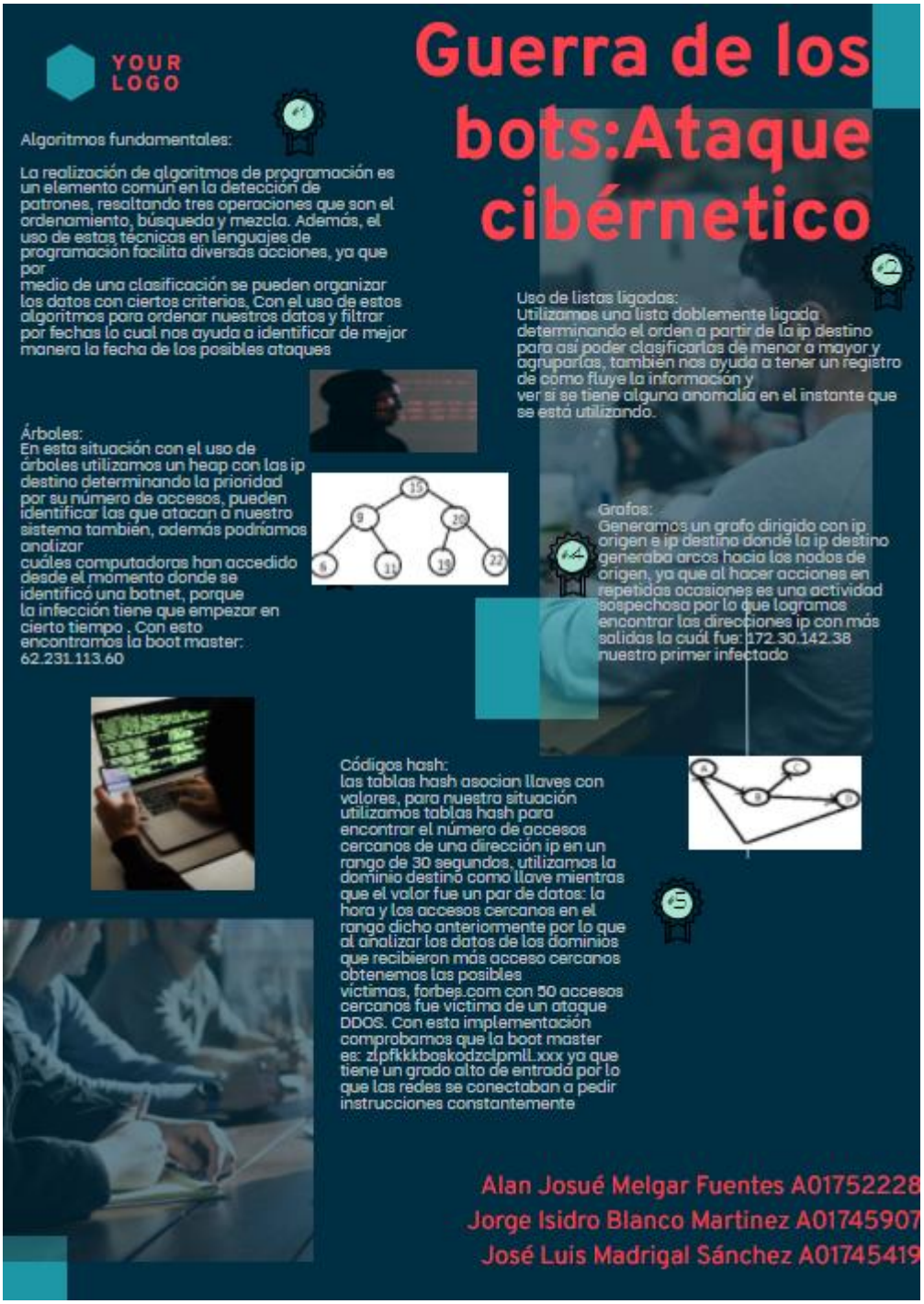
José Luis Madrigal

C++ Data structures to understand botnet

Infographic with the structures used (vectors, linked lists, trees, graphs and hash tables)



Trees gave us the boot master.

Hash Tables gave us the domain with more close accesses (equal or less than 30 seconds), which is the one who was attacked, like a DDOS.

José Luis Madrigal

Implementation of Graph (the one which I liked the most and gave us the first infected host)

File of the Digraph Class

Files

main.cpp

data

Digraph.h

Digraph.h

```cpp
#ifndef DIGRAPH_H_INCLUDED
#define DIGRAPH_H_INCLUDED

#include <iostream>
#include <vector>
#include <algorithm>
#include <string>

using std::vector;
using std::cout;
using std::endl;

/**
* DiGraph implementation using an adjacency list
**/
template <typename T>
class Digraph
{
private:
    // Vector of nodes' values
    vector<T> nodes{};

    // Vector of vectors containing each nodes' neighbors
    vector<vector<T>> adjacency{};

    // Returns the numeric index of a value
    // or -1 if value is not found
```

File of the accesses(there are 33261 lines)

Files

main.cpp

data

Digraph.h

data/equipo5.csv

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

José Luis Madrigal

Main file

main.cpp

```cpp
    return os;
}

int main() {
    // Leer archivo
    vector<string> Ip_o{};
    vector<string> Ip_d{};
    string datos = "data/resultado";
    ifstream archivo(datos);
    string linea;
    char delimitador = ' ';
    char delimitador2 = '.';
    getline(archivo, linea);
    while (getline(archivo, linea))
    {
        stringstream stream(linea);
        string Fecha, Hora, IP_Origen, Puerto_Origen,
        Dominio_Origen, IP_Destino, Puerto_Destino,
        Dominio_Destino;
        getline(stream, Fecha, delimitador);
        getline(stream, Hora, delimitador);
        getline(stream, IP_Origen, delimitador);
        getline(stream, Puerto_Origen, delimitador);
        getline(stream, Dominio_Origen, delimitador);
        getline(stream, IP_Destino, delimitador);
        getline(stream, Puerto_Destino, delimitador);
        getline(stream, Dominio_Destino, delimitador);

        Ip_o.push_back(IP_Origen);
        Ip_d.push_back(IP_Destino);
        }
    Digraph<string> dg{};
```

Running the code (IP adress node with the highest output grade)

```
> 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
```