

## C++ Data structures to understand botnet

Authors: José Luis Madrigal Sánchez, Jorge Isidro Blanco Martínez and Alan Josué Melgar Fuentes

Here we can see an infographic with the structures used, which are vectors, linked lists, trees, graphs and hash tables. During the semester, we used a csv file that contained many registers of connections made by hosts in a network, including things like date, time, domains, IP addresses and ports.



# Guerra de los bots:Ataque cibérnetico

**Algoritmos fundamentales:**

La realización de algoritmos de programación es un elemento común en la detección de patrones, resaltando tres operaciones que son el ordenamiento, búsqueda y mezcla. Además, el uso de estas técnicas en lenguajes de programación facilita diversas acciones, ya que por medio de una clasificación se pueden organizar los datos con ciertos criterios. Con el uso de estos algoritmos para ordenar nuestros datos y filtrar por fechas lo cual nos ayuda a identificar de mejor manera la fecha de los posibles ataques

**Árboles:**

En esta situación con el uso de árboles utilizamos un heap con las ip destino determinando la prioridad por su número de accesos, pueden identificar las que atacan a nuestro sistema también, además podríamos analizar cuáles computadoras han accedido desde el momento donde se identificó una botnet, porque la infección tiene que empezar en cierto tiempo. Con esto encontramos la boot master: 62.231.113.60

**Uso de listas ligadas:**

Utilizamos una lista doblemente ligada determinando el orden a partir de la ip destino para así poder clasificarlos de menor a mayor y agruparlos, también nos ayuda a tener un registro de cómo fluye la información y ver si se tiene alguna anomalía en el instante que se está utilizando.

**Grafos:**

Generamos un grafo dirigido con ip origen e ip destino donde la ip destino generaba arcos hacia los nodos de origen, ya que al hacer acciones en repetidas ocasiones es una actividad sospechosa por lo que logramos encontrar las direcciones ip con más salidas la cual fue: 172.30.142.38 nuestro primer infectado

**Códigos hash:**

Las tablas hash asocian llaves con valores, para nuestra situación utilizamos tablas hash para encontrar el número de accesos cercanos de una dirección ip en un rango de 30 segundos, utilizamos la dirección destino como llave mientras que el valor fue un par de datos: la hora y los accesos cercanos en el rango dicho anteriormente por lo que al analizar los datos de los dominios que recibieron más acceso cercanos obtenemos las posibles víctimas, forbes.com con 50 accesos cercanos fue víctima de un ataque DDOS. Con esta implementación comprobamos que la boot master es: zlpfkbbaskodzclpmll.xxx ya que tiene un grado alto de entrada por lo que las redes se conectaban a pedir instrucciones constantemente

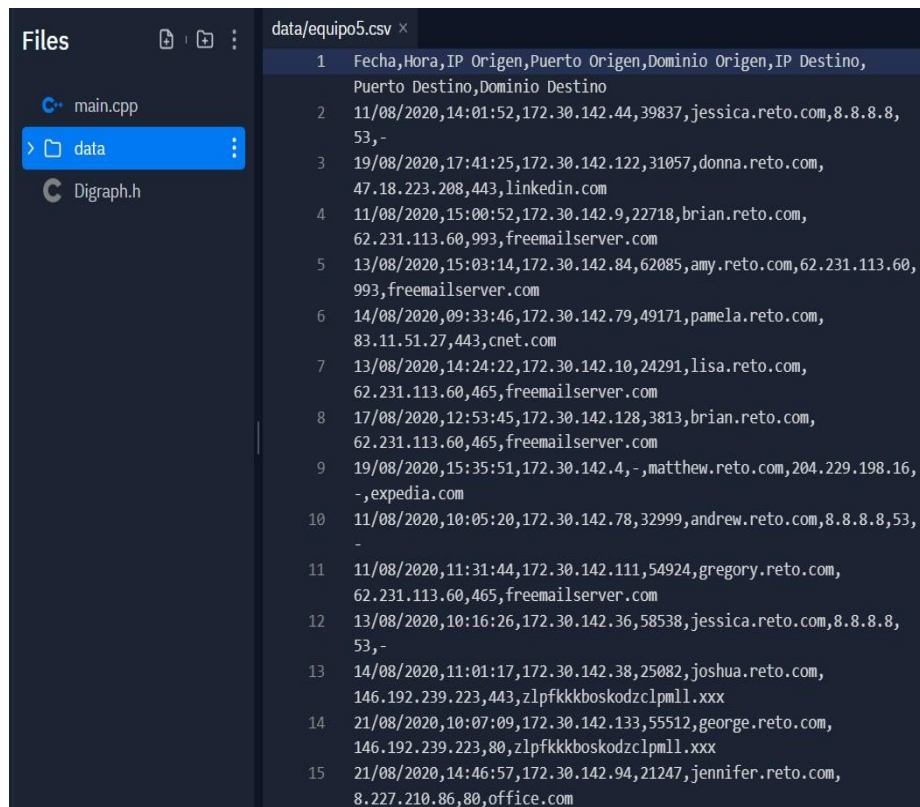
**Alan Josué Melgar Fuentes A01752228**  
**Jorge Isidro Blanco Martinez A01745907**  
**José Luis Madrigal Sánchez A01745419**

Basically, we made scripts to sort the registers by different criteria and organizing them in an specific data structure so we can find some pattern or suspicious activity.

We answered 3 basic questions:

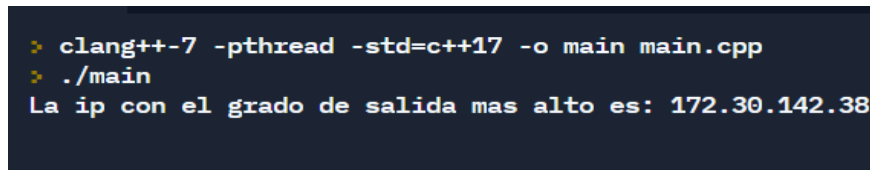
- Boot master (trees): 62.231.113.60
- First infected host (graphs): 172.30.142.38
- More attacked domain (hash tables): forbes.com

Here we can see the file that was used and was given by our teacher.



```
data/equipo5.csv
1 Fecha,Hora,IP Origen,Puerto Origen,Dominio Origen,IP Destino,
  Puerto Destino,Dominio Destino
2 11/08/2020,14:01:52,172.30.142.44,39837,jessica.reto.com,8.8.8.8,
  53,-
3 19/08/2020,17:41:25,172.30.142.122,31057,donna.reto.com,
  47.18.223.208,443,linkedin.com
4 11/08/2020,15:00:52,172.30.142.9,22718,brian.reto.com,
  62.231.113.60,993,freemailserver.com
5 13/08/2020,15:03:14,172.30.142.84,62085,amy.reto.com,62.231.113.60,
  993,freemailserver.com
6 14/08/2020,09:33:46,172.30.142.79,49171,pamela.reto.com,
  83.11.51.27,443,cnet.com
7 13/08/2020,14:24:22,172.30.142.10,24291,lisa.reto.com,
  62.231.113.60,465,freemailserver.com
8 17/08/2020,12:53:45,172.30.142.128,3813,brian.reto.com,
  62.231.113.60,465,freemailserver.com
9 19/08/2020,15:35:51,172.30.142.4,-,matthew.reto.com,204.229.198.16,
  -,expedia.com
10 11/08/2020,10:05:20,172.30.142.78,32999,andrew.reto.com,8.8.8.8,53,
  -
11 11/08/2020,11:31:44,172.30.142.111,54924,gregory.reto.com,
  62.231.113.60,465,freemailserver.com
12 13/08/2020,10:16:26,172.30.142.36,58538,jessica.reto.com,8.8.8.8,
  53,-
13 14/08/2020,11:01:17,172.30.142.38,25082,joshua.reto.com,
  146.192.239.223,443,zlpfkkkboskodzclpml1.xxx
14 21/08/2020,10:07:09,172.30.142.133,55512,george.reto.com,
  146.192.239.223,80,zlpfkkkboskodzclpml1.xxx
15 21/08/2020,14:46:57,172.30.142.94,21247,jennifer.reto.com,
  8.227.210.86,80,office.com
```

Finally, we can see the message that is generated after making all the necessary operations and filters, in this case, node (IP address) with the highest output rate, which is the first infected host.



```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
La ip con el grado de salida mas alto es: 172.30.142.38
```