VA Project – Visual Network Analysis

Alessandro Giannetti Gianluca Tasciotti

In this paper, we are going to illustrate a possible visualisation of a dataset in which are present a huge quantity of attacks. We will see some techniques to represent it and which problems we crossed to show it in the best way.

1 Introduction

Nowadays, everyone uses a personal computer to different reasons: searching for information on the web, working at home or using some services provided by the internet.
However, we should be careful when we surf on the net because unfortunately there are some dangerous websites that are able to steal precious information about the user or denied a service sending a huge quantity of requests.
These services, that have been offered, should be protected in such a way as to avoid economic hardship for those affected by the attack.
More in details, we are going to thresh out the way in which the act to carry out the attack in order to prevent these attacks.

2 Dataset selected

The field that we have been chosen to analyse is the cyber-attacks, in order to find a common attribute to prevent attacks in our computers. In this website (https://www.unb.ca/cic/datasets/ids-2017.html), we found our dataset which is in .csv extension and it is based on the HTTP, HTTPS, FTP, SSH, and email protocols.

As we know, it is divided in days and the data capturing started on the 3rd of June 2017 at 9:00 (Monday) and finished on the 7th of June 2017 at 17:00; in total, we have 5 days of observation in which it is possible to consult some details like:

- Flow ID;
- Source IP;
- Source Port;
- Destination IP;
- Destination Port;
- Protocol;
- Timestamp;
- …[1]

[1] Flow Duration, Total Fwd Packets, Total Backward Packets, Total Length of Fwd Packets, Total Length of Bwd Packets, Fwd Packet Length Max, Fwd Packet Length Min, Fwd Packet Length Mean, Fwd Packet Length Std, Bwd Packet Length Max, Bwd Packet Length Min, Bwd Packet Length Mean, Bwd Packet Length Std, Flow Bytes/s, Flow Packets/s, Flow IAT

So, each tuple inside the CSV file describes an event of the communication and information between the two parts. In general, on these days' internet traffic is seen.

However, on Monday (3rd of June 2017) no attacks were recorded, yet on the other days the following attacks were registered: **Brute Force FTP**, **Brute Force SSH**, **DoS**, **Heartbleed**, **Web Attack**, **Infiltration**, **Botnet** and **DDoS**. The attacks were carried out on Tuesday, Wednesday, Thursday and Friday in the morning and afternoon.
A possible tuple has the following pattern:

```
192.168.10.5-104.16.207.165-54865-443-6,104.16.207.165,443,192.168.10.5,54865,6,7/7/2017
3:30,3,2,0,12,0,6,6,6,0,0,0,0,0,4000000,666666.6667,3,0,3,3,3,3,0,3,3,0,0,0,0,0,0,0,0,40,0,66666
6.6667,0,6,6,6,0,0,0,0,0,0,1,0,0,0,0,9,6,0,40,0,0,0,0,0,0,2,12,0,0,33,-
1,1,20,0,0,0,0,0,0,0,0,BENIGN

172.16.0.1-192.168.10.50-53266-843-6,172.16.0.1,53266,192.168.10.50,843,6,7/7/2017
2:54,36794774,2,2,0,12,0,0,0,0,6,6,6,0,0.326133271,0.10871109,1.23E+07,2.12E+07,3.68E+07,19,3.68E+
07,3.68E+07,0,3.68E+07,3.68E+07,3.68E+07,3.68E+07,0,3.68E+07,3.68E+07,0,0,0,0,80,40,0.054355545,0.
054355545,0,6,2.4,3.286335345,10.8,0,0,0,1,0,0,0,0,1,3,0,6,80,0,0,0,0,0,0,2,0,2,12,29200,0,0,40,19
,0,19,19,3.68E+07,0,3.68E+07,3.68E+07,PortScan
```

Where the first parameter represents the **Flow ID**, the second the **Source IP**, the third the **Source Port** and so on…

Flow ID	192.168.10.5-104.16.207.165-54865-443-6	172.16.0.1-192.168.10.50-53266-843-6
Source IP	104.16.207.165	192.168.10.50
Source Port	443	843
Destination IP	192.168.10.5	172.16.0.1
Destination Port	54865	53266
Protocol	6	6
Timestamp	7/7/2017 3:30	7/7/2017 2:54
…	…	…

3 Problems and Solution proposed

Given the fact that we have a report for each day means that we have a lot of information about the communication. Therefore, it is impossible to register everything. In order to visualize this data, we had to filter it out to reduce the number of tuples (this was possible thanks to some scripts written in Python).

Mean, Flow IAT Std, Flow IAT Max, Flow IAT Min, Fwd IAT Total, Fwd IAT Mean, Fwd IAT Std, Fwd IAT Max, Fwd IAT Min, Bwd IAT Total, Bwd IAT Mean, Bwd IAT Std, Bwd IAT Max, Bwd IAT Min,Fwd PSH Flags, Bwd PSH Flags, Fwd URG Flags, Bwd URG Flags, Fwd Header Length, Bwd Header Length,Fwd Packets/s, Bwd Packets/s, Min Packet Length, Max Packet Length, Packet Length Mean, Packet Length Std, Packet Length Variance,FIN Flag Count, SYN Flag Count, RST Flag Count, PSH Flag Count, ACK Flag Count, URG Flag Count, CWE Flag Count, ECE Flag Count, Down/Up Ratio, Average Packet Size, Avg Fwd Segment Size, Avg Bwd Segment Size, Fwd Header Length, Fwd Avg Bytes/Bulk, Fwd Avg Packets/Bulk, Fwd Avg Bulk Rate, Bwd Avg Bytes/Bulk, Bwd Avg Packets/Bulk, Bwd Avg Bulk Rate, Subflow Fwd Packets, Subflow Fwd Bytes, Subflow Bwd Packets, Subflow Bwd Bytes, Init_Win_bytes_forward, Init_Win_bytes_backward, act_data_pkt_fwd, min_seg_size_forward, Active Mean, Active Std, Active Max, Active Min,Idle Mean, Idle Std, Idle Max, Idle Min, Label.

First, to reduce the cardinality of instances we selected only 7 of the most significant out of 85 parameters, which are:

- Source IP;
- Destination Port;
- Destination IP;
- Total Fwd Packets;
- Total Length of Fwd Packets;
- Timestamp;
- Label.

However, even if we used only a few parameters, we noticed that it took too much time to process the data. Therefore, we decided to remove all the samples with the **Label** equal to **BENIGN** so that we could focus on the attacks in order to prevent them from happening. In this way, we reduced the size of our dataset but in terms of the information, we had lost the percentage of benign against the attacks.

Third, while we were trying to show the resulted dataset, it was impossible to observe the high number of links between a couple of addresses. Also, in this case with Python, we were able to "select" some characteristics, that had something in common with other specs. We worked on the **Destination Port**, **Source IP** and **Destination IP** and we selected the most frequently used because there are a lot of parameters that are used rarely and increases the noise of the dataset.
Finally, we obtained a good compromise where we reduced the size of the dataset maintaining the correct proportion of information.
Example:


```
"Source": "172.16.0.1",
"DestinationPort": "80",
"Target": "192.168.10.50",
"Timestamp": "7/7/2017 15:58",
"TotalFwdPackets": "3",
"TotalLengthOfFwdPackets": "26",
"Label": "DDoS"
```


As we can see, we have changed the pattern and format of the dataset: from CSV to JSON. The reason is given the fact that to use D3.js (we will discuss it in the next chapter) and it is required to use a file in JSON format: we were interested in some features of the dataset so we decided to select only the most relevant and we were able to do that thanks to a script written in Python. Finally, by a function of D3.js, we can catch all the instances of the dataset created.

Source	172.16.0.1
Destination Port	80
Target	192.168.10.50
Timestamp	7/7/2017 15:58
Total Fwd Packets	3
Total Length of Fwd Packets	26
Label	DDoS

4 Technologies

To realize this project, we must mention two technologies that have been used:

- Python;
- JavaScript;
- D3.js.

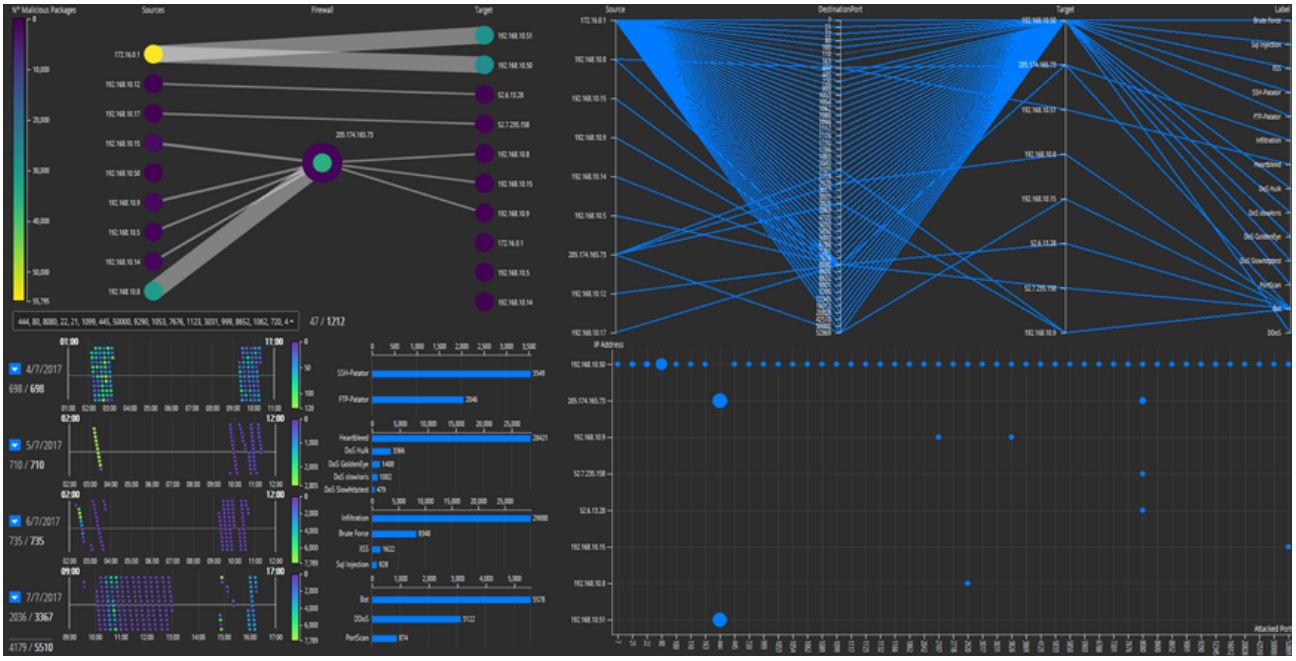
Python is a general-purpose programming language which can be used for a wide variety of applications. A great language for beginners because of its readability and other structural elements designed to make it easy to understand, Python is not limited to basic usage. In fact, it powers some of the world's most complex applications and website.
Python is an interpreted language, meaning that programs written in Python do not need to be compiled in advance in order to run, making it easy to test small snippets of code and making code written in Python easier to move between platforms. Since Python is most operating systems in common use, Python is a universal language found in a variety of different applications[2].
With Python has been possible to modify and work on the dataset since we have seen previously, filtering it in order to show the dataset created.

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.
In addition, we used D3.js, which is a JavaScript library for manipulating documents based on data.

D3 helps you bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation[3].
Instead, thanks to this tool, we were able to illustrate our data using some techniques which we are going to discuss sooner.

5 Visualizations

[2] https://opensource.com/resources/python
[3] https://d3js.org/



To demonstrate in an efficient way our dataset, we have decided to use some nodes to indicate the IP addresses: so, we created a graph where the links are the communication between two nodes.
In the same time, we can have an overview, like a picture, of what is happening time by time, day by day. In order to have a representation of each parameter, we have built a PCA, which will be discussed later.

We have four checkboxes, through them we can select the day of the attacks and for each of them, we have placed a slider to choose also the range hour. On their left, we have presented a bar chart, indicating the number of attacks for each attack for that day.
Lastly, we can observe a scatterplot, where we want to highlight the IP address attacked and in which port.

5.1 Graph

As we stated previously, in the graph we can see the nodes where they identify an IP address. We have decided to represent the dataset in this way because personally, we thought it was the best solution: using a bipartite graph enables us to identify who are the attackers and who are the victims. In general, all the networks are represented by a union of nodes and links which we can easily pinpoint a source and a target in the communication; the links demonstrate the dialogue between the two parties. In our case, we have drawn three layers: in the first one, we have assigned the attackers; in the second one, we noticed that some IP addresses receive packets but also distributes them, which clearly states a firewall; the third one, we have placed the target of the attacks.

These nodes assume a colour (from violet to yellow) which change depending on how many packets they send and receive. If one scroll over them they can understand that there is a clear line of communication between the nodes.
They are connected by a link, which describes the link between the two IP addresses and the results.

5.2 PCA

Given that we have seven features, we draw this plot to have an overview of the situation. We have four axes, one of them describes the following features: **Source**, **Destination Port**, **Target** and **Label**.

Each tuple in the dataset has been represented by a line, where this line intersects a specific value of these four axes.

5.3 Timestamp & Bar Chart

The dataset we have used demonstrates two elements: a slider, where it reveals the beginning of the attacks and the ending. The slider highlights the precise moment when a node in the graph has been selected, once these have been pinpointed, they are lighter than the others.
The second element is a bar chart, which describes the attacks that are present on a specific day and how many packets are sent with that **Label**.

5.4 Scatterplot

Here we have a cartesian plane where the vertical axis shows all the IP addresses that suffered an attack and on the horizontal axis the number of destination ports. The dot changes size based on how many attacks have been taken by the couple IP address-Port.

Above-mentioned we have introduced a preview of our application; in the next chapter we will explain the functionality of these elements.

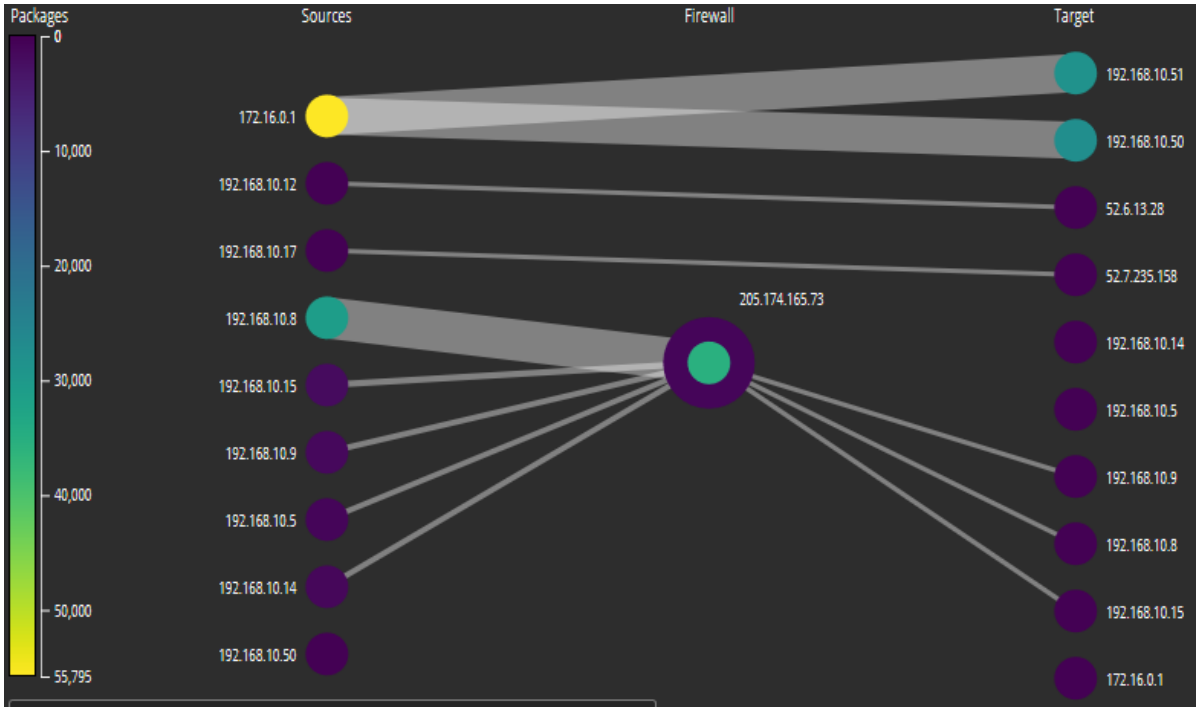
6 Functionality

The graph gives the user a visual impact of what it is happening and also gives us an idea of what the outcome will be. Once we have chosen our target the PCA gives a brief overview of the communication.

The bar chart changes to orange when indicating how many and which packets have been sent or received. Finally, in the scatterplot, we circle the selected elements to visualize them.

On the left of the graph, we can select a range of packets; on the PCA the axes can also be used to filter elements, however, if we want to filter the time or day we can use checkboxes and sliders. Although as aforementioned, when a filter is applied, all the elements change dynamically.

6.1 Graph

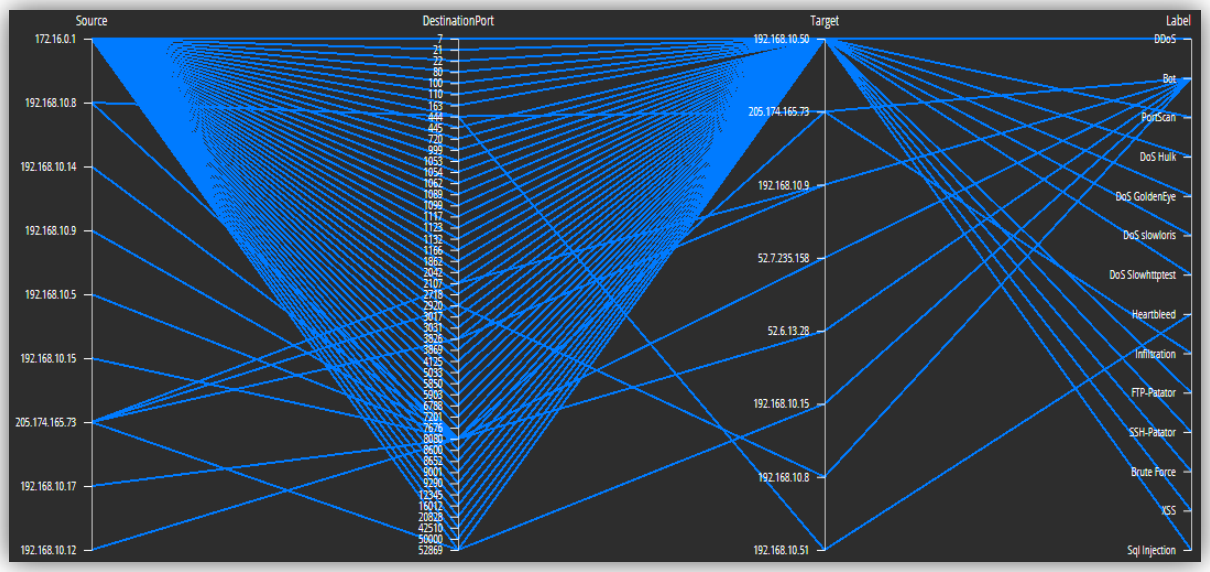


One can observe on the left a bar, which shows how many malicious packets have been sent by all the nodes. They are highlighted based on how many of these packets have been delivered.

The links represent a connection between two nodes and what's more their sizes change based on how many packets have been exchanged.

Furthermore, if we go on a node or on a link, we have a preview of the communication and in particular some details relating to them, for example IP address, forwarded packets, packets exchanged, total length of forwarded packets (in bytes), source and destination. However, if one needs more information relating to the nodes we can simply click on the node and all the information automatically changes to orange.

6.2 PCA



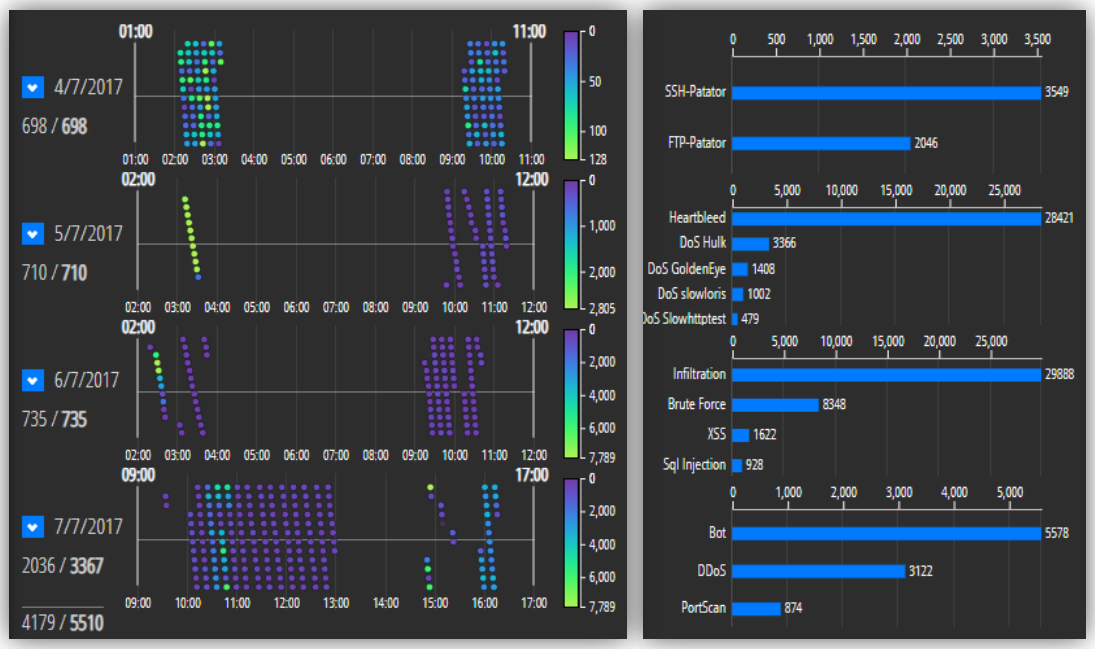
The goal of this representation is to visualize multidimensional data and show relationships between our parameters, to achieve this we used Principal Component Analysis (PCA).
Firstly, we moved from n dimensions to k dimensions and we found it useful for compressing data and using simple visualizations. What we noticed is that n is equal to 7 and k is equal to 1.

For each parameter, we created an axis with their corresponding values and each link that crosses this axis an instance of our dataset.
Therefore, in the first axis, we have the main character of the attacks, in the second which port they attacked the victim in the third the IP address of the victim and in the last axis, we can see which attack had been implemented.

To obtain further information we simply selected a range, or the particular parameter needed, the information required is then automatically highlighted.

Lastly, to be able to analyse in depth certain data and eventual relationships between them, we simply zoomed, dragged and swapped the axis.

6.3 Timestamp & Bar Chart

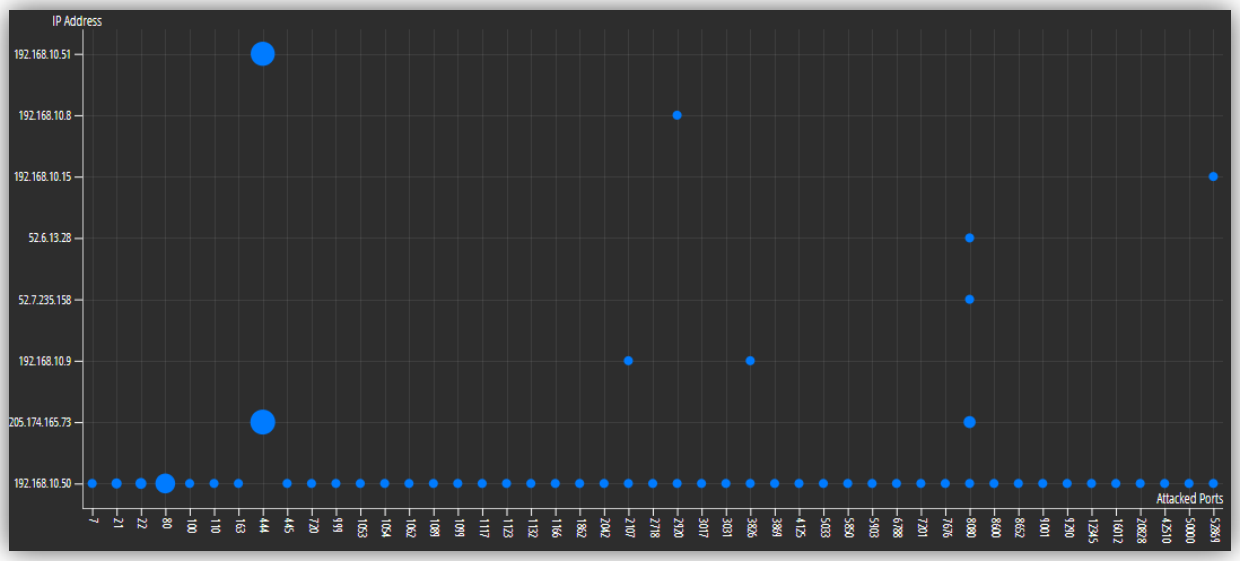


Continuing the analysis of our functionality, we decided to work on the hour of these attacks, because we had the opportunity to analyse and understand when they usually happened. We can see four checkboxes, one for each day, and on the right a bar chart plot that represents in a descending way the attacks on that day and the total number of packets sent.

We noticed a chroma scale that intuitively shows with the assistance of dots which are coloured how many and when the attacks happened. Also, we can work on the slider if we want to filter a range of hours or if we are not interested on a specific day, we can easily remove the tick on the checkbox.

Like the PCA, if we select a node on the graph, the dots will automatically stand out and the bars will be partially coloured in orange (or totally) based on how many packets the selected node has been sent. Furthermore, when we pick out the hour range, the graph and the PCA changed depending on what we had filtered, and the graph's bar changed providing on the total number of malicious packets.

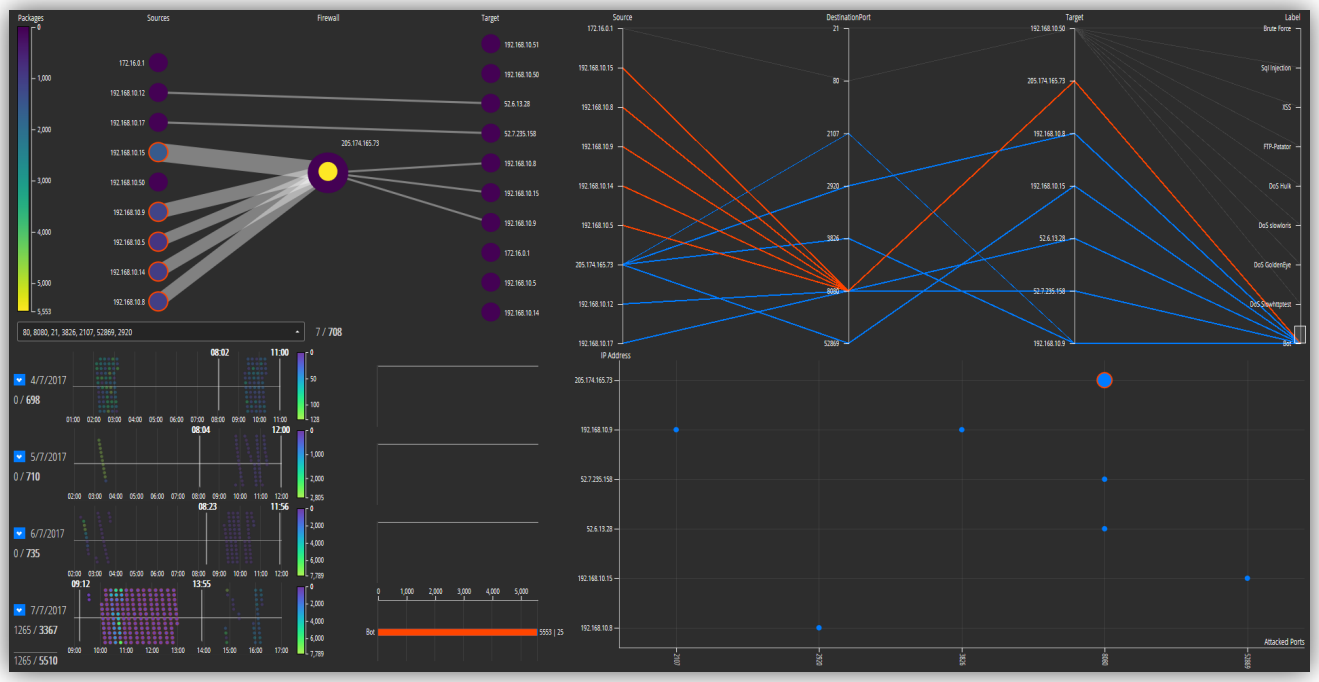
6.4 Scatterplot



On the X-axis, we can see the IP addresses while in the Y-axis the ports that we have chosen by a select (on that note, we are also able to search a certain port). It is also possible to interact with this plot by going on a dot (where we visualised that the size changes based on how many packets the couple IP address-Port received) and on this over action we visualised on the graph the communication and the timestamp slider when it happened.

7 Examples

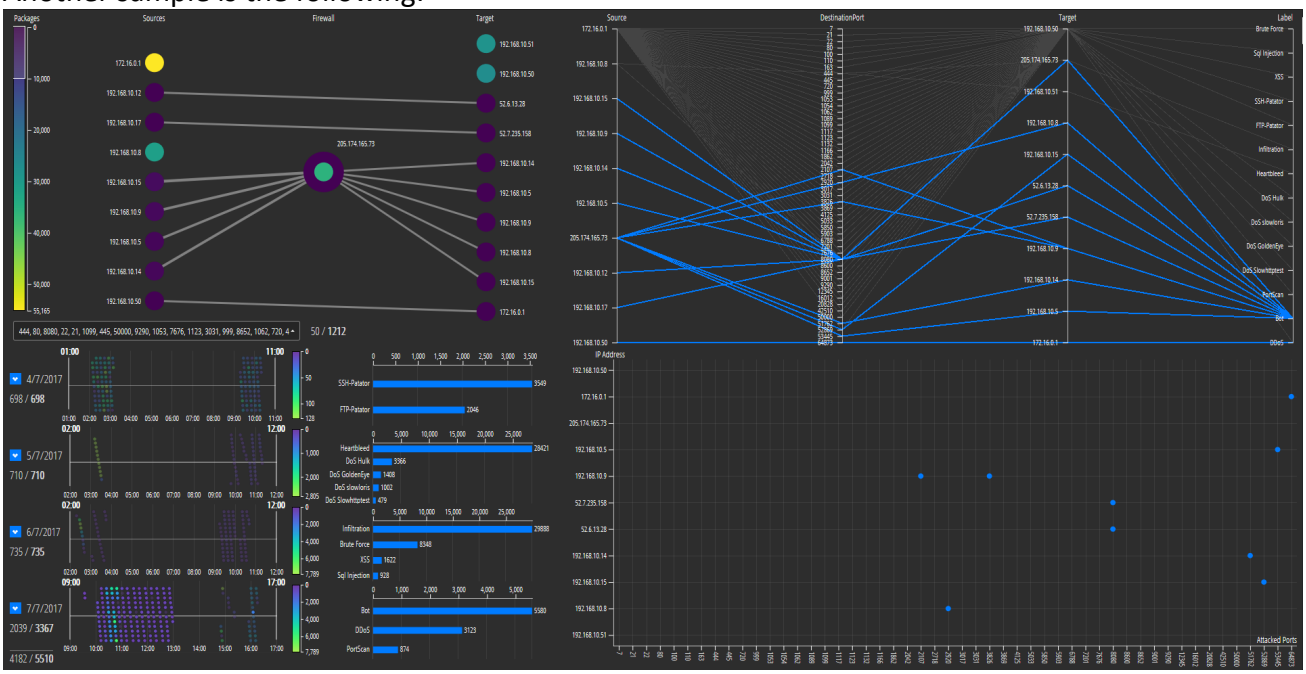
This is how our application works.



One wants to analyse a range hour of each day, so we filtered on a range that started from 8:00 to 13:00. Then, we wanted to get a focus on the nodes that attacked the firewall. We noticed that most of the attacks were labelled by **Bot**, so we selected it from the PCA in order to remove noise and we

understood that the firewall suffered this kind of attack and furthermore it was attacked by the port 8080.

Another sample is the following:



Where we decided to add three new ports (51762, 53445, 64873) on our representation, one noticed links between these nodes: 192.168.10.50 → 172.16.0.1; 205.174.165.73 → 192.168.10.14; 205.174.165.73 → 192.168.10.5, and a filter on the number of packets between 0 to 10000. We also noticed that all the instances that did not respect these filters were turned off therefore resulting that the ones that respected them were clearly highlighted.

8 Conclusion

In conclusion, we can say several things about the dataset analysed: the most common port used which received most of the packets is **444**, which is used 67% of the time and we have shown the node **205.174.165.73** as a firewall because it receives and sends messages.
172.16.0.1 is the attacker that sends the highest number of packets and consequently, the 5th is the day in which it happened but also in which were presented the highest different types of attacks.

What have we observed about these attacks?

We can clearly state that they usually appear during the night/early morning, but the **Bot** attack does not follow this rule: the Bot attack clearly stands out in the morning because it's intention is to overload the requests in a certain server (or client) in order to block the services provided.

Moreover, due to the number of dots and how it is distinctly highlighted, it is possible to note how the attacks are distributed during these days: this means the number of dots is not proportioned to the number of packets sent, simply because a single dot can send more than one packet.

We also found that if one exchanged the axis it was easier to detect further characteristics of the dataset: i.e., swapping **Label** and **Destination port** we can understand **Portscan** is the most frequent

attack that appears on each port; or i.e., **Source** and **Label** we can perceive who makes the highest kind of attacks.

9 References

- GITHUB LINK:
  https://github.com/AlessandroGiannetti/VisualAnalytics

- SLIDES:
  https://www.slideshare.net/AlessandroGiannetti3/visual-network-analysis-148780273/AlessandroGiannetti3/visual-network-analysis-148780273