ICA 1

16/03/2022

CIS1006-N-FJ1-2021 -
Secure Data Acquisition

Barclay Cousin – B1037353
B1037353@live.tees.ac.uk

As part of the task, I was assigned to carry out the cleaning and analysis of the KDD 1999 dataset (University of California, 1999) using Python's library, pandas. The dataframe in question is a simulated intrusion detection system for a military environment that was implemented over a nine-week period. An intrusion detection system monitors network traffic for suspicious activity or threats which in turn create alerts whenever malicious threats are witnessed (Pratt, 2018). Below I will detail key elements and findings of my assignment.

Upon the analysis of data, a selection of findings stood out. I concluded from a python pandas built-in function ("len(df['num_compromised')"), the dataframe contains 42 columns and 493,021 rows. The average duration of a connection was 47.9 seconds found from the ".mean" and "len[df['duration']])" command, thus meaning if a system vulnerability exploit was successful, an abundance of time is available for manoeuvre within the system. Of those connections, fifty-seven percent were made using the ICMP protocol (**Figure 1).** Research suggests that attacks relating to the ICMP protocol requires the server to process a request and respond, in turn consuming CPU resources. A common attack used that exploits this is a Smurf attack (Oriyano, 2016) which we have seen to be the most prolific out of all the attack types logged by the IDS. This attack type causes a denial of service. This concludes and outlines a direct link between the exploit of the ICMP protocol and the Smurf attack present within our dataframe. Furthermore, after investigation, no direct correlation has been showcased between the duration of an attack and the number of files that have been accessed meaning a length of a connection is not relevant in terms of accessing more system files. A clear finding is the severe vulnerabilities within the military environment where there were seventy-three thousand successful system logins **(Figure 2**) and a large variety of attack types faced(**Figure 3).** Attacks varied from Nmap reconnaissance to system denial of service which in turn led to downtime of critical military infrastructure.

Figure 1.

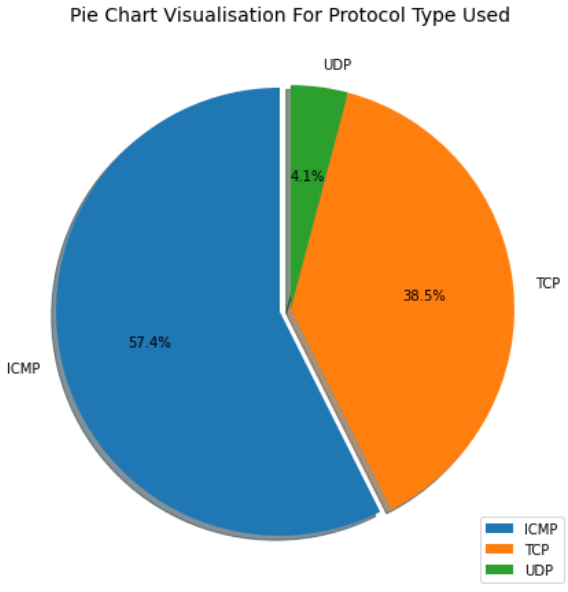


Figure 2.

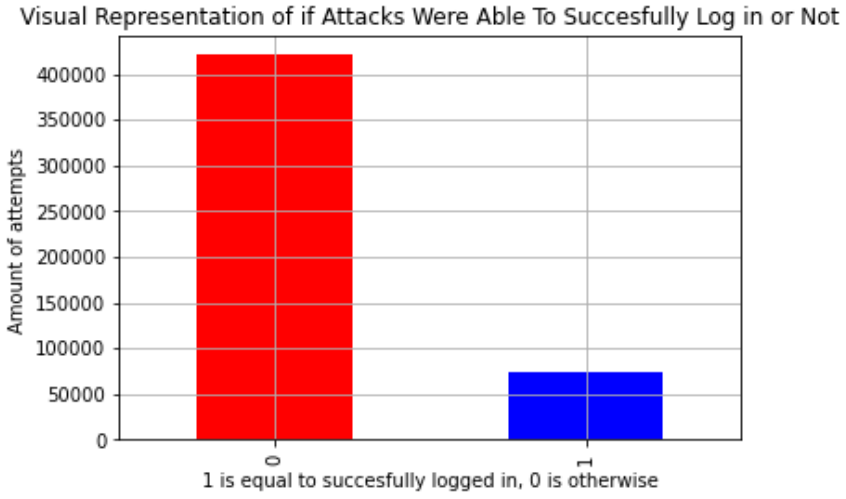
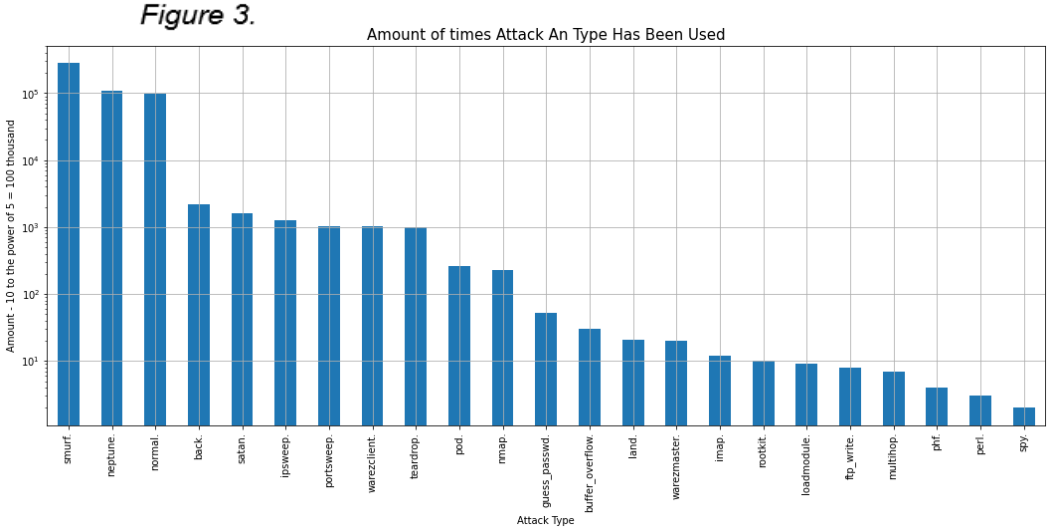


Figure 3.



When initiating the analysis of data, I was confronted with an abundance of issues. When the dataframe is downloaded, it isn't in a readable Pandas extension. In this occurrence, it was necessary to open and save in Microsoft excel with a .xlsx extension. Microsoft Excel was also used for the separation of the columns due to the columns initially not being separated, instead all in the first column. This was fixed by using the inbuilt feature of separating at the comma value. Without this step, no analysis could be completed within Pandas. Once loaded into Pandas and simple head exploratory analysis was tested, a clear problem arose. No columns were assigned names, leading to the inability to filter by column name. This issue was conquered by finding the names on the documentation for the dataframe and assigning them within pandas. The final step was to assign the unnamed end column a suitable name. After research, A pattern arose in the column entries and concluded it was attack types the environment faced. This was then assigned to the column. In this dataset, there were no null values. This was tested by using the ".unique" command. Null values are a large issue when it comes to cleaning data and can lead to issues regarding visualisation due to the inability to plot a null value. Insufficient cleaning leads to a waste of time and resources and can cause unnecessary issues for the user. Replacing null values is possible but unnecessary in this instance. Finally, at a later stage, the file was converted to a CSV for its enhanced processing speed relating to visualization as initially lots of time was wasted waiting for pandas to process as .xlsx.

To present my results I used the python library Matplotlib. Matplotlib allows for visualisations to be created from python code giving the user freedom regarding title names, colours and other aesthetics. My goal surrounding the presentation was to create an easy and pleasant viewing experience for the end-user while obtaining all knowledge required. For my text-based analysis, I opted against both a command-line user interface and GUI. In this instance, I believed it would be better suited for the user at the first execution of code to receive all figures. To allow for clear text visibility I made best use of line breaks and symbols to space out my findings to a point of easy read. Constant feedback to the user is important and is something I introduced. In respect of my visual presentation, I used a plethora of chart types where I deemed necessary. Careful selection was made regarding what data was presented using what graph. I believe the attack type figure is a prime example **(Figure 3).** A pie chart would not be useful as there are many types coming under two percent and would produce tarnished graphical results with section overlap. Instead, I presented in a bar chart and added the log feature which gives amount in a squared format allowing for a graph with visible results. I also added grids on graphs where necessary. Implementation of different colour schemes was vital to ensure colour blind compliance with all viewers as well as a clear differentiation from sections.

I was surprised at the clear vulnerabilities in a military network. It was exposed to be exploited hundreds of thousands of times varying from file access, root user access and many more intrusions. Another surprising result is the volume of successful logins, almost fifteen percent of log-in attempts were successful meaning that password guessing and a brute for obtaining was a viable option on this system. This shows a lack of password strength and is a key highlighted issue. Finally, I was shocked that there was no direct link between files accessed and files created compared to the duration of connection (Fig 4 & 5). Logically the longer you have access to a system, the more files you can access but this was not a trend that has been noted. For the time period of 1999 there was a mass number of cyberattacks for a time in today's world is assumed to be technologically disadvantaged.

Figure 4.

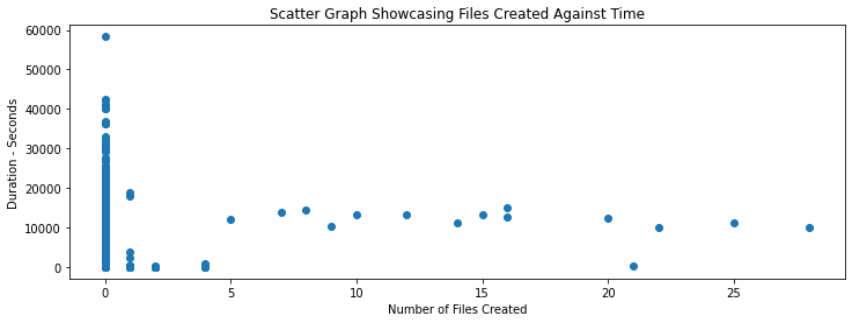
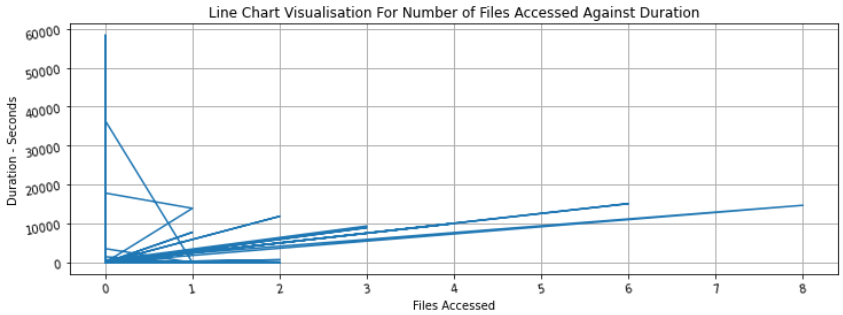


Figure 5.



To conclude, my task was successfully executed in a professional manner with key issues being addressed and highlighted throughout. I believe my findings to be clear, accurate and are of easy viewing and understanding for the end-user. I believe the assignment to be an overall successful representation of the KDD 1999 dataset's clear vulnerabilities showcased via text-based and visual representation, and something further research analysis can be built upon.

References

Oriyano, S. P. (2016). Certified Ethical Hacker Version 9 . In S.-P. Oriyano. Sybex.

Pratt, M. (2018, February 19). Retrieved March 08, 2022, from CSOnline: https://www.csoonline.com/article/3255632/what-is-an-intrusion-detection-system-how-an-ids-spots-threats.html

University of California, CA (1999). The KDD Cup 1999 Data. Irvine, CA: University of California, Department of Information and Computer Science. [Accessed 1 February 20220]. Available from: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html