

PRÁCTICA 6 – RECIBIR DATOS

FECHA: 25/10/21

NOMBRE DEL EQUIPO: EL SIUUU TEAM

PARTICIPANTES: -FISCHER SALAZAR CÉSAR EDUARDO

-LÓPEZ GARCÍA JOSÉ EDUARDO

-MEZA VARGAS BRANDON DAVID

UNIDAD ACADÉMICA: REDES DE COMPUTADORAS

PANORÁMICA

INTRODUCCION

Se ha venido empleando sockets crudos y el manual de Linux para crear códigos que nos ayuden a observar tramas y elementos de red de nuestro ordenador con ayuda de la consola y el Wireshark. La realización de la práctica 6 permitirá esta vez que se haga la realización correcta de recibir datos, observar dichos elementos que han llegado a la trama, y generar tráfico de internet para que se puedan captar datos de diferentes sitios de internet.

En esta ocasión, se retoman elementos que emplearon de la práctica anterior y configurar nuestro programa para que nuestra consola pueda captar las tramas que llegan a nuestro ordenador y poder realizar un análisis de esa información que ha llegado

OBJETIVOS

OBJETIVO PRINCIPAL: PROGRAMAR SOCKETS CRUDOS PARA RECIBIR UNA TRAMA APLICANDO FILTROS PARA RECIBIR TRAMAS CON NUESTRA MAC Y QUE SEAN SOLICITUDES ARP

OBJETIVO SECUNDARIO. INTERPRETAR LAS TRAMAS QUE RECIBIMOS CON EL PROGRAMA

RECURSOS NECESARIOS PARA REALIZAR LA PRÁCTICA

Manuales man socket , man inet_atom, man 2 bind, man 7 ip

Compilador de c

Terminal de Linux

Navegador de internet

PARTE 1: DIAGRAMA DE FLUJO

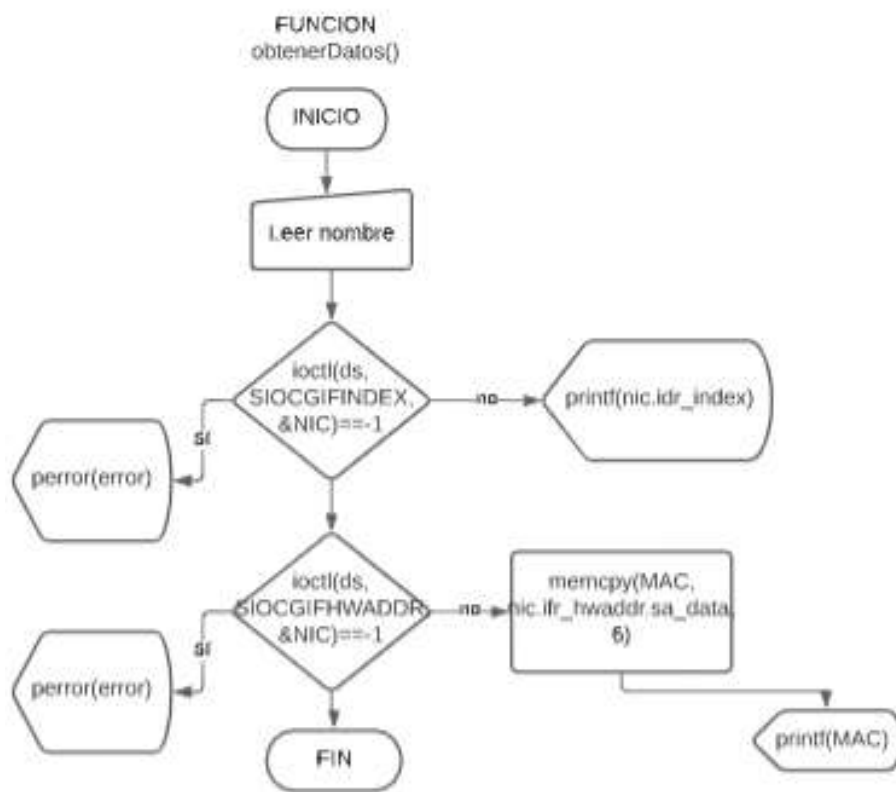


Imagen 1. Diagrama de flujo de la función obtenerDatos().

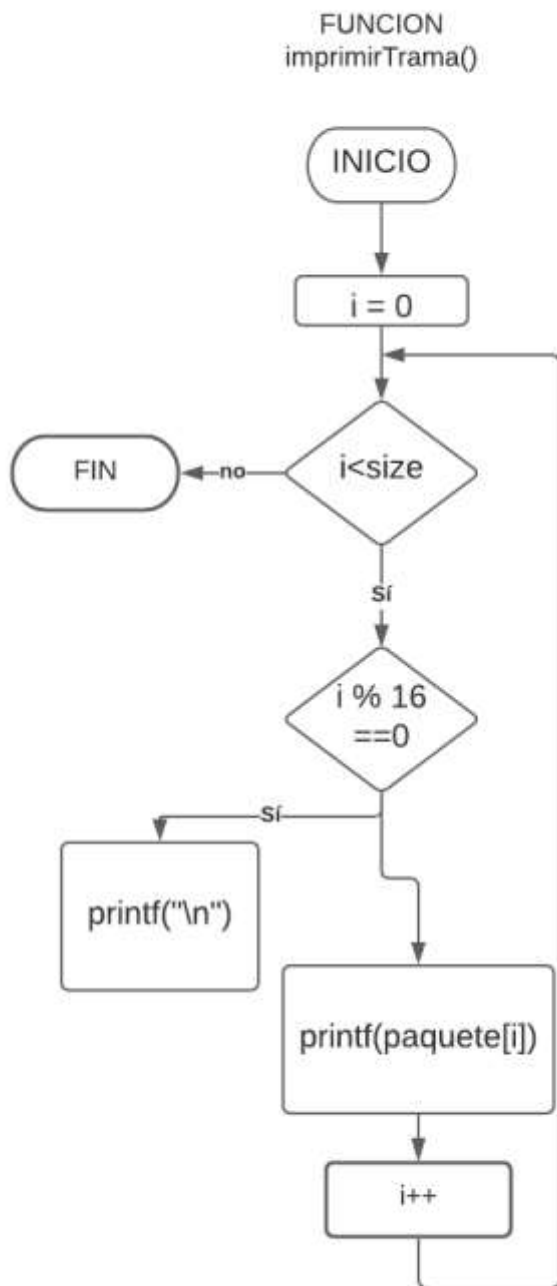


Imagen 2. Diagrama de flujo de la función imprimirTrama()

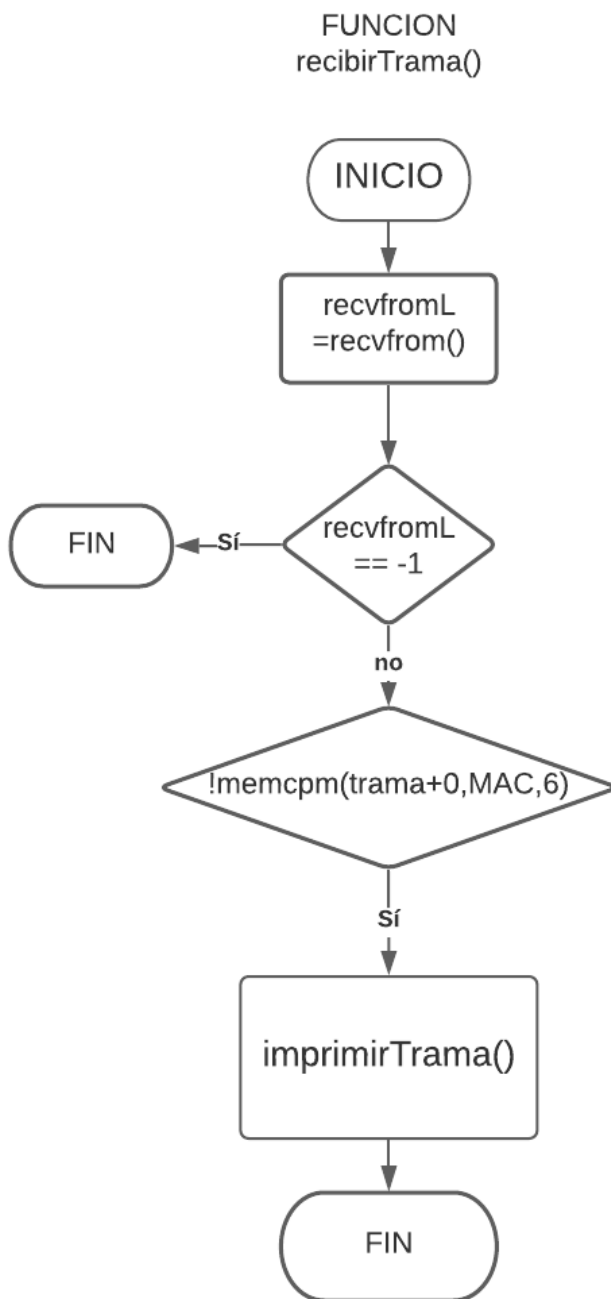


Imagen 3. Diagrama de flujo de la función recibirTrama().

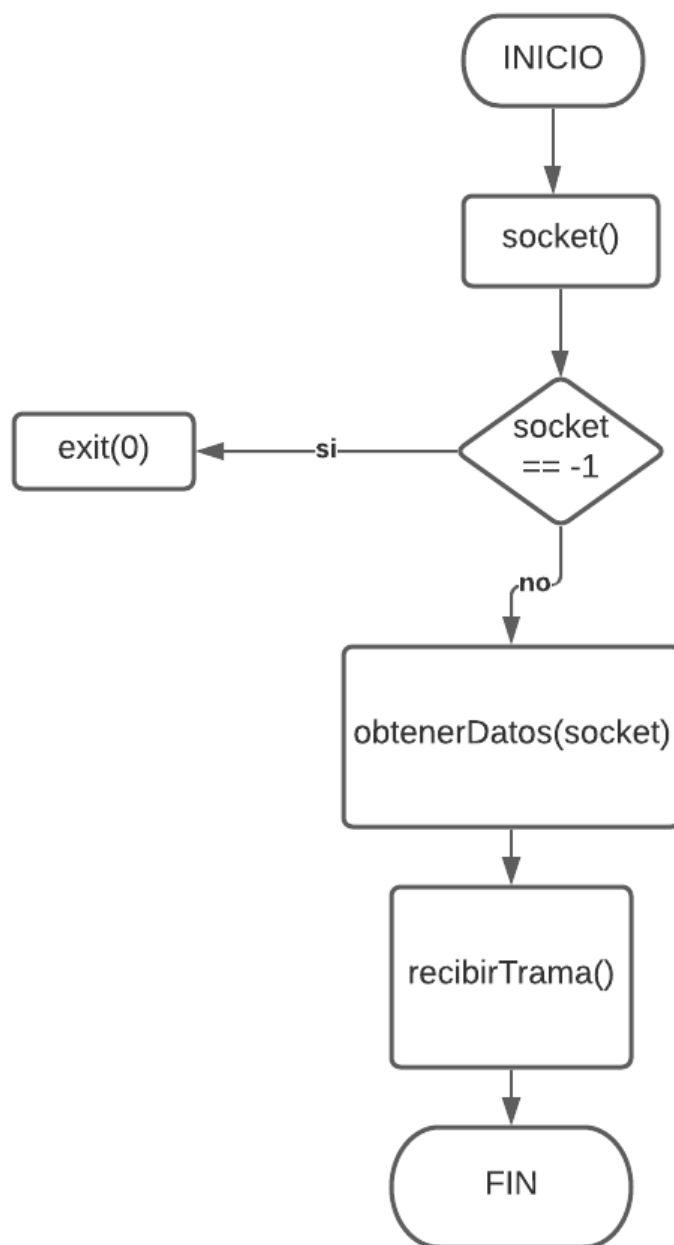


Imagen 4. Diagrama de flujo del main.

PARTE 2: CÓDIGOS, COMANDOS Y EJECUCIÓN Y EXPLICACIÓN.

2.1 INCLUIR CODIGO EXPLICANDO LAS ESTRUCTURAS DEL PROGRAMA, Y FUNCIONES USADAS Y MENCIONAR DE QUE MANUAL DE LINUX CONSULTARON, CAMBIAR NOMBRE DE SUS VARIABLES Y ESTRUCTURAS DE FORMA PERSONAL. RECUERDEN QUE LAS MEJORAS QUE LE HAGAN AL PROGRAMA VISTO EN CLASE AUMENTA SU CALIFICACION.

```
#include <sys/socket.h>
#include <linux/if_packet.h>
#include <linux/if_ether.h>
#include <net/ethernet.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/ioctl.h>
#include <net/if.h>
#include <string.h>

unsigned char originMAC[6]; //para la MAC origen
unsigned char recvTram[1514]; //Para la trama a recibir será del mismo tamaño de 1524

/*
Esta función de obtener datos recibe el descriptor del socket y nos devuelve
el índice de nuestra interfaz de red.
Esta función se encarga de obtener la MAC de nuestra máquina e imprimirla
para identificarla en las tramas que recibiremos.
De igual forma se encarga de regresar el índice de nuestra interfaz de red a partir del nombre de
nuestra interfaz.
*/
int obtenerDatos (int socketDesc)
{
    struct ifreq nic; //estructura ifreq para consultar datos de nuestra interfaz, man 7 ip
    char interName[20]; //variable que almacenara el nombre de nuestra interfaz
    int i, interIndex; //variable para loop y para el índice

    printf("\nInserta el nombre de la interfaz de red: ");
    scanf("%s", interName); //leemos el nombre de la interfaz de red

    //almacenamos el nombre de la interfaz en nic.ifr_name
    strcpy(nic.ifr_name, interName);

    /*
    La función ioctl la usamos para manipular los valores de los parámetros de un socket.
    Proporciona una interfaz para controlar el comportamiento de dispositivos y de sus descriptores,
    en este caso de sockets
    */
    // obtener el índice usando SIOCGIFINDEX
    if(ioctl(socketDesc, SIOCGIFINDEX, &nic) == -1){
        perror("\n Error al obtener el índice de red"); //error si no obtenemos el índice
        exit(0); // salimos del programa
    }
    else{
        interIndex = nic.ifr_ifindex; //almacenamos el índice en nuestra variable
        printf("\n El índice es: %d", interIndex); //imprimimos nuestro índice de red
    }
}
```

```
//obtener la MAC usando SIOCGIFHWADDR
if(ioctl(socketDesc,SIOCGIFHWADDR,&nic)==-1){
    perror("\nError al obtener la MAC"); //error si no obtiene la MAC
    exit(0); //salimos del programa
}
else{
    //si obtenemos la mac la almacenamos en nuestra variable global originMAC,
    memcpy(originMAC, nic.ifr_hwaddr.sa_data,6);

    printf("\n La MAC es: ");
    //realizamos un for hasta 6 para imprimir nuestra MAC, usamos el formato hexadecimal usando %.2x
    for(i=0;i<6;i++)
    {
        printf("%.2x: ", originMAC[i]);
    }
    printf("\n"); //salto de linea para leer mejor la salida del programa
}
}

return interIndex; //Retornamos el indice
}

/*
Esta funcion es para imprimir la trama recibida.
Recibe el paquete (la trama) recibida y el tamaño de esta
*/
void imprimirTrama(unsigned char *packet, int size){

    int i; //variable para loop

    /*
    El siguiente for es para imprimir la trama, vemos que va desde i hasta el tamaño
    de la trama recibida
    */
}
```

```
for(i=0; i<size; i++){  
    /*  
    EL siguiente if es para darle formato parecido al de wireshark, una vez se hayan impreso 16 caracteres  
    salta a la siguiente línea y continua con la impresión de los dígitos  
    */  
    if(i%16 == 0)  
        printf("\n"); //salto de línea para hacer el formato de 16 en 16  
  
    printf("%.2x ", packet[i]); //imprimimos los caracteres de dos en dos en formato hexadecimal  
}  
  
//saltos de línea para mejorar legibilidad de la salida  
printf("\n");  
printf("\n");  
}  
  
/*  
Esta función recibe el socketDesc que es el descriptor de socket y la trama recibida.  
Esta función se encarga de recibir tramas de la red usando recvfrom  
*/  
void recibirTrama(int socketDesc, unsigned char *trama){  
    int recvfromL; //para la función que recibe la trama  
  
    /*  
    En este caso la estructura será la tarjeta de red desde la cual recibiremos  
    colocamos NULL ya que no nos importará desde cual tarjeta vamos a recibir, igual  
    debido a esto colocamos un 0 al final.  
    Usamos un ciclo infinito para recibir tramas en loop.  
    */  
}
```



```
while(1){
    recvfromL = recvfrom(socketDesc, trama, 1514, 0, NULL, 0);
    if(recvfromL == -1){ //si hubo error al recibir regresa -1
        perror("\nError al recibir"); //imprimimos el error
        exit(0); //salimos del programa
    }else{
        /*
            En el siguiente if hacemos una comparación para solo capturar tramas
            que tengan mi MAC y evitar otro tráfico.
            Usamos la función memcmp y si en la trama aparece mi MAC se imprime
        */
        if(!memcmp(trama+0, originMAC, 6)){
            imprimirTrama(trama, recvfromL); //imprimimos la trama recibida
            break; //usamos este break para que solo imprima una trama
        }
    }
}

}

int main(){

    int rawSocket, index; //variable para el socket crudo y el indice de la interfaz

    /*
        primer parametro: familia a trabajar
        segundo parametro: SOCK_RAW: para trabajar en la capa de enlace de datos
        tercer parametro: colocamos todos los protocolos
        socket Devuelve un valor entero
    */
    rawSocket = socket(AF_PACKET, SOCK_RAW, htons(ETH_P_ALL));

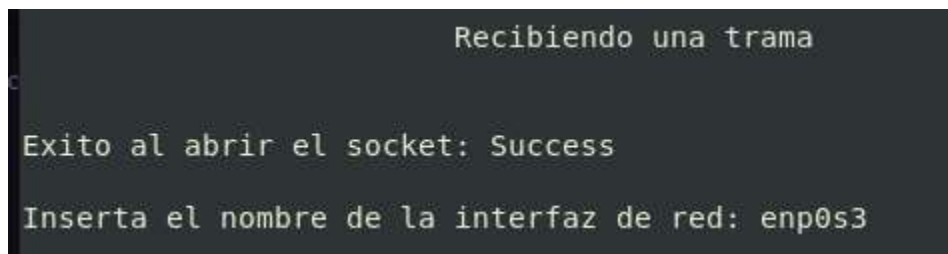
    if(rawSocket == -1)
    {
        perror("\nError al abrir el socket"); //si no se puede abrir un socket manda error
        exit(0); //salimos del programa
    }
    else
    {
        perror("\nExito al abrir el socket"); //se abrio el socket
        index=obtenerDatos(rawSocket); //Aqui obtenemos los datos
        recibirTrama(rawSocket,recvTram); //Recibimos tramas
    }

    close(rawSocket); //Cerramos el socket
    printf("\n"); //salto de lineal para leer mejor la salida
    return 0; //termina el programa
}
```

Imagen 5. Código explicado

De este programa se hicieron algunas mejoras, entre ellas el formato, pues muestra un formato al igual que el programa wireshark al imprimir las tramas, además de hacer un break correcto para que solo se imprima una trama.

2.2 EJECUTAR TOMAR CAPTURA DE PANTALLA DE CADA ETAPA DEL PROGRAMA COMO LO VIMOS EN CALSE



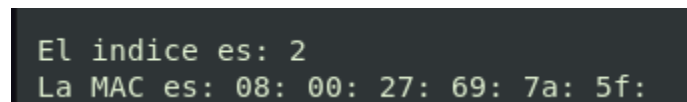
```
Recibiendo una trama

Exito al abrir el socket: Success

Inserta el nombre de la interfaz de red: enp0s3
```

Imagen 6. Primera parte de ejecución.

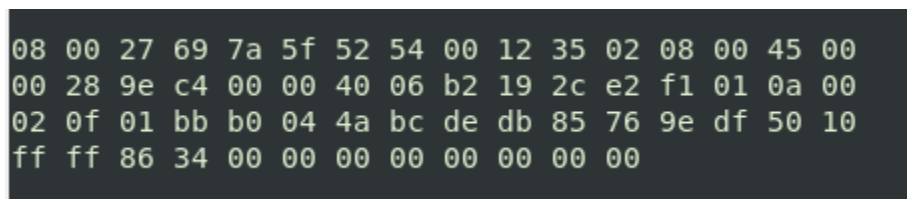
En esta primera parte el programa nos muestra que el socket se abrió correctamente y nos pide el nombre de nuestra interfaz, para así obtener la MAC como se ve en la siguiente imagen.



```
El indice es: 2
La MAC es: 08: 00: 27: 69: 7a: 5f:
```

Imagen 7. Parte dos del programa en ejecución.

Después, el programa se encarga de recibir tramas, solo nos muestra una debido al break que pusimos para terminar el ciclo infinito, de otra forma nos mostraría de manera indeterminada tramas.



```
08 00 27 69 7a 5f 52 54 00 12 35 02 08 00 45 00
00 28 9e c4 00 00 40 06 b2 19 2c e2 f1 01 0a 00
02 0f 01 bb b0 04 4a bc de db 85 76 9e df 50 10
ff ff 86 34 00 00 00 00 00 00 00 00 00
```

Imagen 8. Tramas recibidas por el programa

La vista completa de la ejecución se muestra en la siguiente imagen:

```

Recibiendo una trama

Exito al abrir el socket: Success

Inserta el nombre de la interfaz de red: enp0s3

El indice es: 2
La MAC es: 08: 00: 27: 69: 7a: 5f:

08 00 27 69 7a 5f 52 54 00 12 35 02 08 00 45 00
00 28 9e c4 00 00 40 06 b2 19 2c e2 f1 01 0a 00
02 0f 01 bb b0 04 4a bc de db 85 76 9e df 50 10
ff ff 86 34 00 00 00 00 00 00 00 00 00 00

```

Imagen 9. Ejecución completa

2.3 INCLUYE LA CAPTURA DE PANTALLA DE LAS TRAMAS Y EXPLIQUE LOS DATOS RECIBIDOS EN CADA PARTE DEL PROGRAMA

Basándonos en lo siguiente:



Imagen 10. Trama ethernet capa de enlace

Podemos identificar los siguientes valores de nuestra trama:

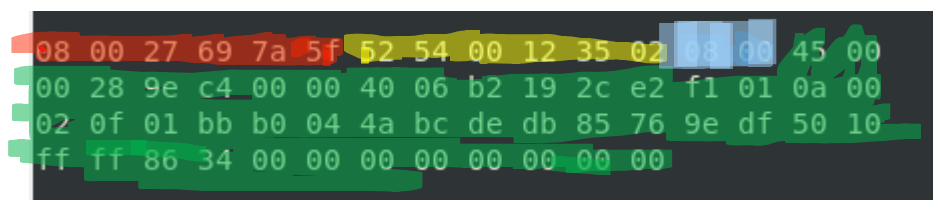


Imagen 11. Trama capturada

Dirección origen

Dirección fuente

Ethertype

Datos

Vemos que en la trama anterior no hubo bits de relleno

2.4 GENERAR TRAFICO CON SU NAVEGADOR ABRIENDO TRES PESTAÑAS, UNA DE UN YOUTUBE EN VIVO, ABRIR EL TEAMS Y OTRA CON EL SAES DE LA ESCOM CONSULTANDO SU SITUACION ESCOLAR Y REVISAR A MENOS TRES TRAMAS QUE LOGREN CAPTURAR DE ESTOS SITIOS, REPORTANDOLO CON CAPTURAS DE PANTALLA

Usando youtube:

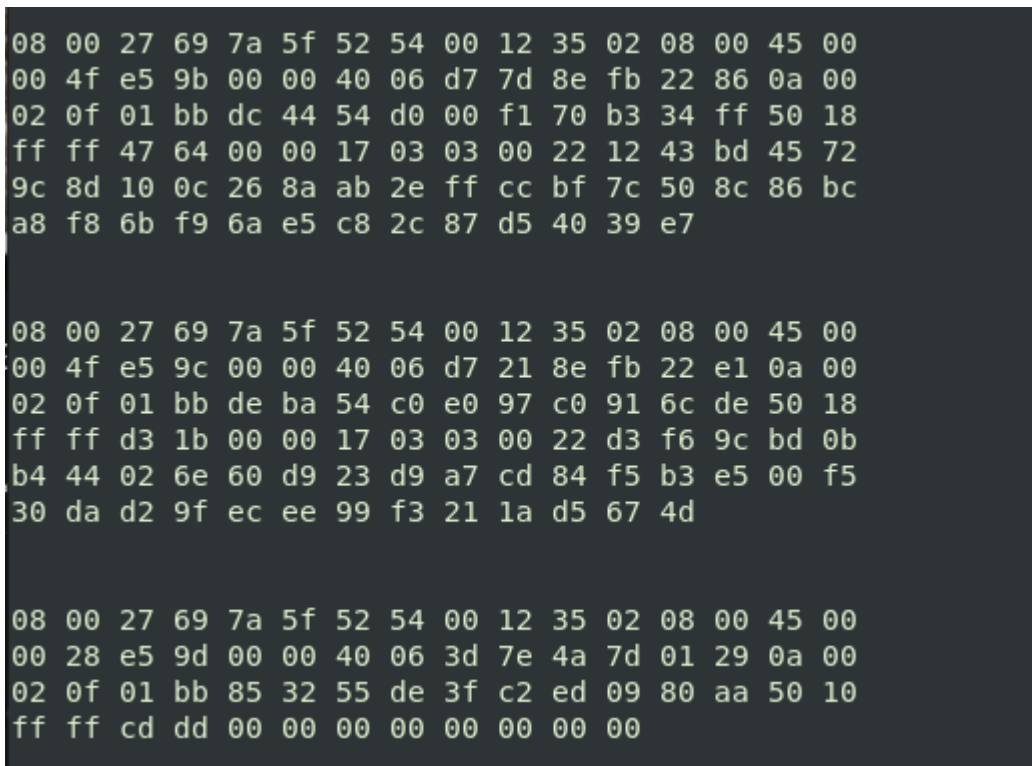


Imagen 12. Tramas usando Youtube.

Con Teams:

```
54 1a 64 13 8a 55 3e 2b e2 de 4e 92 a5 3c 85 c7
fb 7c 02 f2 2a 7d e3 2c 96 8d b5 d0 d2 0c 14 ab
b6 40 b4 04 3f e5 81 4e cb 56 1d c4 4b ce 4b 76
b2 59 8a 38 3c ca cf ee 7d e6 36 65 85 2d 1a 26
27 6a 5f 38 bb d9 45 c9 e7 bc 73 fa a6 94 f6 b0
70 f5 26 ff 0c c8 64 85 25 d3 fd 91 b4 97 9b 38
e1 6f 0b 7c 1a 97 a0 ae a6 ef 4d 53 ae f3 d9 a1
f8 16 23 f2 00 7e 31 9b 35 45 0f cb e8 08 3f 42
7a 89 f2 ef 7e 48 21 62 12 f4 91 9e 3b 2e 97 0f
f7 b8 c7 f5 4d ab b8 81 df 2c e3 17 be 1b 0e 5e
da 5a a0 4e 95 80 d2 f0 2f c5 73 a8 a9 83 08 39
25 95 c3 8a ce d4 d3 ff a3 33 da f0 7f c2 7d 96
76 36 de 16 6a 09 81 38 b8 3c da 2d 35 d3 9d 2d
ae 03 2e 2a 8d 86 d8 cb c0 66 b4 f9 19 f3 ab 15
48 38 b0 ed 21 f3 00 88 b9 14 a3 f9 e8 18 d5 06
77 c0 3f 23 23 aa 7f 71 50 0a 1d 9f 18 68 f3 5b
90 61 d1 29 50 00 9d d3 9d 36 8f 21 aa 55 7e 64
7f d1 6a be 9b ed e3 c1 40 8b d3 d7 2a 94 24 8c
bb 3c ec 2c 24 3c e3 f7 03 20 6b ba 4f c8 d6 43
9b 7c c3 30 15 49 37 ea 66 25 95 86 dc 4a 61 d9
5c ea f5 6f be a9 d8 50 cb fd b9 5b 07 15 9c cb
a6 11 3b d3 67 60 8a f0 24 d4 d1 00 af 4b 5b 0a
80 5d e7 d9 b1 b6 6e 72 e1 bb 6b 0e 61 d1 8d 73
39 0d 7b bd af 52 14 d3 33 f4 ef dd 2c 8c c7 56
8d ab 69 25 77 b5 ff e2 d5 ff ec dc e5 c1 9c 62
f5 fd 29 ad 90 7f a5 5c 71 5d 21 00 4a 66 dc 9d
5e 27 09 29 2f 7c 99 8c 1c c4 3a 43 5d bd fc af
db 5f a6 09 4d 72 d0 cb 7e 5d d7 0e 89 c2 31 81
ec 0c 73 33 d2 2a cd f7 14 f8 df a6 bf cb 64 ec
58 71 58 88 3b 6c 4c 55 21 c1 fa 15 88 cc 8e 65
75 3a 27 4e 4c 4f 8f a6 2f a7 1f e8 20 9b fa 44
fc ea 0b 42 08 48 12 ab 40 9e b7 c2 d3 43 4f 73
74 09 6b 71 23 58 b0 13 16 a4 53 ce 79 64 44 68
9c f5 9d a6 89 a3 a3 8c 4d c9 20 16 39 3f 3f 02
6c 0e 84 f9 51 26 76 07 3d 15 51 dc 04 fb 63 9e
50 ce d2 0f 90 ba ac ca b9 5c 9c 64 e9 c0 ff fc
c3 8e c7 1f 33 99 f6 2c fa a2 77 f7 c7 d1 0f 67
27 ed f0 be d5 e1 25 28 a5 c6 c0 f1 28 a6 a1 3b
75 5a 12 d5 93 c9 12 55 4d 52 b6 f0 df 98 4d a4
52 43 4c 26 2e aa f1 0a b6 77 ee 69 f4 ee b2 23
f4 87 7d e5 fa d6 0e ab 17 a1 58 8c c0 e3 98 05
ac 54 3c 45 28 e2 8a 9e 33 8a c7 5c f4 1e 3d 2b
be 9d 08 41 97 90 c0 06 ee 86 83 3d fa 8a a6 05
0b a7 7c 86 e0 09 18 7e 5c 85 b6 7b a9 42 18 cf
63 cc 3e 8f 6b 8f e1 1e aa ac 10 c2 af d4 f6 e5
26 51 09 f1 fe d3 91 66 60 b6 eb 50 bd 70 ca c1
22 ce 24 ef e6 07 ea f0 b9 71 da 3a fb 28 91 ed
01 44 1b 97 aa c2 4d 30 b1 f3 68 6b aa 7a e0 92
77 fb 11 3c 39 37 2b 12 f9 92 91 f1 af 13 a3 ff
98 8e 6a ea 42 44 de ab 76 c2 0f e0 68 7c 4d 7b
2c 46 87 fc 82 13 0b b4 4c 3f 7d 4f 65 8d ba aa
c1 22 c1 f1 aa 3b 00 95 f7 94 11 42 e4 32 24 4e
26 67 a9 3a e6 e3 bd a8 63 cb 1e ad ef 47 96 80
```

```
08 00 27 69 7a 5f 52 54 00 12 35 02 08 00 45 00
16 38 dc b2 00 00 40 06 89 af bb be 36 91 0a 00
02 0f 01 bb c1 98 59 ae 33 ce 0c cb c5 f8 50 18
ff ff 14 89 00 00 17 29 4d 4c 41 bf 6c 6d 92 13
aa 6e 22 8d 69 8b c1 07 c1 a6 2d e0 21 c9 ff e8
4f a4 26 f2 e2 bc dc 3d 76 cf ac d9 dc 05 7a 1f
a5 4c 41 ac 8c 16 56 ce 61 80 4b a1 a7 25 a7 e1
a6 96 ce 1f 36 4e 96 74 b1 7d d5 2e 5a 3c 3d a8
49 a8 79 5c ef 2d 35 ab 01 d6 df 32 6d d8 84 d2
63 d4 f5 eb 0c 1d a8 49 ae 44 a2 02 e0 21 2e 58
90 bf cc 3d dc 26 6d 64 7e e6 9a 43 c1 d8 37 67
18 de c7 e8 9c 87 34 2f 56 a3 6b d9 80 8a 5b 32
ef 59 bf 72 3e 1e d8 15 be 15 22 40 49 ee cb ec
10 0b a8 ea aa c8 1e b8 2b fa 2b 97 85 0b a4 52
69 e7 13 0a 02 46 9d 9a 09 9d f0 af 69 d5 8d a6
65 de 94 3f dd 60 a2 1c ff ce 58 93 36 6f 7b b8
7a cf 2a 39 44 46 08 00 b7 13 72 e0 f7 5c ea d6
f0 3b 10 3b 63 b6 f5 a8 df f5 c5 e3 47 4f 8b 7e
dd 96 22 86 cb f2 d2 9e 2f 60 50 24 b8 7b 2d 92
37 08 33 c5 43 32 d7 47 f9 fd 62 08 21 7c 45 df
3f f9 aa 0d 1d 16 9f f7 28 13 f6 09 f2 cf 3e 33
f3 c1 d1 c6 53 91 39 d0 3e 29 94 cf a0 50 39 96
2f 25 03 f7 fb 86 36 0c c6 43 3a ce 39 7e 61 e0
3f f5 10 16 62 89 03 7e 24 dd ef 72 c7 fe 0a 85
60 09 61 d9 e8 5e 99 99 5d 30 79 55 48 5d 24 b8
fa d7 73 e4 e1 36 91 b2 e9 2a 30 f5 70 bf 08 b0
9a 74 d5 6c 40 b0 82 74 64 67 15 38 b7 bc 2f 56
2c a6 0a 00 71 8a 89 b2 cb 76 39 9b 2b eb 22 5d
7b 5c 47 37 62 60 7b 7f ab a1 a5 5f ce 41 81 df
9c 86 41 f1 dd 98 ec 8b 88 63 93 c9 9e 9a 8f be
81 23 71 3c 5d d5 78 5c 90 62 f3 92 3a d5 81 6f
bb 16 fd 8a 3d 4b 6a f3 b0 67 30 37 d7 94 0f 1b
b2 32 08 52 db 36 98 ab 02 1b 52 ec 14 05 50 8c
b5 42 8e fd bd e8 3d d7 ca 51 c6 50 8c 74 05 47
3a 4d b3 65 16 5e b9 81 dc 47 53 73 2c 42 f9 6a
19 77 db 27 97 71 b8 2e e3 86 e3 81 53 f1 dc 81
67 22 93 5a 4b cf 12 7c 28 31 c1 9e a6 c7 fc 3e
dc 9f c1 64 10 f2 fd 2b 85 1f 6f f3 e0 85 73 83
11 d6 2f c2 d2 dc fb 6d 17 48 09 ea 22 92 19 75
1d 4d e3 8e e9 24 44 00 06 30 0d 6e 60 f4 5e a6
e2 4a ee 71 59 33 6d 4d e4 c2 66 1f a0 df 47 a1
1a ca bb 89 ac c7 dd 0b 9b 4d 4d 8f 90 75 30 f3
ab 45 20 e2 8a e9 01 06 6f 97 c5 64 1d 39 20 25
d1 93 38 5c f3 30 ee 31 53 4a 74 8d 49 b2 c8 a8
83 62 ad 52 06 31 f7 a7 9d cd 7b 2b a2 0d fa 94
cd 5b b8 02 01 d5 26 c1 01 0b ab 7b c2 5e bc 0c
24 ab d2 a5 16 73 47 9b 37 8b b6 41 a0 f7 50 f7
3a 37 54 45 94 b7 68 96 8a 1b 72 ac 03 42 26 08
b5 ab 76 9a ac a4 d8 34 6a 27 cc 13 5e 94 8b bf
48 1b 99 6a 25 dd 16 a3 7d d5 90 1a b6 24 a3 a1
```

```
08 00 27 69 7a 5f 52 54 00 12 35 02 08 00 45 00
16 38 dc b6 00 00 40 06 89 ab bb be 36 91 0a 00
02 0f 01 bb c1 98 59 ae 49 de 0c cb c5 f8 50 18
ff ff 14 89 00 00 fa 81 b5 56 80 71 b3 9d a4 ab
17 3c e0 e3 1c 42 a8 4a 75 ef 20 d9 2b 17 12 d0
17 e4 3c 99 e1 66 f2 78 f7 67 71 0c 28 c3 2c 05
b0 96 e6 89 b0 b9 0a 26 64 e7 4d 24 cc 7d 96 96
b0 17 63 8e 6c 13 3e ee 91 86 0d 2b ee 2c 3d 8d
2a 54 fa 6d 6f 9a 67 b5 26 7f 17 7d 54 a2 d6 60
be a7 14 e9 53 02 f6 8f 6d 6e 60 fe b8 09 ac f6
b8 24 31 58 44 b4 df 1d 1a 30 66 13 69 34 6b f7
74 a6 46 4d d6 21 c0 08 3a c8 76 9d 2a 4f a6 10
3f 25 d2 5b 28 8e 9e 72 18 a6 d9 5d 35 45 f2 0c
dc f6 9c 6f 99 1d 93 1a 1c 1b 82 5e fc 9b 08 5f
00 07 d7 66 85 39 b1 46 a3 a2 71 ea cd 54 4c 93
22 30 5a 1c 08 59 b2 b2 17 0c b3 9d 10 b7 f2 9e
9d 7e 52 6b 54 2c 68 be b9 d8 05 6b 2a 86 72 3a
69 72 85 7c ed 3d 1e 4b ad 6c f4 9e 20 4a 43 fc
b1 16 96 73 ee 9d a3 1a ec c4 9d 11 a3 ff ae 9f
38 05 15 3c 0d c6 36 60 82 01 a4 41 1b 81 aa b7
25 eb ec b2 f3 ea 63 2f c3 5c ea 96 73 70 d4 b7
3c 24 0c 53 59 32 28 d7 70 c0 13 12 d3 54 1e 49
99 94 2d 34 18 b6 c3 6b bc 63 3b 06 08 c4 f1 3e
34 2c 4e 44 24 24 b3 79 73 86 79 85 88 83 79 61
72 a0 31 aa 80 2e b0 df f7 b4 e3 74 34 d8 1b 06
e0 db ec d1 72 33 ec 69 e4 b8 d4 67 1f 6b c2 64
75 c9 8c 28 72 dd 57 3c 9a 0e 7d 11 cf 89 19 34
d1 65 0c ae c1 47 87 f8 a1 6e b9 dd 8c 82 6a 6f
b6 11 f9 3c 59 78 5e 9b 26 22 38 68 ff 1d 83 6e
0c ac b8 7e 94 c5 c1 09 0a 52 59 6a 87 e7 d2 18
d1 53 48 a8 85 55 c1 27 12 5f eb a7 fa 82 db 52
d3 df a4 f4 63 e9 e9 02 11 71 b9 a8 4c cb aa 31
2c ec c5 de 3e b4 76 01 7d 10 c1 14 43 3b 8f 75
f1 83 73 18 76 98 bc e7 3a 1e f5 5a 27 01 29 50
db 34 d1 69 b5 ab 1f 39 3a 7f 86 b9 a5 24 46 14
fa fd 89 cb 3b 26 4b 15 8b a7 c7 8e 39 49 4e 5b
46 76 3b d3 79 9a 60 83 e1 77 4a 7e fe e8 4d 59
ca df 96 1f 3c 12 41 81 f9 33 60 90 cb 30 67 8a
47 2b f8 47 41 34 cc 49 6d 90 a4 26 ab 24 10 c8
60 9d e4 81 9f 69 b4 f9 25 2b 6b dc b3 2b a4 b0
d7 73 90 b0 5a c3 dc 31 99 16 9b da 88 73 de 98
53 7b f5 8d a1 69 16 95 bd 6e 53 12 6e f0 94 fb
77 83 9f 62 78 f2 6a 9f b6 57 87 ff 03 eb 8d 7c
cb e9 bc f9 6d 70 5f 8e df cf 65 48 ff 09 78 ec
1b 6f b2 80 36 d0 1c ff 18 46 94 4a 09 40 81 fc
05 bf 31 ae 5b dd 41 fe d8 19 75 f6 cf 28 cc a4
c3 61 eb 70 71 10 6e 91 1b 17 46 2c c5 f3 36 e9
ee 72 bd ff 6b bb b4 77 cd 54 78 ce c3 37 0d 46
19 cc 3e c6 f2 a2 1c c4 f1 07 bd c9 07 09 9c 18
```

Imagen 13. Tramas de teams

Notamos que las tramas que recibimos al entrar a teams es demasiado grande, con esto podemos ver que los datos enviados son demasiados.

En el saes:

```

08 00 27 69 7a 5f 52 54 00 12 35 02 08 00 45 00
00 28 f5 22 00 00 40 06 ab e2 94 cc 38 f0 0a 00
02 0f 01 bb 96 1c 5b 47 5a a2 eb 39 1d fb 50 10
ff ff 7f 13 00 00 00 00 00 00 00 00 00 00 00

08 00 27 69 7a 5f 52 54 00 12 35 02 08 00 45 00
00 28 f5 23 00 00 40 06 ab e1 94 cc 38 f0 0a 00
02 0f 01 bb 96 1a 5b 46 b0 0c a3 91 d1 73 50 10
ff ff bd db 00 00 00 00 00 00 00 00 00 00 00

08 00 27 69 7a 5f 52 54 00 12 35 02 08 00 45 00
00 28 f5 24 00 00 40 06 b5 03 17 c1 ac d8 0a 00
02 0f 00 50 d4 9a 5b 19 50 f3 b7 fa e5 fb 50 10
ff ff c0 3e 00 00 00 00 00 00 00 00 00 00 00

```

Imagen 14. Tramas capturadas en el saes

2.5 MODIFIQUE EL PROGRAMA PARA CAMBIAR DE COLOR CADA PARTE DE LA TRAMA QUE RECIBIMOS.

Para este punto se hicieron funciones con los colores necesarios para pintar la trama, esto se ve en la siguiente imagen.

```

void red(){
    printf("\033[1;31m");
}
void reset(){
    printf("\033[0m");
}
void yellow(){
    printf("\033[1;33m");
}
void blue(){
    printf("\033[1;34m");
}
void green(){
    printf("\033[1;32m");
}
void white(){
    printf("\033[1;37m");
}

```

Imagen 15. Colores para el programa

De esta forma al imprimir la trama se tomó como base lo del punto 2.3, haciendo la siguiente modificación del Código en la función imprimir trama.

```
if( i < 6 ) //coloreamos de rojo la direccion origen
    red();
if( i >= 6 && i <= 11 ) //coloreamos de amarillo la direccion fuente
    yellow();
if( i >= 12 && i <= 13 ) //coloreamos de azul el ethertype
    blue();
if( i >= 14 && i <= 1500 ) //coloreamos de verde los datos de la trama
    green();
if( i >= 1501 && i <= 1514 )
    white();
```

Imagen 16. Imprimir trama con colores

DANDO COMO RESULTADO FINAL LO SIGUIENTE:

```
08 00 27 69 7a 5f 52 54 00 12 35 02 08 00 45 00
00 28 f7 13 00 00 40 06 a9 f1 94 cc 38 f0 0a 00
02 0f 01 bb 96 36 5d e0 84 22 b2 7d 54 cd 50 10
ff ff 54 ca 00 00 00 00 00 00 00 00 00 00 00 00

Direccion origen
Direccion fuente
Ethertype
Datos
Bits de relleno
```

Imagen 10. Código con trama coloreada.

Al final, se le agrego un filtro más, este fue para imprimir tramas que hagan peticiones ARP, se utilizó el programa de la practica 5 para hacer él envío de datos usando ARP y este programa de la practica presente las capture. A continuación, se muestran las capturas de esta implementación.

```
if(!memcmp(trama+12,ethertype,2)){
    imprimirTrama(trama, recvfromL); //imprimimos la trama recibida
    // break; //usamos este break para que solo imprima una trama
}
```

Imagen 11. Filtro ARP.

La variable ethertype se inicializo como:

```
unsigned char ethertype[2]={0x08,0x06}; //2 bytes para ethertype y se inicializa con ARP
```

Imagen 12. Variable ethertype.

A continuación se muestran las tramas capturadas ayudándonos del programa mencionado:

```
ff ff ff ff ff ff 08 00 27 69 7a 5f 08 06 68 6f
6c 61 20 61 6d 69 67 6f 73 0a 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Direccion origen
Direccion fuente
Ethertype
Datos
Bits de relleno
```

Imagen 13. Trama capturada con protocolo ARP

3. CONCLUSIONES INDIVIDUALES DE CADA PARTICIPANTE DEL EQUIPO

FISCHER SALAZAR CÉSAR EDUARDO

En la práctica anterior realizamos el envío de un mensaje al cual le definimos la trama, en esta seguimos trabajando con la trama, pero en lugar recibimos, observamos como el programa que ejecutamos nos permitía visualizar la información de los paquetes que iba llegando de los diferentes programas que requerían acceso de la red como Teams y nuestro navegador web, nos auxiliamos nuevamente para corroborar la información que nos mostraba nuestro programa mediante el programa de Wireshark.

Nuevamente esta práctica me sorprendió pues me permite observar ahora el cómo funciona otra parte de la comunicación que existe en las redes.

LÓPEZ GARCÍA JOSÉ EDUARDO

Por medio de esta práctica, y de la anterior, pudimos tener una idea más abierta y clara del manejo de tramas de red, en donde previamente se realizó el envío, y en esta ocasión se ha realizado la recepción.

Se vio llamativo el hecho de que pudieran llegar tramas desde cualquier sitio de internet a nuestro ordenador, y de tal manera que pudiéramos visualizar los datos que estaban llegando usando la herramienta de Wireshark y así identificar cada una de las partes que conformaban dicho paquete recibido.

Así, se dio a entender que las tramas provenientes de diversos sitios, de acuerdo a los utilizados para la demostración, contienen demasiados datos con base en el tamaño de la trama que se recibe.

MEZA VARGAS BRANDON DAVID

Con esta práctica junto con la practica 5 queda muy entendido la parte de las tramas y como es su flujo por las redes, en esta ocasión recibimos tramas.

Es muy interesante ver cómo podemos recibir tramas de todos los sitios que visitamos, considero que se debe tener practica para poder leer completamente y sobre todo entender todos los valores que nos arroja una trama, sin embargo con la realización de estas practica y las anteriores adquirimos una idea muy buena para saber identificar los valores de las tramas.

Además, gracia a esta práctica me doy una idea de cómo el programa que hemos estado utilizando llamado wireshark funciona, pues con este programa hicimos algo muy similar y de una manera sencilla, pues en realidad las funciones que usamos dentro del programa fueron las mismas con las que ya hemos estado trabajando, lo que hicimos fue darles formato a esas tramas recibidas.

Finalmente, se me hizo curioso como al entrar a ciertos sitios la trama enviada es demasiado larga, lo que me hace pensar que los datos que envía o recibe ese sitio son demasiados.