zookeeper localhost:2181' for listing all the available kafka topics with presence of zookeeper on localhost port number 2181 (as in *Fig. 4*).

(2) 'kafka-topics.sh --describe --zookeeper localhost:2181' --topic test_pcap for describing the 'test_pcap' kafka topic with presence of zookeeper on localhost port number 2181 (as in *Fig. 4*), a replication factor of 3 and partitions count 3.

Fig. 3. Checking Kafka Brokers, Zookeeper & Listing/Describing Kafka Topics

5.3 Retrieving Server Ports With 'netstat'

Getting the details of the server port numbers for Kafka brokers (9092, 9093 and 9094), Zookeeper (2181) and the ELK Stack (9600 for Logstash, 9200 for Elasticsearch and 5601 for Kibana) running on localhost as shown in *Fig. 4* using the command 'netstat -plntu'.

Fig. 4. Port Numbers for Kafka Brokers, Zookeeper, and the ELK Stack

5.4 Configuring Logstash Pipeline

Keeping the input as 'kafka' & output as 'elasticsearch' for the logstash (pipeline) configuration file. In addition to that, setting the bootstrap servers (localhost:9092, localhost:9093 and localhost:9093) as in *Fig. 4*, the topic name (test_pcap) as shown in *Fig. 3* and the encoding of the captured data (JSON) for kafka. Similarly, for elasticsearch, keeping its respective host (localhost:9200) as already shown in *Fig. 4* and the encoding of the data (JSON) to be indexed at 'pcap' for further analyzations to be done in the Kibana application. *Fig. 5* demonstrates the same in the logstash's configuration file present at '/etc/logstash/conf.d/kafka-pcap.conf'.

Fig. 5. Logstash Config File in /etc/logstash/conf.d/kafka-pcap.conf

5.5   'ifconfig' Before External W-Fi Adaptor

Using 'ifconfig' for getting the interface configurations (broker0, docker0 , eth0, lo) without the external Wi-Fi adaptor as illustrated in *Fig. 6*.
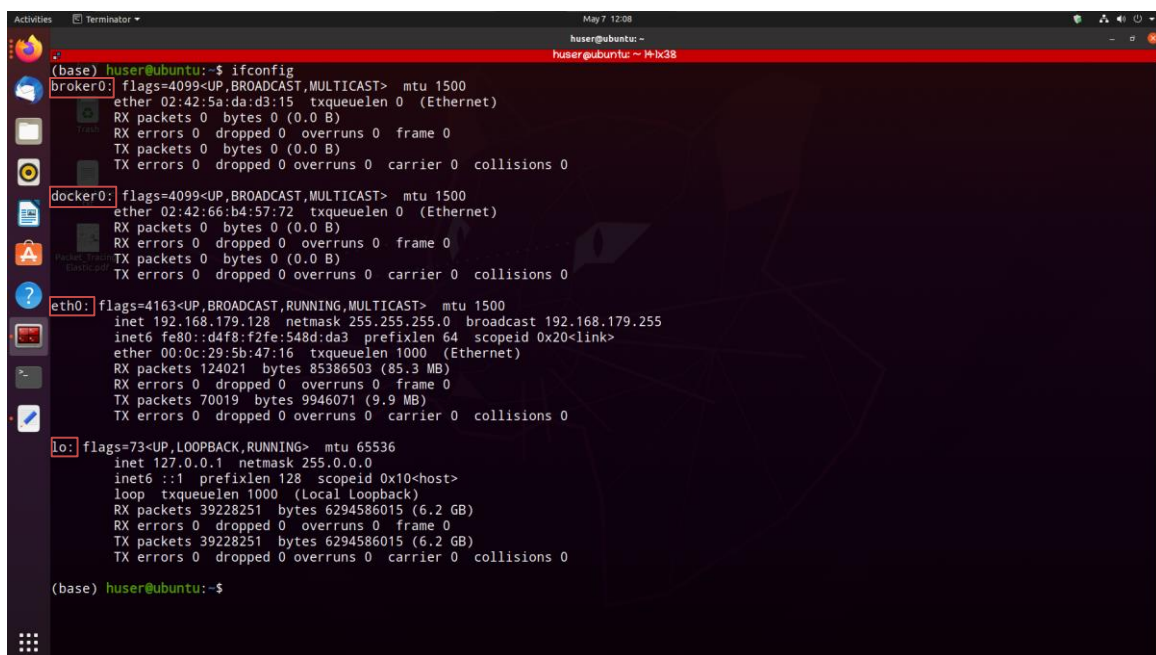
Fig. 6. 'ifconfig' Without External Wi-Fi Adaptor

5.6   'ifconfig' After External W-Fi Adaptor

Using 'ifconfig' for getting the interface configurations (broker0, docker0, eth0, lo) after inserting the external Wi-Fi adaptor (known as wlan0) as shown in *Fig. 7*.

Fig. 7. 'ifconfig' After Inserting the External Wi-Fi Adaptor (wlan0)

5.7 External W-Fi Adaptor in Managed Mode

Default state (Managed) of the external W-Fi adaptor (wlan0) as clearly marked in *Fig. 8* using 'iwconfig wlan0' command.

Fig. 8. wlan0 in Managed Mode

5.8 Configuring 'wlan0' in Monitor Mode

Enabling monitor mode on wlan0 with the following set of commands (mandatory to be used in sequence) as shown in *Fig. 9*.

(1) 'sudo ifconfig wlan0 down' for making the wlan0 down.

(2) 'sudo airmon-ng check kill' for killing unnecessary services that may harm the monitoring interface.

(3) 'sudo iwconfig wlan0 mode monitor' for setting the mode for wlan0 as monitor.

(4) 'sudo ifconfig wlan0 up' for finally making the wlan0 in up state with enabled monitor mode.

Hence, performing 'iwconfig wlan0' to check whether the set of commands executed successfully or not.

Fig. 9. wlan0 in Monotor Mode

5.9 Configuring 'wlan0' in Monitor Mode

Using the following set of commands (as done in *Fig. 10*) for capturing (with tshark) and streaming (using 'kafkacat') the data packets (in any network within range) in kafka.

```
sudo tshark -i wlan0 \
        -T ek \
        -l \
        -e wlan.fc.type -e wlan.fc.type_subtype \
        -e wlan_radio.channel \
        -e wlan_radio.signal_dbm -e wlan_radio.duration \
        -e wlan.ra \
        -e wlan.ra_resolved -e wlan.da -e wlan.da_resolved \
        -e wlan.ta -e wlan.ta_resolved -e wlan.sa \
        -e wlan.sa_resolved -e wlan.staa \
        -e wlan.staa_resolved \
        -e wlan.tagged.all -e wlan.tag.vendor.data \
        -e wlan.tag.vendor.oui.type \
        -e wlan.tag.oui -e wlan.ssid \
        -e wlan.country_info.code \
        -e wps.device_name |\
    grep timestamp|\
    jq -c '{timestamp: .timestamp} + .layers' |\
    kafkacat -X api.version.request=false\
        -b localhost:9092 \
        -P \
        -t test_pcap \
        -T
```

Here, the resolved distribution of MAC addresses of senders are referred as 'sa' and that of receivers are 'ra' or 'da' (enhanced results with resolved names). The most common filter trafic (wlan.fc.type & wlan.fc.type_subtype) can be referred to the Wireshark Display Filters. Similarly, for the other commands, tshark man-pages and kafkacat man-pages would be really helpful as in [13].

Finally, using 'kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test_pcap' for consuming and streaming the captured packets in Kafka consumer console. An example of a packet consumed (in real-time) is highlighted in *Fig. 10*.

Fig. 10. tshark & kafka-consumer

5.10 Discovering Live Data Using Kibana (With Examples)

As a result, getting the consumed data packets indexed in elasticsearch and hence, inspecting it in real-time using the Kibana application (on localhost:5601 as in *Fig. 4*). *Fig. 11*, *Fig. 12*, *Fig. 13,* and *Fig. 14*, show some of the examples of live streaming of data as indexed (in raw format as well as in filtered out tabulated format) in elasticsearch (on the kibana's discover console). Using filters are clearly mentioned in *Fig. 10*.
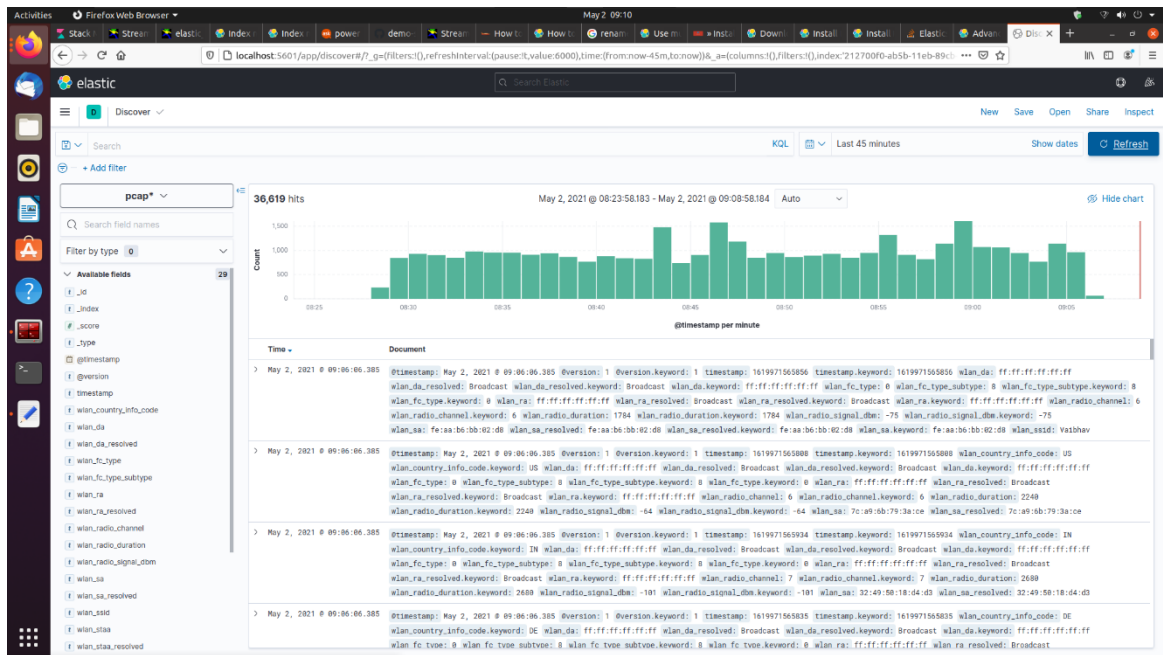
Fig. 11. Streaming Raw Data in Kibana

Fig. 12. Filtering Out Useful Data in Tabulated Form
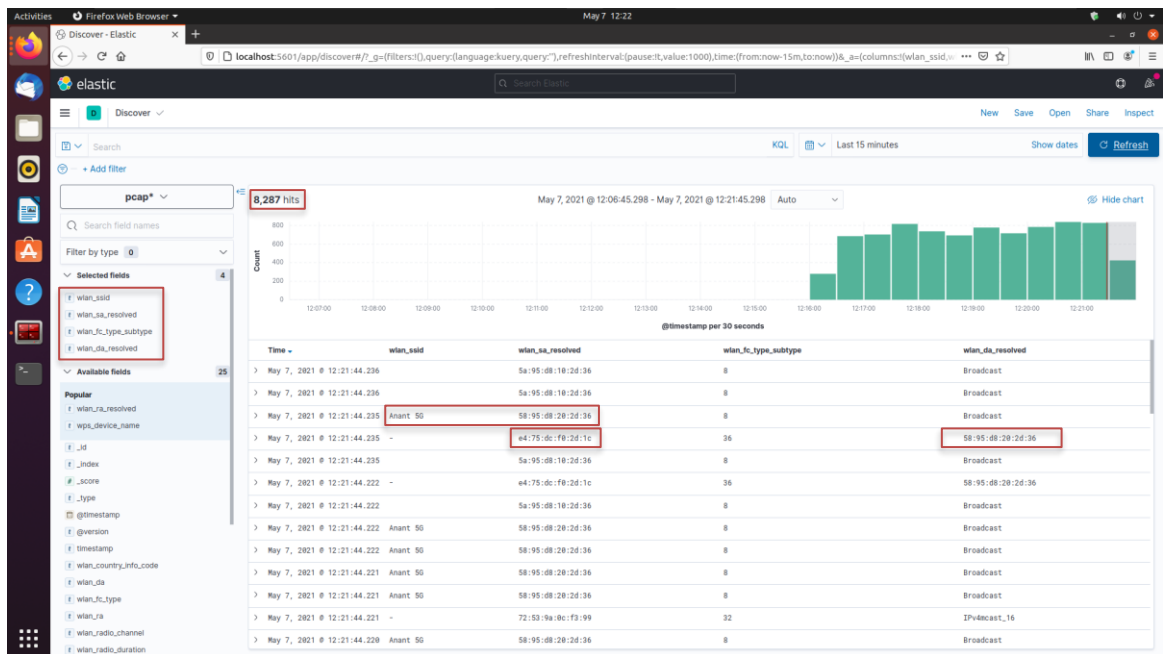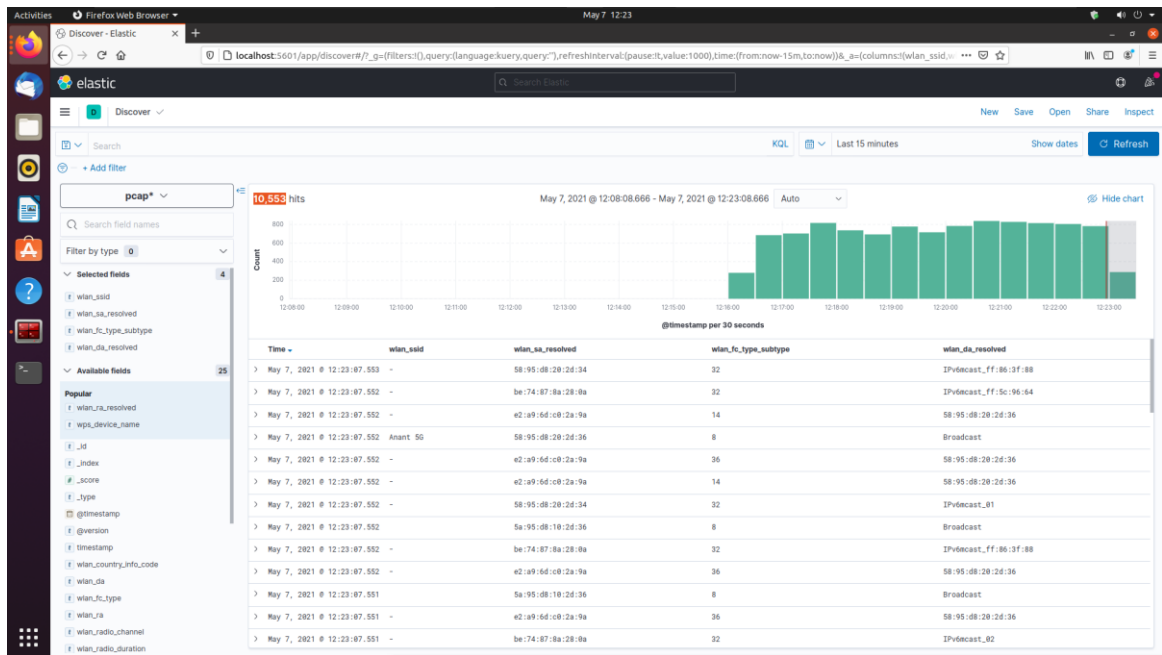
Fig. 13. Some Interesting 'da' Receivers with Resolved Names
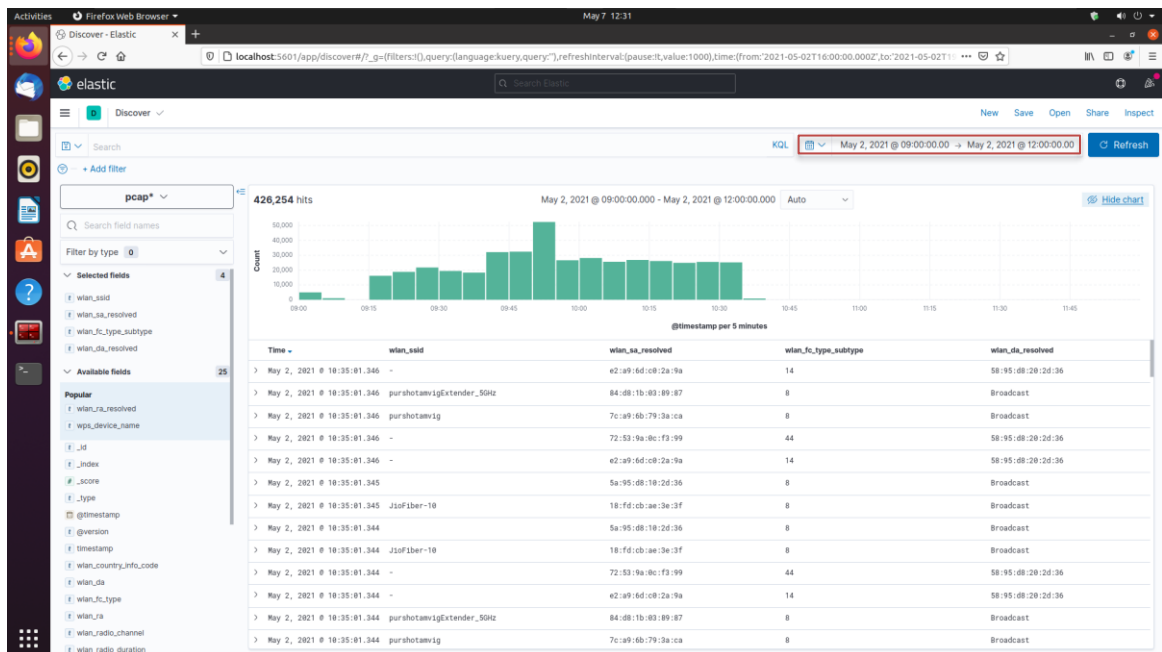
Fig. 14. Filtering Captured Data for Certain Time Frame

5.11 Visualizing Data Using Kibana in Different Time Frames

Fig. 15, and Fig. 16 are some of the examples of heat map visualization being generated in different time frames in Kibana by specifying the metrics, terms and many more useful fields.
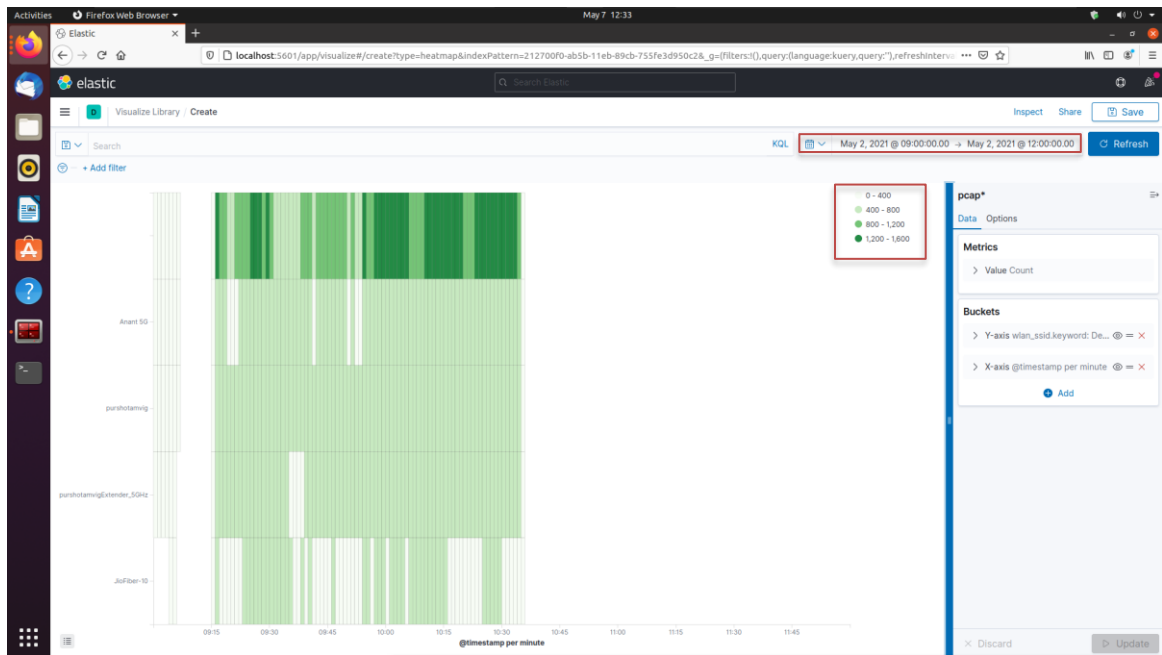
Fig. 15. Heat Map of Data Usage for Up to 5 Available Wi-Fi Devices
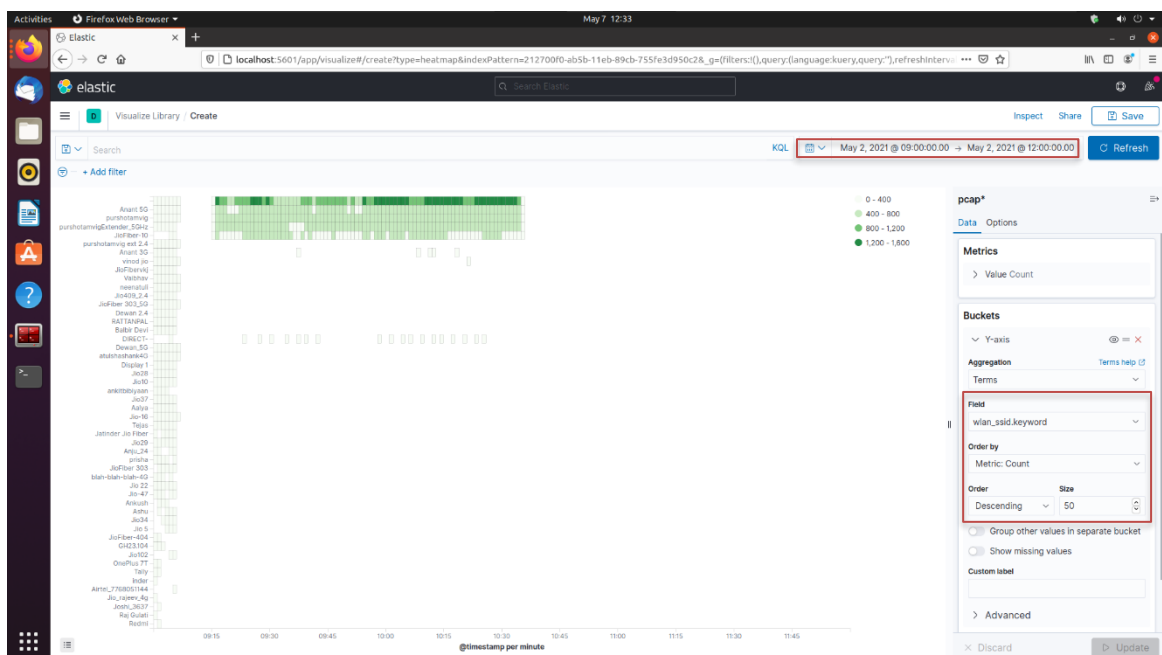
Fig. 16. Heat Map for Up to 50 Available Wi-Fi Devices

5.12 Analyzing & Making Predictions Using Custom Kibana Dashboards

Finally, analyzing the dataset and making precise predictions with the help of various visualizations as offered by Kibana. *Fig. 17* shows an example of a custom

dashboard built to analyze the data in a rich format and predict the attacker with accurate predictions and secure the environment by constantly examining the case.
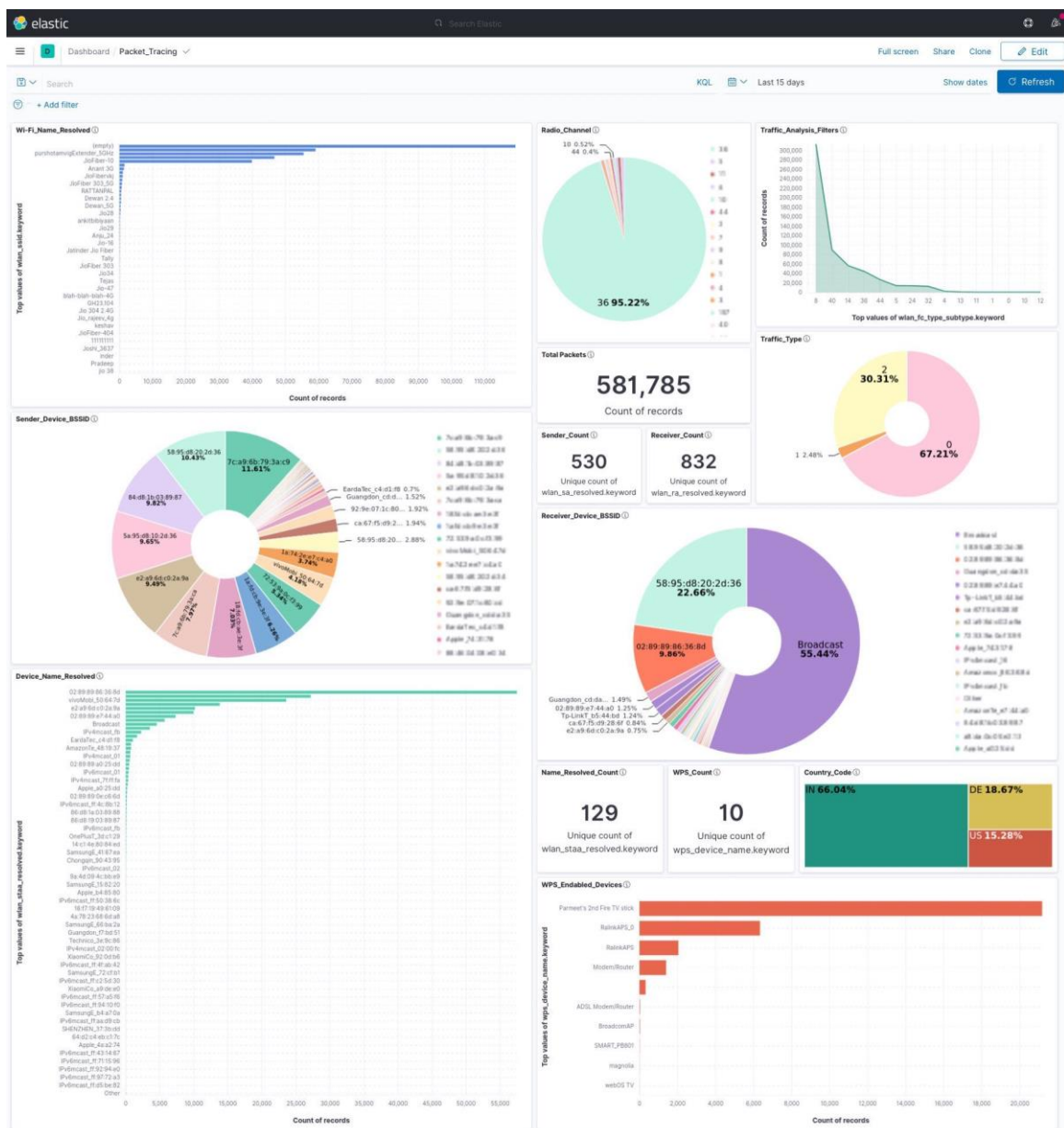
Fig. 17. Custom Dashboard Built in Kibana For Analyzations & Predictions

6. CONCLUSONS AND FUTURE WORK

Why should anybody care about the "Big Data"? Still this question has its own value. Every data scientist can present his/her own perspective. Numerous questions that are still unanswered have meaning and respective answers within the question. But one should be able to make it out in the right sense. Answering some of the queries with its reasoning as well as performing some analyzations with appropriate predictions are vibrant in this

research. Different examples for discovering and envisaging the real-time data in various forms are premeditated. Hence, the estimation of the right black hat can be done with specific analytics and skills depending upon the analyst or the researcher. Making "Big Data" and "Cybersecurity" work with each other was not easy. With the power of T-shark, it became unchallenging to capture out the flowing data packets in the range of the external Wi-Fi adaptor (wlan0). Whereas streaming and indexing the data (JSON formatted) resulted out to be trouble-free with the immensity of kafkacat (Kafka CLI) and the ELK stack respectively. Analyzations and predictions still depend upon the capability of the researcher. But using the right tool is still important. Visualizing and analyzing with the massive Kibana framework makes it a piece of cake. In an ethical way, every analyst can catch out the suspicious activity happening in any network (within range of the external Wi-Fi adaptor). Using the latest and strongest chipset of the adaptor can also help to increase the bounds of the packet capture (pcap). Henceforth, predicting and catching out the cybercriminals in any network turns out to be the main moto of this research. So now, the cybercrime is at stake and efficient analysts in demand. Always performing such activities for ethical reasons becomes the most important takeout. At last, it can be concluded that "Cybersecurity" can be the tip of the iceberg in the world of "Big Data."

REFERENCES

[1] "IPR2019-01327, No. 1006-6 Exhibit - FT 1006 PA Porras and A Valdes, Live Traffic Analysis of TCP IP Gateways, In Proc Symposium on Network and Distributed System Security (P.T.A.B. Jul. 11, 2019)." https://www.docketalarm.com/cases/PTAB/IPR2019-01327/Inter_Partes_Review_of_U.S._Pat._7370358/07-11-2019-Petitioner/Exhibit-1006-6-FT_1006_PA_Porras_and_A_Valdes,_Live_Traffic_Analysis_of_TCP_IP_Gateways,_In_Proc_Symposium_on_Network_and_Distributed_System_Security/ (accessed May 08, 2021).

[2] S. M. Eisenman, X. Fei, X. Zhou, and H. S. Mahmassani, "Number and Location of Sensors for Real-Time Network Traffic Estimation and Prediction," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1964, no. 1, Jan. 2006, doi: 10.1177/0361198106196400128.

[3] Scong Soo Kim and A. L. N. Reddy, "A study of analyzing network traffic as images in real-time," doi: 10.1109/INFCOM.2005.1498482.

[4]   M. R. Joshi and T. H. Hadi, "A Review of Network Traffic Analysis and Prediction Techniques."

[5]   N. al Khater and R. E. Overill, "Network traffic classification techniques and challenges," in *The 10th International Conference on Digital Information Management, ICDIM 2015*, Jan. 2016, pp. 43–48, doi: 10.1109/ICDIM.2015.7381869.

[6]   J. Ahrenholz, C. Danilov, T. R. Henderson, and J. H. Kim, "CORE: A real-time network emulator," Nov. 2008, doi: 10.1109/MILCOM.2008.4753614.

[7]   K. L. Dias, M. A. Pongelupe, W. M. Caminhas, and L. de Errico, "An innovative approach for real-time network traffic classification," *Computer Networks*, vol. 158, pp. 143–157, Jul. 2019, doi: 10.1016/j.comnet.2019.04.004.

[8]   L. Liu, X. Jin, G. Min, and L. Xu, "Real-Time Diagnosis of Network Anomaly Based on Statistical Traffic Analysis," Jun. 2012, doi: 10.1109/TrustCom.2012.233.

[9]   T. Cejka, V. Bartos, M. Svepes, Z. Rosa, and H. Kubatova, "NEMEA: A framework for network traffic analysis," Oct. 2016, doi: 10.1109/CNSM.2016.7818417.

[10]   J. R. Goodall, W. G. Lutters, P. Rheingans, and A. Komlodi, "Focusing on context in network traffic analysis," *IEEE Computer Graphics and Applications*, vol. 26, no. 2, pp. 72–80, Mar. 2006, doi: 10.1109/MCG.2006.31.

[11]   A. M. Karimi, Q. Niyaz, W. Sun, A. Y. Javaid, and V. K. Devabhaktuni, "Distributed network traffic feature extraction for a real-time IDS," in *IEEE International Conference on Electro Information Technology*, Aug. 2016, vol. 2016-August, pp. 522–526, doi: 10.1109/EIT.2016.7535295.

[12]   M. Sullivan and A. Heybey, "Tribeca: A System for Managing Large Databases of Network Traac." Accessed: May 08, 2021. [Online].

[13]   Yong Guan, Xinwen Fu, Dong Xuan, P. U. Shenoy, R. Bettati, and Wei Zhao, "NetCamo: camouflaging network traffic for QoS-guaranteed mission critical applications," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 31, no. 4, Jul. 2001, doi: 10.1109/3468.935042.