The world being represented is an application capable is gathering network traffic data from a network security device (such as a router or firewall) and parsing through the traffic in real time. This is known in the real world as a network Intrusion Detection Service (IDS) tool The application will then be able to use a set of sensors to flag the data as either benign or malicious. The traffic captured is benchmarked against a time stamp (known as the system time). The traffic consists of a set of attributes under three categories which map the ISO layers 2-4:

1. 1.) Frame; which contains attributes representing the data required for node to node data transfer between a network. The attributes include interface id (the network interface the traffic arrived at), arrival time, Frame Number, Frame length, frame type (ethernet, wireless etc.)
2. 2.) Ethernet: this contains attributes representing the data required for addressing, routing and traffic control between computer networks. It includes data like the Destination and Source address and network protocol used
3. 3.) Transport: this contains attributes representing the data required for transferring data while maintaining the quality of service functions. It includes flags designed to provide connection (with recovery) or connectionless oriented data transfers. Some attributes include: TTL, Protocol, Total length, Source Address, Destination Address.
4. 4.) User Datagram: This contains attributes to identify where the processing on the host and remote computer will be. It contains the source port, destination port and length

The application allows the administrator to pin network packets to a network based on the CIDR reported as the source IP. For example, the administrator might assign 10.12.0.0/16 range as belonging in their DMZ. However, they can also specify that 10.12.135.0/24 belongs to the Shanghai DMZ.

Once the data is gathered the application contains a list of sensors which can mark data transfers as being benign or malicious based on a set of attributes the sensor contains. For example, a sensor could detect connection to a botnet using a list of known malicious IP address (Internet Protocol address). If the Transport or Ethernet attributes contains an IP address in the backlist, then the sensor will flag it as malicious. Off importance if that the sensor can flag traffic by specifying a range of packets or one. One of more sensors may flag a traffic based on packet and network. Conversely, any given traffic packet might not be flagged by any sensor at all.

Also, within the application is a set of response that the sensor can trigger if it finds a match from the network traffic. For example, the sensor that filters the traffic for connections to known malicious IP can consist of several response actions such as add the traffic Id to a pre-made report template, fire an email send a notification to the application. The response is also optionally limited with a

threshold given a timespan. For example, the response may be limited not to fire more than 30 times (the threshold) in a 30-minute (timespan) interval.

Steps:

1. First once I used Docker to build an image for my application.
2. Then I built my docker image using the following command: docker build -t ids_image .
3. Running the container as:
    1. docker run --rm --name ids -it ids_image
4. select * from packets order by packetid desc limit 100;

Data Source from CIC IDS dataset: CSE-CIC-IDS2018
AWS Link https://registry.opendata.aws/cse-cic-ids2018/
Full Article: https://www.unb.ca/cic/datasets/ids-2018.html

The idea of this project. We are attempting to decide whether an IP within the network is now part of a botnet. We do this by detecting
whether it is communicating to a known command and control server based on an IP blacklist and the network segment the host is located in.
We decide the network into various Network Hierarchies which denote function of machines in those networks and their importance.
For example we might have a network for the HR department, for Administrative, for the DMZ etc.

The project will examine the traffic from a given network, determine if the server lies in one of the organizations critical infrastructure pieces (using the source IP network CIDR as a feature) and then checks how many times within the last minute, 5 minutes and 15 minute the host has being in communication with the command server (as another feature). Finally we check if the number of bytes transferred between the botnet command within the 1 minute, 5 minute and 15 minute interval (as the 3rd feature). If any two of these numbers are above a threshold, the IDS will raise an alarm on the entire traffic between the two hosts.

The second proposal is to perform network anomaly detection that examines three properties of a network connection:
1.) The number of unique destination internet communicated to by each source address in the organization’s network
2.) The number of bytes transferred by the host in the organization’s network and an external host on the internet
3.) The number of connections between the host in the organization and an external host (internet)

We weight these three on a 1 minute, 5 minute and 15 minute average and see if the number exceeds set thresholds determined by the administrator.
That is a standard deviation based threshold.

ARP SPoofing:
ARP spoof looks like this:
192.168.2.1 is at ab:ab:ab:ab:ab:ab
And
192.168.2.1 is at 10:10:10:10:10:10

https://www.howtogeek.com/107945/how-to-identify-network-abuse-with-wireshark/

References:

Bhuyan, Monowar H., Dhruba Kumar Bhattacharyya, and Jugal K. Kalita. "Network anomaly detection: methods, systems and tools." *Ieee communications surveys & tutorials* 16.1 (2013): 303-336.

Barford, Paul, and David Plonka. "Characteristics of network traffic flow anomalies." *Internet Measurement Workshop*. 2001.

Jiang, Jun, and Symeon Papavassiliou. "Detecting network attacks in the internet via statistical network traffic normality prediction." *Journal of Network and Systems Management* 12.1 (2004): 51-72.

Navaz, A. S., V. Sangeetha, and C. Prabhadevi. "Entropy based anomaly detection system to prevent DDoS attacks in cloud." *arXiv preprint arXiv:1308.6745* (2013)

Galtsev, Aleksey A., and Andrei M. Sukhov. "Network attack detection at flow level." *Smart Spaces and Next Generation Wired/Wireless Networking*. Springer, Berlin, Heidelberg, 2011. 326-334.

Vykopal, Jan, Tomas Plesnik, and Pavel Minarik. "Network-based dictionary attack detection." *2009 international conference on future networks*. IEEE, 2009.

Munz, Gerhard, and Georg Carle. "Real-time analysis of flow data for network attack detection." *2007 10th IFIP/IEEE International Symposium on Integrated Network Management*. IEEE, 2007

Al-Jarrah, Omar, and Ahmad Arafat. "Network Intrusion Detection System using attack behavior classification." *2014 5th International Conference on Information and Communication Systems (ICICS)*. IEEE, 2014.

Wang, Ke, and Salvatore J. Stolfo. "Anomalous payload-based network intrusion detection." *International Workshop on Recent Advances in Intrusion Detection*. Springer, Berlin, Heidelberg, 2004.

Mantere, Matti, Mirko Sailio, and Sami Noponen. "Network traffic features for anomaly detection in specific industrial control system network." *Future Internet* 5.4 (2013): 460-473.

Mantere, Matti, Mirko Sailio, and Sami Noponen. "Network traffic features for anomaly detection in specific industrial control system network." *Future Internet* 5.4 (2013): 460-473.

Sperotto, Anna, et al. "An overview of IP flow-based intrusion detection." *IEEE communications surveys & tutorials* 12.3 (2010): 343-356.

Strayer, W. Timothy, et al. "Botnet detection based on network behavior." *Botnet detection*. Springer, Boston, MA, 2008. 1-24.

Mahoney, Matthew V., and Philip K. Chan. *PHAD: Packet header anomaly detection for identifying hostile network traffic*. 2001.
