# CS-6963 - Digital Forensics
# Module 9 - Programming Assignment
# Network Log Analysis

**Background**: You are working as an information security analyst at an organization when your boss frantically calls you into his office. He has just received a visit from a three letter agency, who advised your boss they have reason to believe one or more systems on your network is infected with elite malware. This malware is often used to exfiltrate data from a network. Your boss, your boss's boss, and your boss's boss's boss are all hitting the panic button.

The only information this agency was able to provide was that this particular malware is known to communicate out to command and control (C2) servers on ports 1337, 1338, 1339, & 1340. It is unknown how many systems on your internal network may be infected, as well as how many C2 servers the infected systems might be communicating with.

Your boss has already tasked the network administrators with starting to restore historical netflow data, and providing it to you for analysis. He thinks the best way to start sifting through this data is to write a script to analyze the data based on what little information we have at this time. It doesn't really matter what you think because he is your boss, so it looks like you are writing a script.

This data will be provided to you in files, with each file representing approximately 1 million connections. The network team has converted the data to CSV format in order to make it easier for you to start analyzing it. You will write a script to process one CSV at a time, and display the results to stdout. (You only have to worry about a single CSV for purposes of this assignment).

The format of the CSV files is as follows:

- Time connection established (standard epoch value)
- Source IP
- Destination IP
- Source Port
- Destination Port
- Bytes sent (source to destination)
- Bytes received (destination to source)
- Total bytes transferred

You are tasked with writing a script in Python which can take one of these CSVs as an argument, and determine the following information:

- How many systems on the internal network are infected?
- What are the IP addresses of the infected systems within the network?
- How many C2 servers are infected systems communicating with?
- What are the IP addresses of the C2 servers the infected systems are communicating with?
- What is the earliest date of the first connection to one of the C2 servers?
- What is the total amount of data (in bytes) sent from internal systems to each of the malware C2 servers?

As with previous assignments, you will be writing a script in Python which will be submitted to Gradescope, where it will be autograded. **It is extremely important you follow the criteria exactly.**

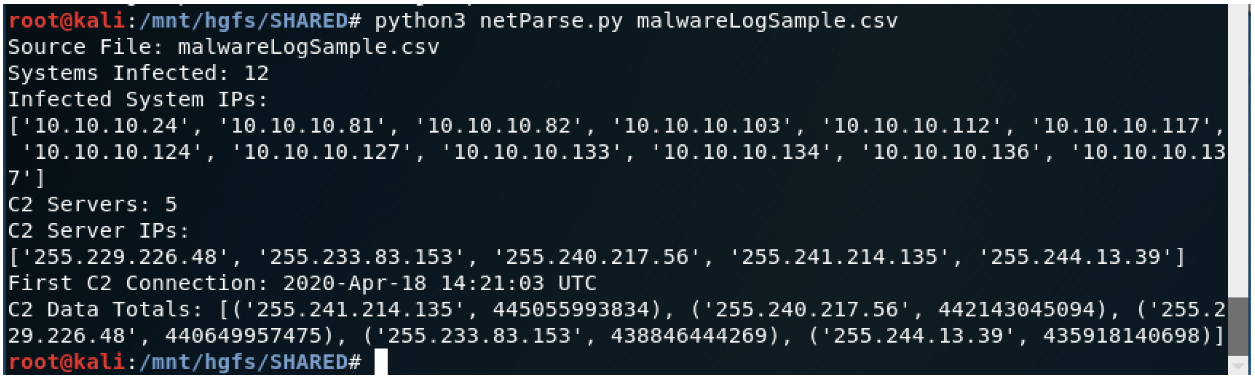
For purposes of this assignment, internal IP addresses belong to the 10.10.10.0/24 subnet. External IP addresses are fictitious, and they all begin with the first octet as 255. The remaining three octets follow standard range (0-255). This should have no bearing on your assignment.

You should consider an internal IP address with at least one connection to an external IP on one of the four ports mentioned above infected.

Requirements:

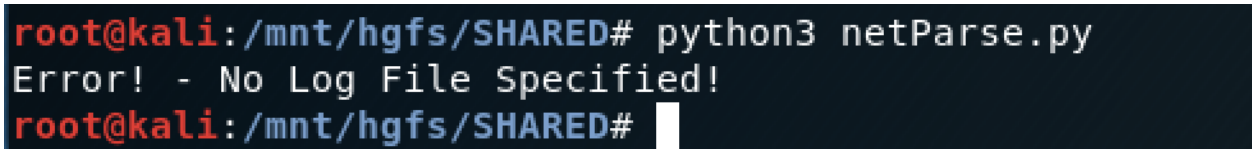
- Your script must be named ‘netParse.py’ (case sensitive)
- It must take exactly one argument, which will be a CSV file without column headers (actual data starts on the first line)
- Your script must properly handle no argument being provided
- It must also handle an argument where the file doesn’t actually exist or is unable to be opened
- Your script should first display the name of the source file it is analyzing.
- Your script must then display all the information previously listed above in the format shown in the screenshot below.
- Where a list of IPs is displayed please follow the exact format listed. **Additionally, the two lists of IP addresses must be sorted in ascending order**.
- Your listings of IP addresses should not contain duplicates.
- The last answer of C2 Data Totals must follow the exact format shown. **This list must be sorted by total bytes sent to each C2 in descending order.**
- **Your script will time out after 30 seconds**

Your script will be graded using Python3. When executed, your program should produce output like this:



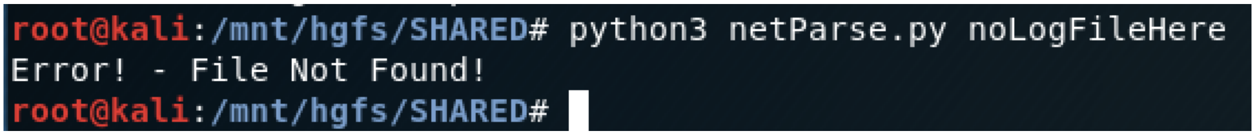
```
root@kali:/mnt/hgfs/SHARED# python3 netParse.py malwareLogSample.csv
Source File: malwareLogSample.csv
Systems Infected: 12
Infected System IPs:
['10.10.10.24', '10.10.10.81', '10.10.10.82', '10.10.10.103', '10.10.10.112', '10.10.10.117',
 '10.10.10.124', '10.10.10.127', '10.10.10.133', '10.10.10.134', '10.10.10.136', '10.10.10.13
7']
C2 Servers: 5
C2 Server IPs:
['255.229.226.48', '255.233.83.153', '255.240.217.56', '255.241.214.135', '255.244.13.39']
First C2 Connection: 2020-Apr-18 14:21:03 UTC
C2 Data Totals: [('255.241.214.135', 445055993834), ('255.240.217.56', 442143045094), ('255.2
29.226.48', 440649957475), ('255.233.83.153', 438846444269), ('255.244.13.39', 435918140698)]
root@kali:/mnt/hgfs/SHARED#
```

If no argument is supplied, your program should display the message “Error! - No Log File Specified!” to stdout as follows:



```
root@kali:/mnt/hgfs/SHARED# python3 netParse.py
Error! - No Log File Specified!
root@kali:/mnt/hgfs/SHARED# 
```

If an argument is supplied, but the program is not able to open it, it should display the message “Error! - File Not Found!” to stdout as follows:



```
root@kali:/mnt/hgfs/SHARED# python3 netParse.py noLogFileHere
Error! - File Not Found!
root@kali:/mnt/hgfs/SHARED# 
```

Each different requirement will be tested and scored independently. You are being provided with a single log file to test with. The autograder will use a different log file. Both follow the exact same format, and are roughly the same size.

Once you submit your script, it will show you the results almost immediately, and let you know how each component was scored. STDOUT & STDERR will also be displayed for any instance of a wrong answer

Like the previous programming assignments, you may re-submit your script as many times as you like up until the deadline, with the system grading and providing feedback each time. Only your last submission will count, so you are welcome to (and encouraged to) keep trying until you have a submission which scores 100.