Poulard Quentin

***Instructions for using the Python program***

First of all we have to choose the file we want to open to read the information inside. In the example program we open the file DumpFile.txt\* written in the line with open:

with open("DumpFile.txt", encoding="utf8") as fh:

res=fh.read()

except:

print("Le fichier n'existe pas %s", os.path.abspath('fichieratraiter.txt'))

ress=res.split('\n')

The line ress=res.split('\n) is used to remove all the \n written at the end of the line -----------------------------------------------------------------------------------------------------------------------------

Then the line file=open will create and open a file (in the example value.csv) which is the exel table where all the collected data will be written. We open the file in write mode to be able to write to it ( W ).

The event line corresponds to the name of the table columns in which the data will be written

tableau\_evenements=np.array([])

fichier=open("graphique.csv", "w")

evenement = "DATE ; SOURCE ; PORT ; DESTINATION ; FLAG ; SEQ ; ACK ; WIN ; OPTIONS ; LENGTH" #intitulé de mes colonnes

fichier.write(evenement + "\n") writing titles in the spreadsheet

-----------------------------------------------------------------------------------------------------------------------------

° We launch the program so that it goes through all the lines

for event in ress:

if event.startswith('11:42'): #

Condition that if the event line (you can call it as you want) starts with the time 11:42 the program continues to browse the lines. In our example all the lines start with 11:42, but if you want it to go through them all, put another variable that is always present in your lines

seq = ""

heure1 = ""

nomip = ""

port = ""

flag = ""

ack = ""

win = ""

options = ""

length = ""

We declare all our variables that correspond to the value that will go in each column. Each time a row is finished, it sets the values to empty

-----------------------------------------------------------------------------------------------------------------------------

Filled the first column (the date column)

texte=event.split(" ") We separate the text of the line each time there is a space because in our file the date is written first and at the end of the date there is a space then the other data heure1=texte[0] We recover the first part that has been divided by the spaces (the first one starts at 0) and we assign it to the time variable

For the second column (the source column)

texte=event.split(" ") We separate the text of the line at each space because there are source addresses that do not have dots like IP addresses (example: BP-Linux8, there is no dot but it is separated from other data with a space just after

AdrIP1=texte[2].split(".") We take the 3rd part which is separated by spaces (because there are 2 spaces to get to the source: 11:42:04.766694 IP BP-Linux8) and we also separate it by points for the IP addresses.

if len(AdrIP1) == 2:

nomip=AdrIP1[0]

if len(AdrIP1) == 3:

nomip=AdrIP1[0]+ "." +AdrIP1[1]

if len(AdrIP1) == 4:

nomip=AdrIP1[0]+ "." +AdrIP1[1]+ "." +AdrIP1[2]

if len(AdrIP1) == 5:

nomip=AdrIP1[0]+ "." +AdrIP1[1]+ "." +AdrIP1[2]+ "." +AdrIP1[3]

if len(AdrIP1) == 6:

nomip=AdrIP1[0]+ "." +AdrIP1[1]+ "." +AdrIP1[2]+ "."+AdrIP1[3]+"."+ AdrIP1[4]

-----------------------------------------------------------------------------------------------------------------------------

° For the flag column

texte=event.split("[") We separate (cut) from the first bracket in the line because the flag is written in the brackets

if len(texte) > 1:

flag1=texte[1].split("]") we take the part after the first hook and we cut at the second hook

flag=flag1[0] to have the flag it is necessary to take the part on the left of the second bracket, thus [0] (because 0 is the first part)

For the sequence column

texte=event.split(",") We separate (cut) at the comma because the sequence is written after the first comma

if len(texte) > 1: #s'il y a plus de 1 partie à partir du crochet

if texte[1].startswith(" seq"): If the text [1] ( text written after the comma , [1] because it is the second end after the separation by "," starts with "seq"

seq1=texte[1].split(" ") cut at the space and take the text just after

sequence=seq1[2] We have 2 parts separated by ":" between the split ',' and what the whole sequence number

--------------------------------------------------------------------------------------------------------------------------------------

For the ack column

if len(texte) > 2:

if texte[2].startswith(" ack"): If the text [2] (text written after the second comma (third part)) of the comma separation begins with "ack"

ack1=texte[2].split(" ") cut at the space and take the text just after

ack=ack1[2] There are 2 parts between the split ',' and what I am looking for

On some lines there is no ack , so you have to take the part [1] after the comma split

if texte[1].startswith(" ack"):

ack1=texte[1].split(" ") cut at the space and take the text just after

ack=ack1[2]

------------------------------------------------------------------------------------------------------------------------------------

For the WIN column

if len(texte) > 3: if the number of parts of the text separated is more than 3

If there is an ack, you have to take the 3rd part after the comma separation

if texte[3].startswith(" win"): If the text [3] begins with "win"

win1=texte[3].split(" ") cut at the space and take the text just after win=win1[2] We take the text after the space to have the number and we do not write WIN in the table

If there is no ack, we take the second part

if texte[2].startswith(" win"): If the text [2] begins with "win"

win1=texte[2].split(" ") cut at the space and take the text just after

win=win1[2] We take the text after the space to have the number and we do not write WIN in the table

------------------------------------------------------------------------------------------------------------------------------------

For the option column

texte=event.split("[") We separate from the " ["

if len(texte) > 2: +

options1=texte[2].split("]") We start from the separation of the bracket [ and we take the third part ( text [2] ) after the separation to arrive at the place where is written option options=options1[0] we take the first part ( left part of the 2nd ) to have the option

------------------------------------------------------------------------------------------------------------------------------------

For the lenght column

If there is the option part on the line

texte=event.split("]")

if len(texte) > 2: check the number of parts (split on the hook)

length1=texte[2].split(" ") we start from the first "[" and what we are looking for is in text [2]. We split to the space to have that the number

length=length1[2] We take the part 2 after the spaces to have the number

IF there is no option part on the line

texte=event.split(",")

if len(texte) > 3:

if texte[3].startswith(" length"): If it starts with "length"

length1=texte[3].split(" ") we cut to space

length=length1[2] We want the "2" to have only the number (text [1] before the space is the word "length").

------------------------------------------------------------------------------------------------------------------------------------

To make the program stop and write in the table

if event.startswith("11:42:55.536521") : 11:42:55.536521 is the time of the last line of the file, so if the line starts with its

prog=0 we stop the loop and browse the lines of the txt file

evenement=heure1+";"+nomip+ ";" +port+ ";" + nomip2+ ";"+flag+ ";" +sequence+ ";" +ack+ ";" +win+ ";" +options+ ";" +length

fichier.write(evenement + "\n") write "event" in the csv and \n to return to the line (not to write on the same line)

------------------------------------------------------------------------------------------------------------------------------------

To count the number of times each source address comes back

if nomip=="BP-Linux8": If the source ip variable is equal to the condition we add 1 to the variable a which will be used to make the graphs

a = a + 1

if nomip=="ns1.lan.rt":

b=b+1

if nomip=="190-0-175-100.gba.solunet.com.ar":

c=c+1

x=["www.aggloroanne.fr","ns1.lan.rt","BP-Linux8","190-0-175-100.gba.solunet.com.ar","par21s04-in-f4.1e100.net","mauves.univ-st-etienne.fr","par10s38-in-f3.1e100.net","par21s23-in-f10.1e100.net","par21s23-in-f2.1e100.net"] We put in abscissa the name of the source address

y=[e,b,a,c,d,f,g,h,i] We put in abscissa the name of the source address

fig, ax = plt.subplots(figsize=(20,10)) Choose the size of the graphic

ax.set\_yticks(np.arange(0,4000,400)) Choose the max and min value of the ordinate and the number of spaces between each marker ( 400 - 0-400-800-1200 ... )

ax.set\_title ("DDOS", color="#000000", y=1.05) title of the graphic as well as color and size

fig.autofmt\_xdate(rotation=90) inclination of the name of the abscissa

ax.bar(x, y)

fig.savefig("longueur.png", dpi=300 , bbox\_inches="tight") save the graphic in png mode to be able to recover it in the html page