

Nombre: Brayan Ramirez Benitez

1.- Una trama IP con opciones -ICMP- Imprimir las opciones en hexadecimal

Código:

```
1
2     unsigned char IHL = (T[14]&15)*4, i;
3     printf("\n.:Cabecera IP:.\n");
4     printf("Protocolo: %d\n", T[23]);
5     if(IHL>20){
6         if(T[23] == 1){
7             printf("Opciones \n");
8             for(i=34;i<=14+IHL-1;i++)
9                 printf("0x%.2x \n", T[i]);
10        }
11    }
12
```

Trama:

```
unsigned char T[64] = {0x22, 0x22, 0x22, 0x22, 0x22, 0x22, 0x1A, 0x1A,
0x1A, 0x1A, 0x1A, 0x1A, 0x08, 0x00, 0x45, 0x14, 0x00, 0x00, 0x00, 0x01,
0x00, 0x00, 0x03, 0x01, 0x00, 0x00, 0x1a, 0x1a, 0x1a, 0x1a, 0x2b, 0x2b,
0x2b, 0x2b, 0x01, 0x02, 0x03, 0x04};
```

Captura

```
.:Cabecera Ethernet:..
MAC Destino: 22: 22: 22: 22: 22: 22
MAC Origen: 1a: 1a: 1a: 1a: 1a: 1a

.:Cabecera IP:..
Protocolo: 1
Opciones
0x01
0x02
0x03
0x04
```

2.- Una trama IP de Costo mínimo Imprimir TTL

Código:

```
unsigned char IHL = (T[14]&15)*4, i;
printf("\n.:Cabecera IP:.\n");
if(T[15]&2)
    printf("TTL = %d saltos\n", T[22]);
```

Trama:

```
unsigned char T[64] = {0x22, 0x22, 0x22, 0x22, 0x22, 0x22, 0x1A, 0x1A,
0x1A, 0x1A, 0x1A, 0x1A, 0x08, 0x00, 0x46, 0x02, 0x00, 0x00, 0x00, 0x02,
0x00, 0x00, 0x03, 0x01, 0x2d, 0x70, 0x1a, 0x1a, 0x1a, 0x1a, 0x2b, 0x2b,
0x2b, 0x2b, 0x01, 0x02, 0x03, 0x04};
```

Captura:

```
...Cabecera Ethernet:..  
MAC Destino: 22: 22: 22: 22: 22: 22  
MAC Origen: 1a: 1a: 1a: 1a: 1a: 1a  
  
...Cabecera IP:..  
TTL = 3 saltos
```

3.- Verificar el checksum de las tramas IP, en caso de que este correcto imprimir 😊 en caso de que sea incorrecto 😞 e imprimir el checksum correcto.

a) Llamar a su función checksum

Código:

```
void AnalizaIP(unsigned char *T){  
    unsigned char IHL = (T[14]&15)*4, i;  
    printf("\n...Cabecera IP:..\n");  
    VerificarChk(T, 14 + IHL - 1);  
}  
  
void VerificarChk(unsigned char *T, unsigned char tam){  
    //T[14] a T[14+IHL-1] y tam = 14 + IHL - 1  
    unsigned char i;  
    unsigned int chk = 0;  
  
    for(i = 14; i<=tam ;i+=2){  
        if(!(i == 24))  
            chk += (T[i] << 8) | T[i | 1];  
    }  
  
    chk = (~(chk&65535) | (chk >> 16))&65535;  
  
    if(chk == (T[24] << 8) | T[25])  
        printf("El Checksum 0x%.4x es correcto :)", (T[24] << 8)  
| T[25]);  
    else{  
        printf("El Checksum 0x%.4x no es correcto :( \n", (T[24]  
<< 8) | T[25]);  
        printf("El Checksum calculado es: 0x%.4x\n", chk);  
    }  
}
```

Nombre: Brayan Ramirez Benitez

Para el Checksum incorrecto

Trama:

```
unsigned char T[64] = {0x22, 0x22, 0x22, 0x22, 0x22, 0x22, 0x1A, 0x1A,
0x1A, 0x1A, 0x1A, 0x1A, 0x08, 0x00, 0x46, 0x02, 0x00, 0x00, 0x00, 0x02,
0x00, 0x00, 0x03, 0x01, 0x2d, 0x70, 0x1a, 0x1a, 0x1a, 0x1a, 0x2b, 0x2b,
0x2b, 0x2b, 0x01, 0x02, 0x03, 0x04};
```

Captura:

```
...:Cabecera Ethernet:..
MAC Destino: 22: 22: 22: 22: 22: 22
MAC Origen: 1a: 1a: 1a: 1a: 1a: 1a

...:Cabecera IP:..
El Checksum 0x0000 no es correcto :(
El Checksum calculado es: 0x2d70
```

Para el Checksum correcto

Trama:

```
unsigned char T[64] = {0x22, 0x22, 0x22, 0x22, 0x22, 0x22, 0x1A, 0x1A,
0x1A, 0x1A, 0x1A, 0x1A, 0x08, 0x00, 0x46, 0x02, 0x00, 0x00, 0x00, 0x02,
0x00, 0x00, 0x03, 0x01, 0x2d, 0x70, 0x1a, 0x1a, 0x1a, 0x1a, 0x2b,
0x2b, 0x2b, 0x2b, 0x01, 0x02, 0x03, 0x04};
```

Captura:

```
...:Cabecera Ethernet:..
MAC Destino: 22: 22: 22: 22: 22: 22
MAC Origen: 1a: 1a: 1a: 1a: 1a: 1a

...:Cabecera IP:..
El Checksum 0x2d70 es correcto :)
```

4.- Una trama UDP cuyo encapsulado IP no tenía opciones, devolver el valor del Offset en decimal

Código:

```
unsigned char IHL = (T[14]&15)*4, i;
printf("\n...:Cabecera IP:..\n");
if(IHL<=20){
    if(T[23] == 17){
        printf("Offset: %d bytes\n", (((T[20]&31)<<8)|T[21])*8);
    }
}
```

Nombre: Brayan Ramirez Benitez

Trama:

```
unsigned char T[64] = {0x22, 0x22, 0x22, 0x22, 0x22, 0x22, 0x1A, 0x1A,
0x1A, 0x1A, 0x1A, 0x1A, 0x08, 0x00, 0x45, 0x00, 0x00, 0x00, 0x00, 0x01,
0x00, 0x0A, 0x00, 0x11, 0x00, 0x00, 0x1a, 0x1a, 0x1a, 0x1a, 0x2b,
0x2b, 0x2b, 0x2b};
```

Captura:

```
...:Cabecera Ethernet:...
MAC Destino: 22: 22: 22: 22: 22: 22
MAC Origen: 1a: 1a: 1a: 1a: 1a: 1a

...:Cabecera IP:...
Offset: 80 bytes
```

Código completo:

```
1 void AnalizaIP(unsigned char *T){
2     unsigned char IHL = (T[14]&15)*4, i;
3
4     printf("\n...:Cabecera IP:... \n");
5     //Parte 1
6     printf("Protocolo: %d\n", T[23]);
7     if(IHL>20){
8         if(T[23] == 1){
9             printf("Opciones \n");
10            for(i=34;i<=14+IHL-1;i++)
11                printf("0x%.2x \n", T[i]);
12        }
13    }
14    //Parte 2
15    if(T[15]&2)
16        printf("TTL = %d saltos\n", T[22]);
17
18    //parte 3
19    VerificarChk(T, 14 + IHL - 1);
20
21    //parte 4
22    if(IHL<=20){
23        if(T[23] == 17){
24            printf("Offset: %d bytes\n",
25                ((T[20]&31)<<8) | T[21])*8);
```

Nombre: Brayan Ramirez Benitez

```
26         }
27     }
28 }
29
30 void VerificarChk(unsigned char *T, unsigned char tam){
31     //T[14] a T[14+IHL-1] y tam = 14 + IHL - 1
32     unsigned char i;
33     unsigned int chk = 0;
34
35     for(i = 14; i<=tam ;i+=2){
36         if(!(i == 24))
37             chk += (T[i] << 8) | T[i | 1];
38     }
39
40     chk = (~(chk&65535) | (chk >> 16)))&65535;
41
42     if(chk == (T[24] << 8) | T[25])
43         printf("El Checksum 0x%.4x es correcto :)", (T[24] <<
44 8) | T[25]);
45     else{
46         printf("El Checksum 0x%.4x no es correcto :( \n",
47 (T[24] << 8) | T[25]);
48         printf("El Checksum calculado es: 0x%.4x\n", chk);
49     }
50 }
```