**INSTITUTO POLITÉCNICO NACIONAL**

**ESCUELA SUPERIOR DE CÓMPUTO******

Analizador IP

***Mora Ayala José Antonio***

***Torres Carrillo Josehf Miguel Angel***

Profesora:

M. en C. NIDIA ASUNCIÓN CORTEZ DUARTE

Contenido

[Analizador IP 1](#_Toc88940602)

[Instrucciones 3](#_Toc88940603)

[Una trama IP con opciones ICMP 5](#_Toc88940604)

[Captura de pantalla 5](#_Toc88940605)

[Código del programa 5](#_Toc88940606)

[Una trama IP de costo mínimo 6](#_Toc88940607)

[Captura de pantalla 6](#_Toc88940608)

[Código del programa 6](#_Toc88940609)

[Verificar el checksum de las tramas IP 7](#_Toc88940610)

[Captura de pantalla 7](#_Toc88940611)

[Código del programa 7](#_Toc88940612)

[Una trama DUP cuyo encapsulado IP no tenía opciones 9](#_Toc88940613)

[Captura de pantalla 9](#_Toc88940614)

[Código del programa 9](#_Toc88940615)

## Instrucciones

En un archivo de word incluye los 4 ejercicios sobre el protocolo IP

* Pega el codigo correspondiente resultante
* Modifica tramas IP (crea) para que entren a las condiciones solicitadas
* Pon al menos 5 salidas (1 por cada inciso) como captura de pantalla.
* Una con checksum correcto y otra con incorrecto

Tramas Probadas

|  |
| --- |
| //Trama IP con opciones de tipo ICMP - Imprimir las opciones en hexadecimal  //MAC Destino T[0] - T[5]  unsigned char T1**[]** **=** **{**0x00**,**0x1f**,**0x45**,**0x9d**,**0x1e**,**0xa2**,**  //MAC Origen T[6] - T[11]  0x00**,**0x23**,**0x8b**,**0x46**,**0xe9**,**0xad**,**  //Tipo T[12] - T[13]  0x08**,**0x00**,**  //Version/IHL T[14]  0x4F**,**  //Tipo de servicion T[15]  0x00**,**  //Tamano total T[16] - T[17]  0x80**,**0x42**,**  //Identificador T[18] - T[19]  0x04**,**0x55**,**  //Bandera/ Desplazamiento Fragmento T[20] / T[20] - T[21]  0x34**,**0x11**,**  //Tiempo de vida T[22]  0x80**,**  //Protocolo T[23]  0x01**,**  //Checksum T[24] - T[25]  0x6b**,**0xf0**,**  //IP Origen T[26] - T[29]  0x94**,**0xcc**,**0x39**,**0xcb**,**  //Ip Destino T[30] - T[33]  0x94**,**0xcc**,**0x67**,**0x02**,**  //Opciones y relleno T[34] - T[73]  0xaa**,**0xbb**,**0xcc**,**0xdd**,**0x04**,**0x0c**,**0x00**,**0x35**,**0x00**,**  0x2e**,**0x85**,**0x7c**,**0xe2**,**0x1a**,**0x01**,**0x00**,**0x00**,**0x01**,**  0x00**,**0x00**,**0x00**,**0x00**,**0x00**,**0x00**,**0x03**,**0x77**,**0x77**,**  0x77**,**0x03**,**0x69**,**0x73**,**0x63**,**0x05**,**0x65**,**0x73**,**0x63**,**  0x6f**,**0x6d**,**0x03**,**0x29**};**  //MAC Destino T[0] - T[5]  unsigned char T2**[]** **=** **{**0x00**,**0x1f**,**0x45**,**0x9d**,**0x1e**,**0xa2**,**  //MAC Origen T[6] - T[11]  0x00**,**0x23**,**0x8b**,**0x46**,**0xe9**,**0xad**,**  //Tipo T[12] - T[13]  0x08**,**0x00**,**  //Version/IHL T[14]  0x40**,**  //Tipo de servicion T[15]  0x02**,**  //Tamano total T[16] - T[17]  0x80**,**0x42**,**  //Identificador T[18] - T[19]  0x04**,**0x55**,**  //Bandera/ Desplazamiento Fragmento  0x34**,**0x11**,**  //Tiempo de vida T[22]  0x80**,**  //Protocolo T[23]  0x01**,**  //Checksum T[24] - T[25]  0xbc**,**0xec**,**  //IP Origen T[26] - T[29]  0x94**,**0xcc**,**0x39**,**0xcb**,**  //Ip Destino T[30] - T[33]  0x94**,**0xcc**,**0x67**,**0x02**,**  //Opciones y relleno T[34] - T[73]  **};**  unsigned char T4**[]** **=** **{**0x00**,**0x1f**,**0x45**,**0x9d**,**0x1e**,**0xa2**,**  //MAC Origen T[6] - T[11]  0x00**,**0x23**,**0x8b**,**0x46**,**0xe9**,**0xad**,**  //Tipo T[12] - T[13]  0x08**,**0x00**,**  //Version/IHL T[14]  0x40**,**  //Tipo de servicion T[15]  0x02**,**  //Tamano total T[16] - T[17]  0x80**,**0x42**,**  //Identificador T[18] - T[19]  0x04**,**0x55**,**  //Bandera/ Desplazamiento Fragmento T[20] / T[20] - T[21]  0x34**,**0x11**,**  //Tiempo de vida T[22]  0x80**,**  //Protocolo T[23]  0x11**,**  //Checksum T[24] - T[25]  0xbc**,**0xec**,**  //IP Origen T[26] - T[29]  0x94**,**0xcc**,**0x39**,**0xcb**,**  //Ip Destino T[30] - T[33]  0x94**,**0xcc**,**0x67**,**0x02**,**  //Opciones y relleno T[34] - T[73]  **};** |

## Una trama IP con opciones ICMP

Imprimir las opciones en hexadecimal

### Captura de pantalla

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

### Código del programa

|  |
| --- |
| void programa1**(){**  unsigned short int tot **=** T1**[**12**]<<**8 **|** T1**[**13**];**  **if** **(**tot **==** 2048**){**  printf**(**"Trama IP \n"**);**  **if** **(**T1**[**23**]** **==** 1**)** printf**(**"TIPO ICMP \n"**);**  **else** printf**(**"TIPO OTRO\n"**);**    unsigned char IHL **=** **(**T1**[**14**]&**15**)\***4**;**  **if** **(**IHL **>** 20**){**  printf**(**"Tamano de Opciones %d\n"**,** IHL**);**    **for** **(**unsigned char i **=** 34**;** i **<=** **(**14 **+** IHL **-**1**);** i**++)**  printf**(**"0x%.2x "**,** T1**[**i**]);**  **}**  **else** printf**(**"No tiene Opciones\n"**);**  **}**  **else{**  printf**(**"No es una trama IP\n"**);**  **}**  **}** |

## Una trama IP de costo mínimo

Imprimir TTL

### Interfaz de usuario gráfica, Texto, Sitio web Descripción generada automáticamenteCaptura de pantalla

### Código del programa

|  |
| --- |
| void programa2**(){**  unsigned short int tot **=** T2**[**12**]<<**8 **|** T2**[**13**];**  **if** **(**tot **==** 2048**){**  printf**(**"Trama IP \n"**);**    **if** **(**T2**[**15**]&**2**){**    printf**(**"Es de costo Minimo\n"**);**  printf**(**"%d Saltos\n"**,** T2**[**22**]);**  **}**  **else** printf**(**"Otro\n"**);**  **}**  **else{**  printf**(**"No es una trama IP\n"**);**  **}**  **}** |

## 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. Imprimir las opciones en hexadecimal
* Llamar a su función Checksum

### Captura de pantalla

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

### Código del programa

|  |
| --- |
| void checksum**(){**  size\_t n **=** **sizeof(**T1**)/sizeof(**T1**[**0**]);**    unsigned char T3Aux**[**n**-**14**];**  **for** **(**unsigned char i **=** 0**;** i **<** n**-**14**;** i**++)**  T3Aux**[**i**]** **=** T1**[**i**+**14**];**  unsigned int suma **=** 0x00**,** acarreo **=** 0x00**;**  unsigned char i**,** tam**;**  i **=** 0x00**;**  tam **=** 0x14**;**  // Agarrando valores pares  **while** **(**tam **>** 1**)**  **{**  acarreo **=** **(((**T3Aux**[**i**]** **<<** 8**)** **&** 0xFF00**)** **|** **((**T3Aux**[**i **+** 1**])** **&** 0xFF**));**  suma **+=** acarreo**;**  **if** **((**suma **&** 0xFFFF0000**)** **>** 0**)**  **{**  suma **=** suma **&** 0xFFFF**;**  suma **+=** 1**;**  **}**  i **+=** 2**;**  tam **-=** 2**;**  **}**  suma **=** **~**suma**;**  suma **=** suma **&** 0xFFFF**;**  **if** **(**suma **==** 0**)**  **{**  printf**(**"El checksum es correcto :)\n"**);**  **}**  **else**  **{**  printf**(**"El checksum no es correcto :(\n"**);**  suma **=** 0x00**,** acarreo **=** 0x00**;**  i **=** 0x00**;**  tam **=** 0x14**;**  T3Aux**[**10**]** **=** 0**;**  T3Aux**[**11**]** **=** 0**;**  **while** **(**tam **>** 1**)**  **{**  acarreo **=** **(((**T3Aux**[**i**]** **<<** 8**)** **&** 0xFF00**)** **|** **((**T3Aux**[**i **+** 1**])** **&** 0xFF**));**  suma **+=** acarreo**;**  **if** **((**suma **&** 0xFFFF0000**)** **>** 0**)**  **{**  suma **=** suma **&** 0xFFFF**;**  suma **+=** 1**;**  **}**  i **+=** 2**;**  tam **-=** 2**;**  **}**  suma **=** **~**suma**;**  suma **=** suma **&** 0xFFFF**;**  printf**(**"El checksum correcto es: 0x%.4x\n"**,** suma**);**  **}**  **}** |

## Una trama UDP cuyo encapsulado IP no tenía opciones

Devolver valor del offset en decimal

### Texto Descripción generada automáticamenteCaptura de pantalla

### Código del programa

|  |
| --- |
| void programa4**(){**    unsigned short int tot **=** T1**[**12**]<<**8 **|** T1**[**13**];**  **if** **(**tot **==** 2048**){**  printf**(**"Trama IP \n"**);**  **if** **(**T4**[**23**]** **==** 17**)** printf**(**"TIPO UDP \n"**);**  **else** printf**(**"TIPO OTRO\n"**);**    unsigned char IHL **=** **(**T4**[**14**]&**15**)\***4**;**  **if** **(**IHL **>** 20**){**  printf**(**"Tiene Opciones %d\n"**,** IHL**);**  **}**  **else** printf**(**"No tiene Opciones\n"**);**  printf**(**"Offset: %d\n"**,** **(((**T4**[**20**]&**31**)<<**8**)** **|** T4**[**21**])\***8**);**  **}**  **else{**  printf**(**"No es una trama IP\n"**);**  **}**  **}** |