Anomaly Detection in Network Activities

Adepu Geetha Saatvik, Kaki Haradeep

Chandigarh University, Chandigarh, India

Email: adepusaatvik@gmail.com

haradeepkaki2001@gmail.com

Abstract:

Our project concerns the field of Cyber Security and aims at detecting DOS or DDOS attacks. Although we had developed the project with the Institute Gateway in our mind, the software works pretty well with any other network or any personal computer. The main aim of our team was to build a software from scratch that is light, gives the user real time information about the said network and could work independently. The project presented below is developed with the sole motive of detecting a DOS attack by distinguishing it from legitimate network traffic efficiently. However, we aim at improving our software with the help of various deep learning techniques while keeping our software light and fast.

Introduction:

Network Behaviour Anomaly Detection (NBAD) affords a method to network security threat detection. It is a balancing technology to systems that determine security threats based on supported packet signatures. NBAD is the process of endlessly watching a network for unusual events or trends. NBAD is an integral part of network behaviour analysis (NBA), that offers security in addition to the services provided by ancient anti-threat applications like firewalls, intrusion detection systems, antivirus computer code, and spyware-detection computer code. Laptop security has become a necessity because of the proliferation of knowledge technologies in the standard of living. The mass usage of processed systems has given rise to crucial threats like zero-day vulnerabilities, mobile threats, etc. Despite analysis within the security domain having hyperbolic considerably, nonetheless, the threat often seems to be mitigated. The evolution of laptop networks has greatly exacerbated laptop security considerations, notably web security in today's networking setting and advanced computing facilities. Though web Protocols (IPs) weren't designed to position a high priority on security problems, network directors nowadays have to be compelled to handle an outsized style of intrusions made by people with malicious intents and huge botnets.

In educational analysis,[1] however, anomaly detection approach is perceived to be additionally powerful because of its higher potential to deal with novel attacks as compared to misuse-based strategies. According to threat Report of the Symantec web Security, there have been over three billion attacks of malware was reported in 2010 and therefore the variety of Denial-of-Service attacks increased hyperbolically by 2013 (Symantec web security threat report, 2014). As explicit in Verizon's knowledge Breach Investigation Report 2014, 63,437 security breaches were launched by hackers distributed throughout the world. The world State of knowledge security Survey 2015 (The international State of knowledge Security Survey, 2015) found a rise in the number of such incidents. Therefore, the detection of network attacks has given the much priority these days. Additionally, the experience needed to commit cybercrimes has diminished because of simply offered tools. Anomaly detection is a crucial knowledge analysis task that detects abnormal or malicious knowledge from a given dataset.

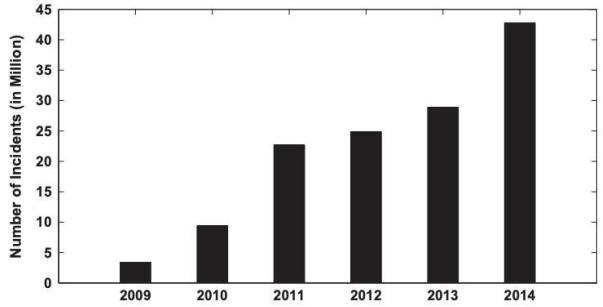


Fig. 1. Growth of information security incidents (The Global State of Information Security Survey, 2015).

Literature Review/Related work:

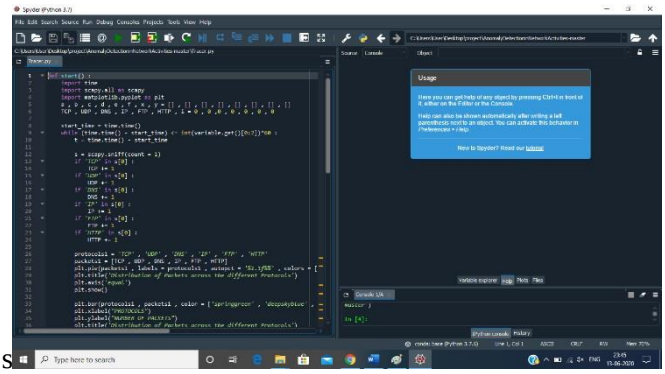
Innovative cyberattacks are featured by several layers, varieties and stages, with the goal of duplicitous the monitors. Existing of anomaly detection systems search typically traffics and logs alone for proof of attacks however ignore any analysis regarding attack processes.[2] As an example, the traffic observation strategies will solely detect the attack flows roughly however fail to provide us with detailed information about the attack. Most security observance systems utilize a signature-based approach to observe threats. They typically monitor packets on the network and appearance of patterns within the packets that match their information of signatures representing preidentified well-known security threats.[3] NBAD based systems are significantly useful in the detection of security threat vectors in two instances where signature-based systems can't be used: (i) new zero-day attacks, and (ii) once the threat traffic is encrypted as command and management channels. Example: Botnets and Spams. [10] With the entrance and volatile growth of the world wide web and ecommerce environments, adaptive/automatic network/service intrusion and anomaly
detection in wide space knowledge networks and ecommerce infrastructures is quick gaining vital analysis and is of global importance.

Data Resources:

The software tools used in our project are:

1. The Anaconda Installation of Python
2. The Spyder IDE

We have used the Anaconda Installation of Python since it comes pre-installed with most of the basic python libraries such as Pandas, Matplotlib, NumPy, etc. Another benefit is that the Anaconda installation comes with the Spyder IDE is a very useful tool since it helps us visualize the results of any part of our program code inside the same terminal in its kernel without having to run the whole code. It proves to be a very useful tool when it comes to visualizing Data Science results.



The Python libraries used in this project are:

1. The Scapy Library
2. The Tkinter Library
3. The Matplotlib Library
4. The Math Library
5. The Time Library

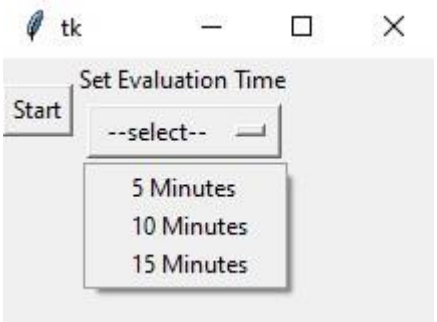
Methodology:

In our project, we make use of the Scapy Library of Python to sniff network packets. The Tkinter Library is used to make the GUI in python to provide the user with a more interactive interface.[4] The Matplotlib Library is used to plot the pie chart, the bar graph, and the time plot. The Math Library is used to do some mathematical formatting and the Time library is used to get the current time.

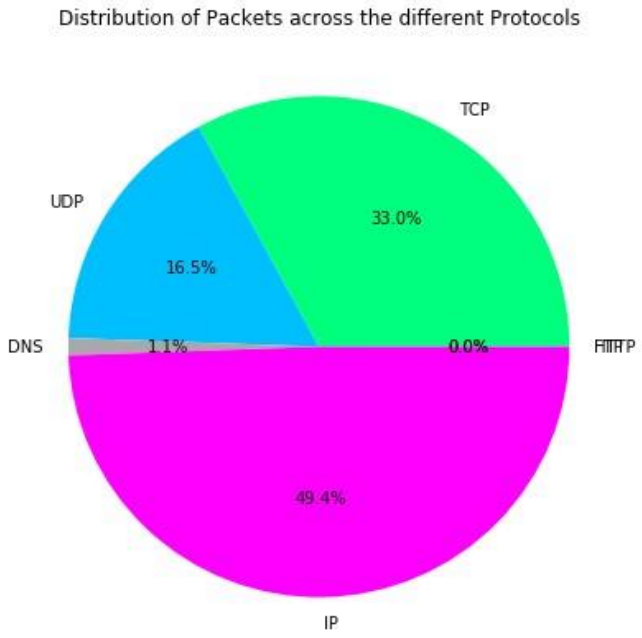
When we run the program, the main function of the code is executed first which results in a GUI popping up which consists of a column span of two using a Label Widget and a drop-down menu placed at grid position (3, 2) containing three options:

1. Five minutes, 2. Ten minutes, 3. Fifteen minutes using an Option Menu widget. The GUI also consists of a start button placed at grid position (2, 1) with a

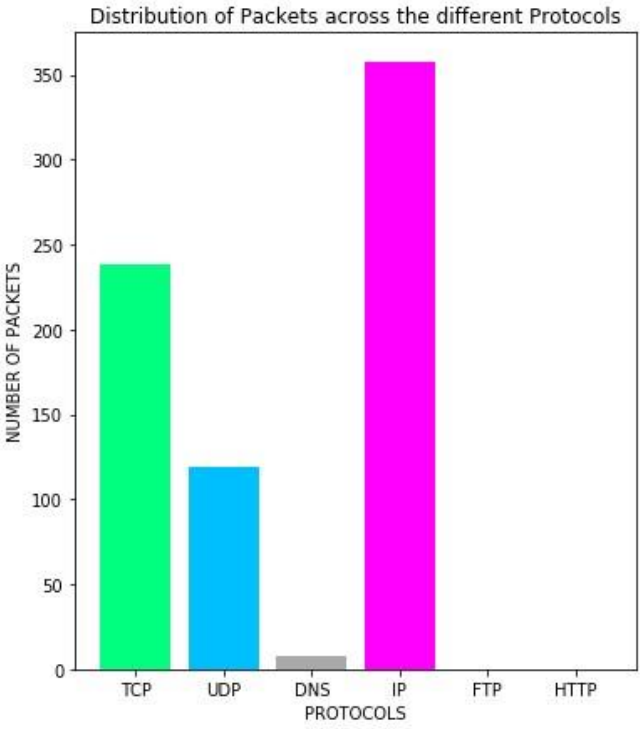
row span of 2 using a Button widget.



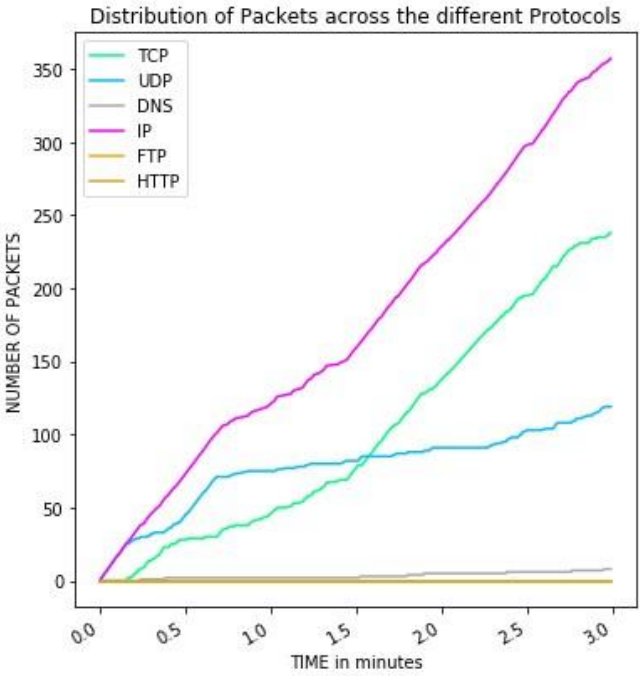
On selecting the time from the given options, and clicking on the Start button beside it, the program runs the start function which has been passed as an argument of the button widget B. The start function runs a loop for the amount of time that the user had selected in the GUI.[5] In each execution of the loop, the sniff function that has been imported from the Scapy Library captures a single network packet using the parameter count = 0 and searches for the protocols that are present in that packet and increases the number of packets in that protocol by one if that protocol is present in the packet. In this project we are sniffing packets with respect to six protocols namely: TCP, UDP, DNS, IP, FTP, and HTTP. [4] After updating the count for each protocol, the program plots the pie chart using the above information showing the percentages of packets of the total number of packets with respect to each protocol.



The bar graph is drawn considering the total number of packets in each protocol up to that time.



To plot the timeslot, we use a list for each of the six protocols and one list for the timestamp. We append the total number of packets for each protocol up to that time and timestamp into the respective lists after each iteration of the loop.



An IF condition at the tail of the loop checks for the number of packets sent in that second. After many iterations of the loop, it was found that in one second, a maximum number of 3 packets can be captured on one second for any legitimate request. So we have set the limit to three and whenever the number of packets

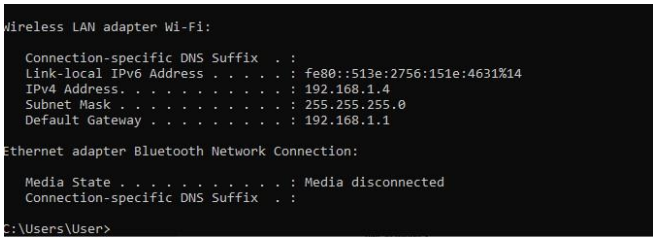
captured exceeds three, i.e.[9] whenever the floor value of the last timestamp in the list of timestamps is same as the floor value of the fourth last timestamp in the list, it indicates an anomaly in the network and the possibility of a DOS attack. If such an anomaly occurs, a message box - created with the help of the Tkinter Library of Python - pops up notifying the anomaly and the program breaks out of the loop.

If no such anomaly occurs, the program runs for the time selected by the user in the GUI.

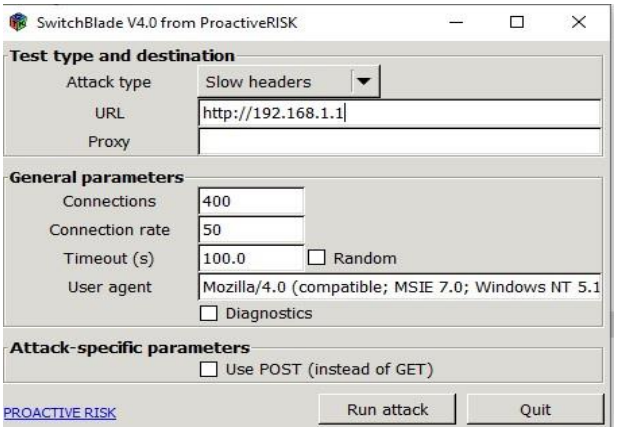
To put our program to test, we launched an artificial DOS attack onto our own computer and checked if our program worked correctly.

We can launch an artificial attack using software Switch Blade by following simple steps:

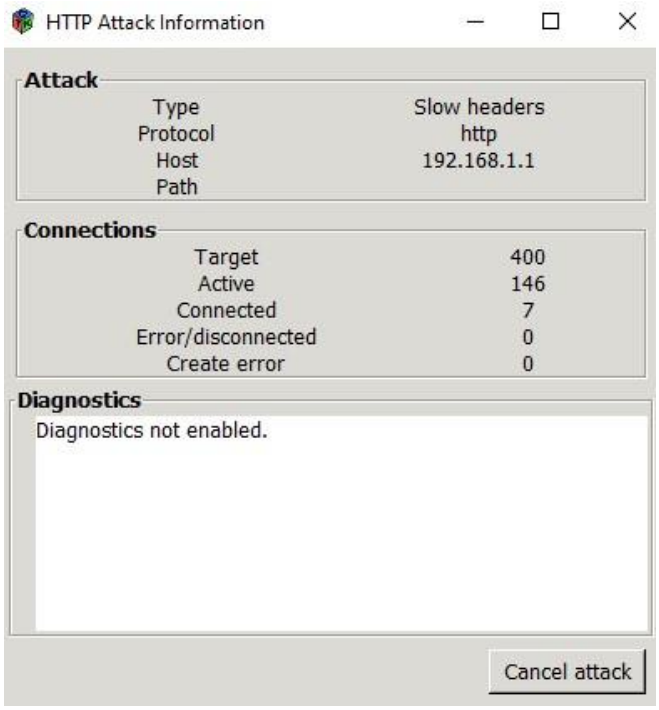
1. Open the command prompt/terminal on the computer
2. Enter the following command : ipconfig



3. Copy the default gateway address
4. Open Switch Blade GUI and type the address in the url section.



5.To start, click on the button run.

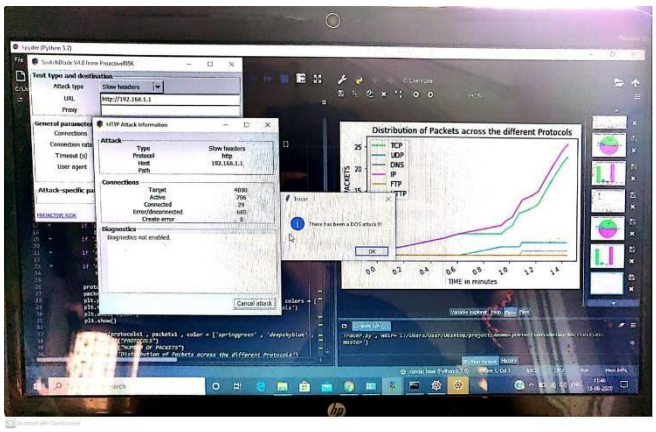


Now, in order to the attack to be have more effective, we must attack the target computer with the pings from another computer. The overhead attack can be used to attack the routers, web servers etc.

Results:

In our project, the pie chart indicates the percentage of the total number of packets that belong to each protocol. The bar graph gives us information about the total number of packets with respect to each protocol. The time plot gives us information about the total number of packets belonging to each protocol with respect to time spent after starting the evaluation. These plots change with each iteration of the above mentioned while loop and thus provide us with real time information about the number of packets with respect to each protocol.

Thus, our project helps users to keep track of their live internet traffic and detect an anomaly whenever there is a DOS attack.



Performance evaluation/validation:

Presently, there are many software which can detect a DOS attack. Most of the software available are actually packet sniffers. Some of the most popular software in this field are:

1. [6] Wire Shark is a mostly used as a network protocol analyser that lets you to monitor everything about your network and the packet flow going through it.
2. Tcp dump is a tool in command line that was primarily designed for UNIX operating systems, and It does not have a gorgeous user interface, but all packet data needed can be seen on display.
3. [8] SolarWinds examines content of the packet to regulate even the smallest detail as well as what applications cause the most traffic inside the network and which connections take the longest.

Microsoft has very recently come up with its Microsoft Azure DDOS Protection services which monitors as well as detects prevalent DOS or DDOS attacks. [7] As compared to other existing technologies, our software is much lighter, much faster and gives users the real time information about the network traffic in their computer.

Conclusion:

Our project is at the present moment in the early stages of its development. As of now our project is tailored to perform only those tasks that have been discussed above. It is yet to reach its full potential. The primary aspects of the project which we have decided to work upon are:

1. Faster computation speeds.
2. Lighter and more efficient software.
3. Ability to detect the name of the protocol that had been used to launch the attack.
4. Large scale implementation of our project in industrial servers.
5. For PC users, we have come up with an idea to tailor our software according to their daily needs. Internet usage is not the same throughout the day and often varies widely across the days of the week. So with the help of Artificial Intelligence we have attempted to study the daily and timely internet usage patterns of users and as a result be better able to differentiate between legitimate requests and attacks.

But, with respect to the cyber security community as a whole, as technology is increasing day by day, new methods and various machine learning techniques perpetually plan to improve the information discovery process. Given the actual fact that web traffic doubles every year and network traffic are increasing at a quick rate creating it a difficult task to watch a network in real-time. Existing anomaly detection techniques are largely for watching one system or one network by winding up native analysis for attacks. Hence, between instances of such standalone anomaly detection techniques,[11] no communication and interaction exist. Certainly, such an answer won't be able to sight subtle and extremely distributed attacks. Thus, for the safety of enormous networks and huge IT ecosystems (i.e. cloud services), cooperative techniques are very economical that comprise many monitors that act as sensors and collect knowledge. Due to the inconvenience of implementations of cooperative techniques like [12] CIDSs (Collaborative Intrusion Detection Systems), future analysis efforts are necessary for in-depth quantitative analysis with state-of-the-art network infrastructure.

Acknowledgement:

With the rapid increase in technology, there has been a rapid increase in the number of cases of cyber security related frauds. So, it is very important that we come up with new technology to counter this very problem. With this project, we have just tried our best to take the next big step in this direction. With this we would like to wind up our project.

References:

[1] Petersen, B. and Chung, E., Broadcom Corp, 2009. *Network activity anomaly detection*. U.S. Patent Application 12/015,387.

[2] Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G. and Vázquez, E., 2009. Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, *28*(1-2), pp.18-28. [3] Krügel, C., Toth, T. and Kirda, E., 2002, March. Service specific anomaly detection for network intrusion detection. In *Proceedings of the 2002 ACM symposium on Applied computing* (pp. 201-208).

[4] Simmross-Wattenberg, F., Asensio-Perez, J.I., Casaseca-De-La-Higuera, P., Martin-Fernandez, M., Dimitriadis, I.A. and Alberola-Lopez, C., 2011. Anomaly detection in network traffic based on statistical inference and\alpha-stable modeling. *IEEE Transactions on Dependable and Secure Computing*, *8*(4), pp.494-509.

[5] Gao, J., Hu, G., Yao, X. and Chang, R.K., 2006,

August. Anomaly detection of network traffic based on wavelet packet. In *2006 Asia-Pacific Conference on Communications* (pp. 1-5). IEEE. [6] Chan, P.K., Mahoney, M.V. and Arshad, M.H., 2005. Learning rules and clusters for anomaly detection in network traffic. In *Managing Cyber Threats* (pp. 81-99). Springer, Boston, MA.

[7] Noble, C.C. and Cook, D.J., 2003, August. Graphbased anomaly detection. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 631-636).

[8] Pavithirakini, S., Bandara, D.D.M.M., Gunawardhana, C.N., Perera, K.K.S., Abeyrathne, B.G.M.M. and Dhammearatchi, D., 2016. Improve the Capabilities of Wireshark as a tool for Intrusion Detection in DOS Attacks. *International Journal of Scientific and Research Publications*, *6*(4), pp.378-384.

[9] Othman, Z.A. and Eljadi, E.E., 2011, July. Network anomaly detection tools based on association rules. In *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics* (pp. 1-7). IEEE.

[10] Mantere, M., Sailio, M. and Noponen, S., 2014, April. A module for anomaly detection in ICS networks. In *Proceedings of the 3rd international conference on High confidence networked systems* (pp. 49-56).