|  |  |  |
| --- | --- | --- |
| **ipn** | **INSTITUTO POLITÉCNICO NACIONAL**  **ESCUELA SUPERIOR DE CÓMPUTO** |  |

**Redes de Computadoras**

**“TCP Transmission Control Protocol”**

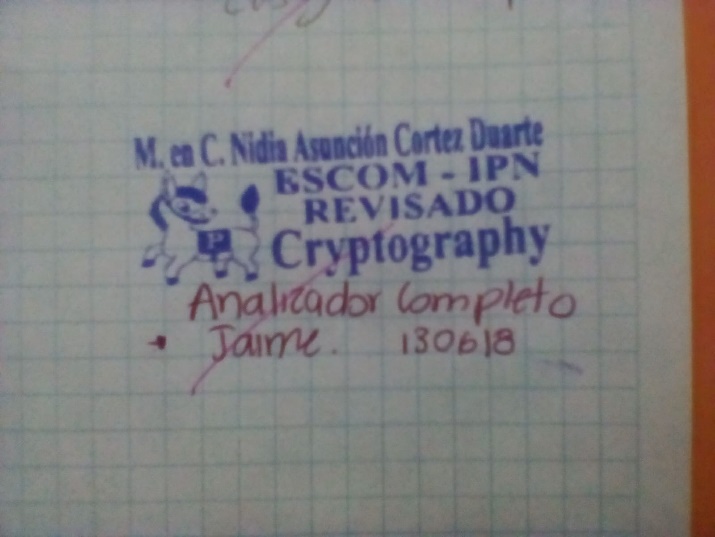
**By:**

**Bastida Prado Jaime Armando**

Professor:

M. en C. NIDIA ASUNCIÓN CORTEZ DUARTE

June 2018

****

**Index**

Contenido

[Problem: 1](#_Toc381082154)

[Hypothesis: 1](#_Toc381082155)

[Software (libraries, packages, tools): 3](#_Toc381082156)

[Procedure: 3](#_Toc381082157)

[Results (Data): 4](#_Toc381082158)

[Conclusions: 6](#_Toc381082159)

[References: 6](#_Toc381082160)

[Code 6](#_Toc381082161)

# Problem:

The problem here is to continue adding functionality to our program for analyzing frames this time TCP frames.

**TCP frames appear when the Protocol field in the IP header is equal to 6**. TCP is a transport layer protocol used by applications that require guaranteed delivery. It is a sliding window protocol that provides handling for both timeouts and retransmissions.

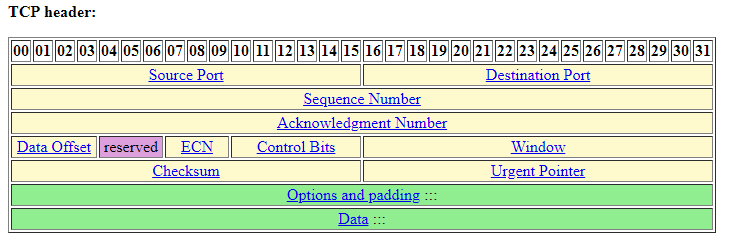
TCP establishes a full duplex virtual connection between two endpoints. Each endpoint is defined by an IP address and a TCP port number. The operation of TCP is implemented as a finite state machine.

The byte stream is transferred in segments. The window size determines the number of bytes of data that can be sent before an acknowledgement from the receiver is necessary.

The TCP header goes after the IP header and before the Data:



*Figure 1: Ethernet header*



*Figure 2: TCP header*

**Source Port.** 16 bits.

**Destination Port.** 16 bits.

**Sequence Number.** 32 bits.

The sequence number of the first data byte in this segment. If the SYN bit is set, the sequence number is the initial sequence number and the first data byte is initial sequence number + 1.

**Acknowledgment Number.** 32 bits.

If the ACK bit is set, this field contains the value of the next sequence number the sender of the segment is expecting to receive. Once a connection is established this is always sent.

**Data Offset.** 4 bits.

The number of 32-bit words in the TCP header. This indicates where the data begins. The length of the TCP header is always a multiple of 32 bits.

**Reserved.** 3 bits.

Must be cleared to zero.

**Flags (CEUAPRSF).** 8 bits.

ECN bits (used when ECN employed; else 00)

* CWR (1 = sender has cut congestion window in half)
* ECN-Echo (1 = receiver cuts congestion window in half)

U (1 = consult urgent pointer, notify server application of urgent data)

A (1 = consult acknowledgement field)

P (1 = Push data)

R (1 = Reset connection)

S (1 = Synchronize sequence numbers)

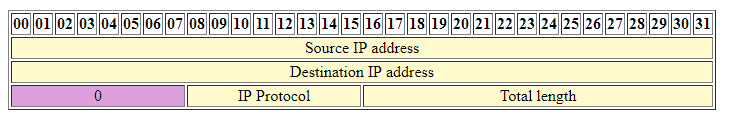
F (1 = no more data; Finish connection)

**Window.** 16 bits, unsigned.

The number of data bytes beginning with the one indicated in the acknowledgment field which the sender of this segment is willing to accept.

**Checksum.** 16 bits.

This is computed as the 16-bit one's complement of the one's complement sum of a pseudo header of information from the IP header, the TCP header, and the data, padded as needed with zero bytes at the end to make a multiple of two bytes. The pseudo header contains the following fields:



*Figure 3: Checksum pseudo-header*

**Urgent Pointer.** 16 bits, unsigned.

If the URG bit is set, this field points to the sequence number of the last byte in a sequence of urgent data.

**Options.** 16 bits, unsigned.

Options occupy space at the end of the TCP header. All options are included in the checksum. An option may begin on any byte boundary. The TCP header must be padded with zeros to make the header length a multiple of 32 bits. There are several but the ones most used are:

0 = End of **Options** list 3 = Window scale

1 = No operation (pad) 4 = Selective ACK ok

2 = Maximum segment size 5 = Timestamp

**Data.** Variable length.

# Hypothesis:

The hypothesis is that we can develop the program to analyze the TCP frames by consulting the information given by the teacher and other information given in the Web.

# Software (libraries, packages, tools):

The equipment used:

-Laptop

-Pencil

-Sheets

-The program was codified in C language using text editor Sublime Text 3

-Compiler: GNU (GCC) compiler.

# Procedure:

After we have analyzed the information in the IP header we now proceed to analyze the TCP header. As we stated before if the Protocol field is 6 in the IP header then we are dealing with a TCP frame:

# 

*Figure 4: TCP case*

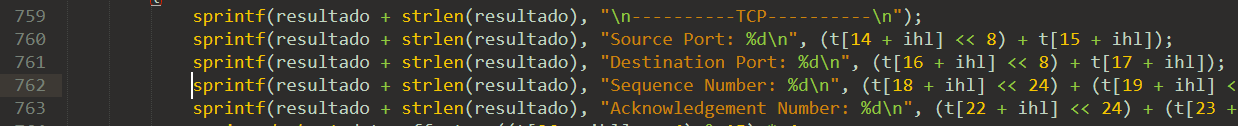
Now, first thing we are going to do is to obtain **Source and Destination Ports** along with **Sequence Number** and **Acknowledgement Number**. **Source and Destination** are 2 bytes long contained into t[14 + ihl], t[15 + ihl], t[16 + ihl] and t[17 + ihl], Why these numbers are being summed?, you may ask well, this is because ihl is the length of the IP header that comes before TCP and we have already stored that value into the variable “ihl”, “14” because that’s the length of the Ethernet header, talking in bytes of course. So, when we write 14 + ihl, we are referring to the first byte of whichever header that follows Ethernet and IP headers.

Having that said, to print **Source Port** for example, we do the following operation: “(t[14 + ihl] << 8) + t[15 + ihl]” what is happening here is that **Source Port** is 16 bits long and we have stored only 8 bits by each position of the array “t” so, to print 16 bits we have to do a left shift to t[14 + ihl] and then add t[15 + ihl], the operation looks like this:

Suppose t[14 + ihl] = 10101010 and t[15 + ihl] = 11110000

* t[14 + ihl] << 8 = **10101010**00000000
* (t[14 + ihl] << 8) + t[15 + ihl] = 10101010**11110000**

The same principle applies with the rest of the fields printed in this section, note that **Sequence and Acknowledgement Number** are 32 bits long so, left shifts are incremented.



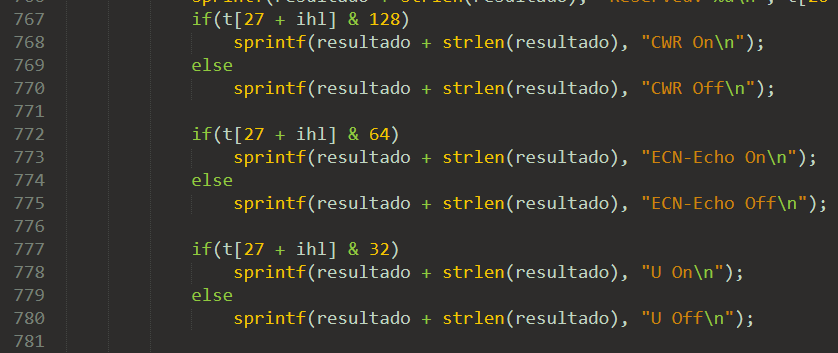
*Figure 5: Obtaining Source and Destination Ports along with Sequence and Acknowledgement Numbers*

We are going to save **Data Offset** into a variable because is going to be a value that will be checked several times. You can think of **Data Offset** as if it was **IHL** from the IP header, but in TCP header:



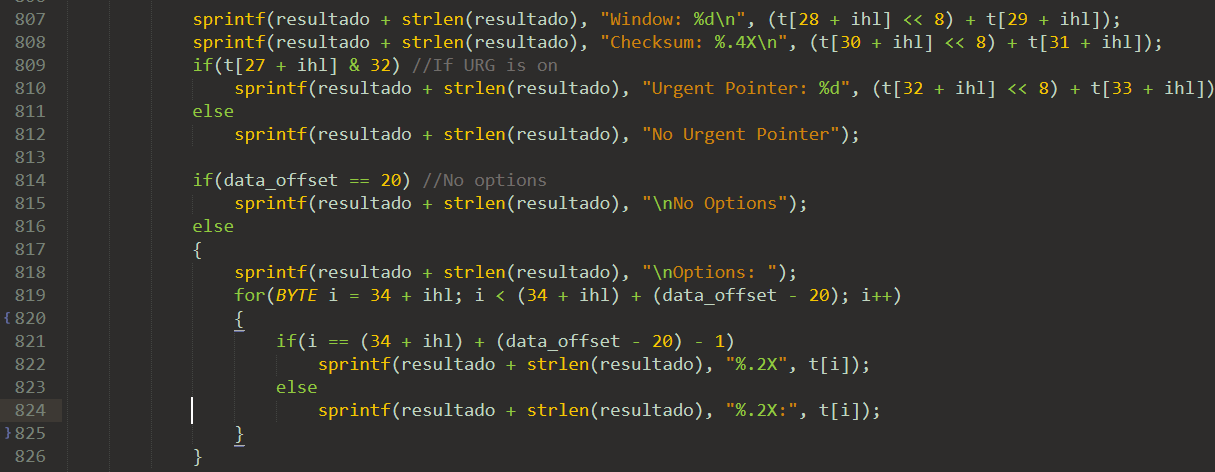
*Figure 6: Storing Data Offset into a variable*

Now, the next if-else statements will check for the **Flags** contained always in the byte 27 + ihl:



*Figure 7: Checking the Flags*

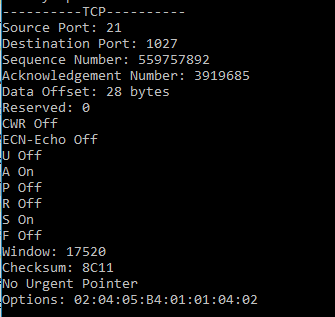
Finally, print the **Window, Checksum** (It is just printed, not calculated), **Urgent Pointer** (If the URG flag is On) and **Options** (if **Data Offset** is not 20 bytes).



*Figure 8: Obtaining the rest of the header*

# Results (Data):

Here an example using one of the 20 frames the teacher gave us:



*Figure 9: Result showing in console a TCP frame*

# Conclusions:

The hypothesis discussed earlier was accepted since we could develop the program only with the information given by the teacher.

TCP protocol seems to be the most used option when trying to send secure information through the Internet.

# References:

K.N. King, C Programming A Modern Approach Second Edition. W.W. Norton, 2008.

Network Sorcery. (2012). TCP, Transmission Control Protocol. Retrieved from <http://www.networksorcery.com/enp/protocol/tcp.htm>

# Code

1. #include <stdio.h>
2. #include <string.h>
3. #include <stdlib.h>
5. **typedef** unsigned **char** **BYTE**;
7. **int** analizaTrama(**BYTE** t[], **BYTE** resultado[]);
9. **int** main(**void**)
10. {
11. //LLC frames
12. /\*BYTE t[ ][32] = { {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x03,0xf0,0xf0, 0x7f,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 1
14. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x03,0xf0,0xf1, 0x73,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 2
16. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf0, 0x01,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00},//Trama 3
18. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1, 0x01,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 4
20. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x12,0xf0,0xf0, 0x00,0x01,0x0e,0x00,0xff,0xef,0x19,0x8f,0xbc,0x05,0x7f,0x00,0x23,0x00,0x7f,0x23}, //Trama 5
22. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x12,0xf0,0xf0, 0x00,0x03,0x0e,0x00,0xff,0xef,0x19,0x8f,0xbc,0x05,0x7f,0x00,0x23,0x00,0x7f,0x23}, //Trama 6
24. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1, 0x01,0x03,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 7
26. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1, 0x01,0x03,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 8
28. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0xac,0xf0,0xf0, 0x02,0x02,0x0e,0x00,0xff,0xef,0x16,0x04,0x00,0x00,0x00,0x00,0x28,0x00,0x7f,0x23}, //Trama 9
30. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1, 0x01,0x04,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 10
32. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x5f,0xf0,0xf0, 0x02,0x04,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x7f,0x23}, //Trama 11
34. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf1, 0x01,0x04,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 12
36. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x91,0xf0,0xf0, 0x04,0x04,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x7f,0x23}, //Trama 13
38. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1, 0x01,0x06,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 14
40. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x46,0xf0,0xf0, 0x04,0x06,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x7f,0x23}, //Trama 15
42. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf1, 0x01,0x06,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 16
44. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x7e,0xf0,0xf0, 0x06,0x06,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x7f,0x23}, //Trama 17
46. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1, 0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 18
48. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf1, 0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 19
50. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x7e,0xf0,0xf0, 0x08,0x08,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x7f,0x23}, //Trama 20
52. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1, 0x01,0x0a,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 21
54. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf1, 0x01,0x0a,0x0e,0x00,0xff,0xef,0x19,0x8f,0xbc,0x05,0x7f,0x00,0x23,0x00,0x7f,0x23}, //Trama 22
56. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x12,0xf0,0xf0, 0x0a,0x0b,0x0e,0x00,0xff,0xef,0x14,0x00,0x00,0x00,0x28,0x00,0x00,0x00,0x7f,0x23}, //Trama 23
58. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1, 0x01,0x0d,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 24
60. {0x03,0x00,0x00,0x00,0x00,0x01,0x00,0x04,0xac,0x44,0x4d,0x02,0x00,0x8b,0xf0,0xf0, 0x03,0x2c,0x00,0xff,0xef,0x08,0x00,0x00,0x00,0x00,0x00,0x28,0x00,0x42,0x34,0x20}, //Trama 25
62. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x35,0xf0,0xf0, 0x0c,0x0a,0x0e,0x00,0xff,0xef,0x16,0x04,0x00,0x00,0x00,0x00,0x28,0x00,0x7f,0x23}, //Trama 26
64. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x35,0xf0,0xf0, 0x0a,0x0e,0x0e,0x00,0xff,0xef,0x16,0x04,0x00,0x00,0x00,0x00,0x28,0x00,0x7f,0x23}, //Trama 27
66. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x12,0xf0,0xf0, 0x0e,0x0d,0x0e,0x00,0xff,0xef,0x14,0x00,0x00,0x00,0x28,0x00,0x00,0x00,0x7f,0x23}, //Trama 28
68. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1, 0x01,0x11,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 29
70. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x12,0xf0,0xf0, 0x10,0x0d,0x0e,0x00,0xff,0xef,0x18,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7f,0x23}, //Trama 30
72. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1, 0x01,0x13,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 31
74. {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x03,0xf0,0xf0, 0x53,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 32
76. {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x03,0xf0,0xf1, 0x73,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}}; //Trama 33\*/
77. //ARP frames
78. //Remove the "t1" I made that way because I wanted to prove something else
79. **BYTE** t1[20][125]={
80. /\* TRAMA 1 \*/
81. {
82. 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x23, /\* .......# \*/
83. 0x8b, 0x46, 0xe9, 0xad, 0x08, 0x06, 0x00, 0x04, /\* .F...... \*/
84. 0x08, 0x00, 0x06, 0x04, 0x00, 0x01, 0x00, 0x23, /\* .......# \*/
85. 0x8b, 0x46, 0xe9, 0xad, 0x94, 0xcc, 0x39, 0xcb, /\* .F....9. \*/
86. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x94, 0xcc, /\* ........ \*/
87. 0x39, 0xfe                                      /\* 9. \*/
88. },
90. /\*TRAMA 2 \*/
91. {
92. 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x00, 0x1f, /\* .#.F.... \*/
93. 0x45, 0x9d, 0x1e, 0xa2, 0x08, 0x06, 0x00, 0x01, /\* E....... \*/
94. 0x08, 0x00, 0x06, 0x04, 0x00, 0x02, 0x00, 0x1f, /\* ........ \*/
95. 0x45, 0x9d, 0x1e, 0xa2, 0x94, 0xcc, 0x39, 0xfe, /\* E.....9. \*/
96. 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x94, 0xcc, /\* .#.F.... \*/
97. 0x39, 0xcb, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* 9....... \*/
98. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
99. 0x00, 0x00, 0x00, 0x00                          /\* .... \*/
100. },
102. /\* TRAMA 3 \*/
103. {
104. 0x00, 0x1f, 0x45, 0x9d, 0x1e, 0xa2, 0x00, 0x23, /\* ..E....# \*/
105. 0x8b, 0x46, 0xe9, 0xad, 0x08, 0x00, 0x46, 0x00, /\* .F....F. \*/
106. 0x80, 0x42, 0x04, 0x55, 0x34, 0x11, 0x80, 0x11, /\* .B.U4... \*/
107. 0x6b, 0xf0, 0x94, 0xcc, 0x39, 0xcb, 0x94, 0xcc, /\* k...9... \*/
108. 0x67, 0x02, 0xaa, 0xbb, 0xcc, 0xdd, 0x04, 0x0c, /\* g....... \*/
109. 0x00, 0x35, 0x00, 0x2e, 0x85, 0x7c, 0xe2, 0x1a, /\* .5...|.. \*/
110. 0x01, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
111. 0x00, 0x00, 0x03, 0x77, 0x77, 0x77, 0x03, 0x69, /\* ...www.i \*/
112. 0x73, 0x63, 0x05, 0x65, 0x73, 0x63, 0x6f, 0x6d, /\* sc.escom \*/
113. 0x03, 0x69, 0x70, 0x6e, 0x02, 0x6d, 0x78, 0x00, /\* .ipn.mx. \*/
114. 0x00, 0x1c, 0x00, 0x01                          /\* .... \*/
115. },
117. /\* TRAMA 4 \*/
118. {
119. 0x00, 0x02, 0xb3, 0x9c, 0xdf, 0x1b, 0x00, 0x02, /\* ........ \*/
120. 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x04, 0xf0, 0xf0, /\* ........ \*/
121. 0x01, 0xc9, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
122. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
123. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
124. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
125. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
126. 0x00, 0x00, 0x00, 0x00, 0x00, 0x7c, 0x9b, 0x6d  /\* .....|.m \*/
127. },
129. /\* TRAMA 5 \*/
130. {
131. 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x23, /\* .......# \*/
132. 0x8b, 0x46, 0xe9, 0xad, 0x08, 0x06, 0x00, 0x10, /\* .F...... \*/
133. 0x08, 0x00, 0x06, 0x04, 0x00, 0x03, 0x00, 0x23, /\* .......# \*/
134. 0x8b, 0x46, 0xe9, 0xad, 0x94, 0xcc, 0x39, 0xcb, /\* .F....9. \*/
135. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x94, 0xcc, /\* ........ \*/
136. 0x3a, 0xe1                                      /\* :. \*/
137. },
139. /\* TRAMA 6\*/
140. {
141. 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x00, 0x1f, /\* .#.F.... \*/
142. 0x45, 0x9d, 0x1e, 0xa2, 0x08, 0x06, 0x00, 0x10, /\* E....... \*/
143. 0x08, 0x00, 0x06, 0x04, 0x00, 0x04, 0x00, 0x1f, /\* ........ \*/
144. 0x45, 0x9d, 0x1e, 0xa2, 0x94, 0xcc, 0x3a, 0xe1, /\* E.....:. \*/
145. 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x94, 0xcc, /\* .#.F.... \*/
146. 0x39, 0xcb, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* 9....... \*/
147. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
148. 0x00, 0x00, 0x00, 0x00                          /\* .... \*/
149. },
151. /\* TRAMA 7 \*/
152. {
153. 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x00, 0x1f, /\* .#.F.... \*/
154. 0x45, 0x9d, 0x1e, 0xa2, 0x08, 0x00, 0x45, 0x00, /\* E.....E. \*/
155. 0x00, 0x6f, 0x90, 0x30, 0x40, 0x00, 0xfb, 0x11, /\* .o.0@... \*/
156. 0x24, 0xe7, 0x94, 0xcc, 0x67, 0x02, 0x94, 0xcc, /\* $...g... \*/
157. 0x39, 0xcb, 0x00, 0x35, 0x04, 0x0c, 0x00, 0x5b, /\* 9..5...[ \*/
158. 0xe8, 0x60, 0xe2, 0x1a, 0x85, 0x80, 0x00, 0x01, /\* .`...... \*/
159. 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x03, 0x77, /\* .......w \*/
160. 0x77, 0x77, 0x03, 0x69, 0x73, 0x63, 0x05, 0x65, /\* ww.isc.e \*/
161. 0x73, 0x63, 0x6f, 0x6d, 0x03, 0x69, 0x70, 0x6e, /\* scom.ipn \*/
162. 0x02, 0x6d, 0x78, 0x00, 0x00, 0x1c, 0x00, 0x01, /\* .mx..... \*/
163. 0xc0, 0x14, 0x00, 0x06, 0x00, 0x01, 0x00, 0x00, /\* ........ \*/
164. 0x0e, 0x10, 0x00, 0x21, 0x04, 0x64, 0x6e, 0x73, /\* ...!.dns \*/
165. 0x31, 0xc0, 0x1a, 0x03, 0x74, 0x69, 0x63, 0xc0, /\* 1...tic. \*/
166. 0x1a, 0x77, 0xec, 0xdf, 0x29, 0x00, 0x00, 0x2a, /\* .w..)..\* \*/
167. 0x30, 0x00, 0x00, 0x0e, 0x10, 0x00, 0x12, 0x75, /\* 0......u \*/
168. 0x00, 0x00, 0x00, 0x2a, 0x30                    /\* ...\*0 \*/
169. },
171. /\* TRAMA 8 \*/
172. {
173. 0x00, 0x1f, 0x45, 0x9d, 0x1e, 0xa2, 0x00, 0x23, /\* ..E....# \*/
174. 0x02, 0x03, 0x04, 0x05, 0x08, 0x00, 0x45, 0x00, /\* .F....E. \*/
175. 0x00, 0x42, 0x04, 0x56, 0x00, 0x00, 0x80, 0x11, /\* .B.V.... \*/
176. 0x6b, 0xef, 0x94, 0xcc, 0x39, 0xcb, 0x94, 0xcc, /\* k...9... \*/
177. 0x67, 0x02, 0x04, 0x0c, 0x00, 0x35, 0x00, 0x2e, /\* g....5.. \*/
178. 0xff, 0x87, 0x68, 0x2a, 0x01, 0x00, 0x00, 0x01, /\* ..h\*.... \*/
179. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x77, /\* .......w \*/
180. 0x77, 0x77, 0x03, 0x69, 0x73, 0x63, 0x05, 0x65, /\* ww.isc.e \*/
181. 0x73, 0x63, 0x6f, 0x6d, 0x03, 0x69, 0x70, 0x6e, /\* scom.ipn \*/
182. 0x02, 0x6d, 0x78, 0x00, 0x00, 0x01, 0x00, 0x01  /\* .mx..... \*/
183. },
185. /\* TRAMA 9 \*/
186. {
187. 0x00, 0x1f, 0x45, 0x9d, 0x1e, 0xa2, 0x00, 0x23, /\* ..E....# \*/
188. 0x8b, 0x46, 0xe9, 0xad, 0x08, 0x00, 0x45, 0x00, /\* .F....E. \*/
189. 0x00, 0x3c, 0x04, 0x57, 0x00, 0x00, 0x80, 0x01, /\* .<.W.... \*/
190. 0x98, 0x25, 0x94, 0xcc, 0x39, 0xcb, 0x94, 0xcc, /\* .%..9... \*/
191. 0x3a, 0xe1, 0x08, 0x00, 0x49, 0x5c, 0x03, 0x00, /\* :...I\.. \*/
192. 0x01, 0x00, 0x61, 0x62, 0x63, 0x64, 0x65, 0x66, /\* ..abcdef \*/
193. 0x67, 0x68, 0x69, 0x6a, 0x6b, 0x6c, 0x6d, 0x6e, /\* ghijklmn \*/
194. 0x6f, 0x70, 0x71, 0x72, 0x73, 0x74, 0x75, 0x76, /\* opqrstuv \*/
195. 0x77, 0x61, 0x62, 0x63, 0x64, 0x65, 0x66, 0x67, /\* wabcdefg \*/
196. 0x68, 0x69                                      /\* hi \*/
197. },
199. /\* TRAMA 10 \*/
200. {
201. 0x00, 0x02, 0xb3, 0x9c, 0xdf, 0x1b, 0x00, 0x02, /\* ........ \*/
202. 0x02, 0x03, 0x04, 0x05, 0x00, 0x04, 0xf0, 0xf0, /\* ........ \*/
203. 0x0c, 0xc7, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
204. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
205. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
206. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
207. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
208. 0x00, 0x00, 0x00, 0x00, 0x00, 0x7c, 0x9b, 0x6d  /\* .....|.m \*/
209. },
211. /\* trama 11 \*/
212. {
213. 0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x11, 0x11, /\* ........ \*/
214. 0x11, 0x11, 0x11, 0x11, 0x00, 0x03, 0xf0, 0xf0, /\* ........ \*/
215. 0x63, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
216. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
217. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
218. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
219. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
220. 0x00, 0x00, 0x00, 0x00, 0x43, 0x05, 0x90, 0x6d  /\* ....C..m \*/
221. },
223. /\* trama 12 \*/
224. {
225. 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x00, 0x1f, /\* .#.F.... \*/
226. 0x02, 0x03, 0x04, 0x05, 0x08, 0x00, 0x45, 0x00, /\* E.....E. \*/
227. 0x00, 0x3c, 0x01, 0xb5, 0x00, 0x00, 0x3f, 0x01, /\* .<....?. \*/
228. 0xdb, 0xc7, 0x94, 0xcc, 0x3a, 0xe1, 0x94, 0xcc, /\* ....:... \*/
229. 0x39, 0xcb, 0x00, 0x00, 0x51, 0x5c, 0x03, 0x00, /\* 9...Q\.. \*/
230. 0x01, 0x00, 0x61, 0x62, 0x63, 0x64, 0x65, 0x66, /\* ..abcdef \*/
231. 0x67, 0x68, 0x69, 0x6a, 0x6b, 0x6c, 0x6d, 0x6e, /\* ghijklmn \*/
232. 0x6f, 0x70, 0x71, 0x72, 0x73, 0x74, 0x75, 0x76, /\* opqrstuv \*/
233. 0x77, 0x61, 0x62, 0x63, 0x64, 0x65, 0x66, 0x67, /\* wabcdefg \*/
234. 0x68, 0x69                                      /\* hi \*/
235. },
237. /\* trama 13 \*/
238. {
239. 0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x01, /\* ........ \*/
240. 0x02, 0x03, 0x04, 0x05, 0x00, 0x03, 0xf0, 0xf1, /\* ........ \*/
241. 0x73, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
242. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
243. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
244. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
245. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
246. 0x00, 0x00, 0x00, 0x00, 0x43, 0x05, 0x90, 0x6d  /\* ....C..m \*/
247. },
249. /\* trama 14 \*/
250. {
251. 0x00, 0x1f, 0x45, 0x9d, 0x1e, 0xa2, 0x00, 0x1c, /\* ..E..... \*/
252. 0xc0, 0x7b, 0x35, 0xa1, 0x08, 0x00, 0x48, 0x00, /\* .{5...H. \*/
253. 0x00, 0x48, 0x5c, 0x7d, 0x00, 0x00, 0x80, 0x01, /\* .H\}.... \*/
254. 0x6c, 0x88, 0x94, 0xcc, 0x39, 0xc3, 0x94, 0xcc, /\* l...9... \*/
255. 0x00, 0x49, 0x07, 0x0b, 0x04, 0x00, 0x00, 0x00, /\* .I...... \*/
256. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0x00, /\* ........ \*/
257. 0x3b, 0x5c, 0x02, 0x00, 0x10, 0x00, 0x61, 0x62, /\* ;\....ab \*/
258. 0x63, 0x64, 0x65, 0x66, 0x67, 0x68, 0x69, 0x6a, /\* cdefghij \*/
259. 0x6b, 0x6c, 0x6d, 0x6e, 0x6f, 0x70, 0x71, 0x72, /\* klmnopqr \*/
260. 0x73, 0x74, 0x75, 0x76, 0x77, 0x61, 0x62, 0x63, /\* stuvwabc \*/
261. 0x64, 0x65, 0x66, 0x67, 0x68, 0x69              /\* defghi \*/
262. },
264. /\* trama 15\*/
265. {
266. 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x00, 0x1f, /\* .#.F.... \*/
267. 0x45, 0x9d, 0x1e, 0xa2, 0x80, 0x35, 0x00, 0x01, /\* E....5.. \*/
268. 0x08, 0x00, 0x06, 0x04, 0x00, 0x03, 0x00, 0x1f, /\* ........ \*/
269. 0x45, 0x9d, 0x1e, 0xa2, 0x94, 0xcc, 0x3a, 0xe1, /\* E.....:. \*/
270. 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x94, 0xcc, /\* .#.F.... \*/
271. 0x39, 0xcb, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* 9....... \*/
272. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
273. 0x00, 0x00, 0x00, 0x00, 0xd8, 0xee, 0xdf, 0xb0  /\* ........ \*/
274. },
276. /\* trama 16 \*/
277. {
278. 0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x22, 0x22, /\* ........ \*/
279. 0x22, 0x22, 0x22, 0x22, 0x00, 0x03, 0xf0, 0xf0, /\* ........ \*/
280. 0x73, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
281. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
282. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
283. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
284. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
285. 0x00, 0x00, 0x00, 0x00, 0x43, 0x05, 0x90, 0x6d  /\* ....C..m \*/
286. },
288. /\*trama 17 \*/
289. {
290. 0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x02, /\* ........ \*/
291. 0xb3, 0x9c, 0xdf, 0x1b, 0x00, 0x03, 0xf0, 0xf0, /\* ........ \*/
292. 0x53, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
293. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
294. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
295. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
296. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
297. 0x00, 0x00, 0x00, 0x00, 0x43, 0x05, 0x90, 0x6d  /\* ....C..m \*/
298. },
300. /\* trama 18 \*/
301. {
302. 0x02, 0xff, 0x53, 0xc3, 0xe9, 0xab, 0x00, 0xff, /\* ..S..... \*/
303. 0x66, 0x7f, 0xd4, 0x3c, 0x08, 0x00, 0x45, 0x00, /\* f..<..E. \*/
304. 0x00, 0x30, 0x2c, 0x00, 0x40, 0x00, 0x80, 0x06, /\* .0,.@... \*/
305. 0x4b, 0x74, 0xc0, 0xa8, 0x01, 0x02, 0xc0, 0xa8, /\* Kt...... \*/
306. 0x01, 0x01, 0x04, 0x03, 0x00, 0x15, 0x00, 0x3b, /\* .......; \*/
307. 0xcf, 0x44, 0x00, 0x00, 0x00, 0x00, 0x70, 0x20, /\* .D....p  \*/
308. 0x20, 0x00, 0x0c, 0x34, 0x00, 0x00, 0x02, 0x04, /\*  ..4.... \*/
309. 0x05, 0xb4, 0x01, 0x01, 0x04, 0x02              /\* ...... \*/
310. },
312. /\*trama 19 \*/
313. {
314. 0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x02, /\* ........ \*/
315. 0xb3, 0x9c, 0xdf, 0x1b, 0x00, 0x03, 0xf0, 0xf1, /\* ........ \*/
316. 0x53, 0x0b, 0x0e, 0x00, 0xff, 0xef, 0x14, 0x00, /\* ........ \*/
317. 0x00, 0x00, 0x28, 0x00, 0x00, 0x00, 0x7f, 0x23, /\* ..(....# \*/
318. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
319. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
320. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
321. 0x00, 0x00, 0x00, 0x00, 0x01, 0x99, 0x98, 0x6d  /\* .......m \*/
322. },
323. /\* trama 20 \*/
324. {
325. 0x00, 0xff, 0x66, 0x7f, 0xd4, 0x3c, 0x02, 0xff, /\* ..f..<.. \*/
326. 0x53, 0xc3, 0xe9, 0xab, 0x08, 0x00, 0x45, 0x00, /\* S.....E. \*/
327. 0x00, 0x30, 0x05, 0xc4, 0x40, 0x00, 0x80, 0x06, /\* .0..@... \*/
328. 0x71, 0xb0, 0xc0, 0xa8, 0x01, 0x01, 0xc0, 0xa8, /\* q....... \*/
329. 0x01, 0x02, 0x00, 0x15, 0x04, 0x03, 0x21, 0x5d, /\* ......!] \*/
330. 0x3a, 0x44, 0x00, 0x3b, 0xcf, 0x45, 0x70, 0x12, /\* :D.;.Ep. \*/
331. 0x44, 0x70, 0x8c, 0x11, 0x00, 0x00, 0x02, 0x04, /\* Dp...... \*/
332. 0x05, 0xb4, 0x01, 0x01, 0x04, 0x02              /\* ...... \*/
333. }};
335. ////////////////////////////////////////////////////////////////////////////////////////////////////////////////
336. **BYTE** t[20][125] = {{
337. 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x23, /\* .......# \*/
338. 0x8b, 0x46, 0xe9, 0xad, 0x08, 0x06, 0x00, 0x04, /\* .F...... \*/
339. 0x08, 0x00, 0x06, 0x04, 0x00, 0x01, 0x00, 0x23, /\* .......# \*/
340. 0x8b, 0x46, 0xe9, 0xad, 0x94, 0xcc, 0x39, 0xcb, /\* .F....9. \*/
341. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x94, 0xcc, /\* ........ \*/
342. 0x39, 0xfe                                      /\* 9. \*/
343. },
345. {
346. 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x00, 0x1f, /\* .#.F.... \*/
347. 0x45, 0x9d, 0x1e, 0xa2, 0x08, 0x06, 0x00, 0x01, /\* E....... \*/
348. 0x08, 0x00, 0x06, 0x04, 0x00, 0x02, 0x00, 0x1f, /\* ........ \*/
349. 0x45, 0x9d, 0x1e, 0xa2, 0x94, 0xcc, 0x39, 0xfe, /\* E.....9. \*/
350. 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x94, 0xcc, /\* .#.F.... \*/
351. 0x39, 0xcb, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* 9....... \*/
352. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
353. 0x00, 0x00, 0x00, 0x00                          /\* .... \*/
354. },
355. {
356. 0x00, 0x1f, 0x45, 0x9d, 0x1e, 0xa2, 0x00, 0x23, /\* ..E....# \*/
357. 0x8b, 0x46, 0xe9, 0xad, 0x08, 0x00, 0x46, 0x00, /\* .F....F. \*/
358. 0x80, 0x42, 0x04, 0x55, 0x34, 0x11, 0x80, 0x11, /\* .B.U4... \*/
359. 0x6b, 0xf0, 0x94, 0xcc, 0x39, 0xcb, 0x94, 0xcc, /\* k...9... \*/
360. 0x67, 0x02, 0xaa, 0xbb, 0xcc, 0xdd, 0x04, 0x0c, /\* g....... \*/
361. 0x00, 0x35, 0x00, 0x2e, 0x85, 0x7c, 0xe2, 0x1a, /\* .5...|.. \*/
362. 0x01, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
363. 0x00, 0x00, 0x03, 0x77, 0x77, 0x77, 0x03, 0x69, /\* ...www.i \*/
364. 0x73, 0x63, 0x05, 0x65, 0x73, 0x63, 0x6f, 0x6d, /\* sc.escom \*/
365. 0x03, 0x69, 0x70, 0x6e, 0x02, 0x6d, 0x78, 0x00, /\* .ipn.mx. \*/
366. 0x00, 0x1c, 0x00, 0x01                          /\* .... \*/
367. },
368. {
369. 0x00, 0x02, 0xb3, 0x9c, 0xdf, 0x1b, 0x00, 0x02, /\* ........ \*/
370. 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x04, 0xf0, 0xf1, /\* ........ \*/
371. 0x09, 0x8d, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
372. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
373. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
374. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
375. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
376. 0x00, 0x00, 0x00, 0x00, 0x00, 0x7c, 0x9b, 0x6d  /\* .....|.m \*/
377. },
378. {
379. 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x23, /\* .......# \*/
380. 0x8b, 0x46, 0xe9, 0xad, 0x08, 0x06, 0x00, 0x10, /\* .F...... \*/
381. 0x08, 0x00, 0x06, 0x04, 0x00, 0x03, 0x00, 0x23, /\* .......# \*/
382. 0x8b, 0x46, 0xe9, 0xad, 0x94, 0xcc, 0x39, 0xcb, /\* .F....9. \*/
383. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x94, 0xcc, /\* ........ \*/
384. 0x3a, 0xe1                                      /\* :. \*/
385. },
386. {
387. 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x00, 0x1f, /\* .#.F.... \*/
388. 0x45, 0x9d, 0x1e, 0xa2, 0x08, 0x06, 0x00, 0x10, /\* E....... \*/
389. 0x08, 0x00, 0x06, 0x04, 0x00, 0x04, 0x00, 0x1f, /\* ........ \*/
390. 0x45, 0x9d, 0x1e, 0xa2, 0x94, 0xcc, 0x3a, 0xe1, /\* E.....:. \*/
391. 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x94, 0xcc, /\* .#.F.... \*/
392. 0x39, 0xcb, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* 9....... \*/
393. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
394. 0x00, 0x00, 0x00, 0x00                          /\* .... \*/
395. },
396. {
397. 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x00, 0x1f, /\* .#.F.... \*/
398. 0x45, 0x9d, 0x1e, 0xa2, 0x08, 0x00, 0x45, 0x00, /\* E.....E. \*/
399. 0x00, 0x6f, 0x90, 0x30, 0x40, 0x00, 0xfb, 0x11, /\* .o.0@... \*/
400. 0x24, 0xe7, 0x94, 0xcc, 0x67, 0x02, 0x94, 0xcc, /\* $...g... \*/
401. 0x39, 0xcb, 0x00, 0x35, 0x04, 0x0c, 0x00, 0x5b, /\* 9..5...[ \*/
402. 0xe8, 0x60, 0xe2, 0x1a, 0x85, 0x80, 0x00, 0x01, /\* .`...... \*/
403. 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x03, 0x77, /\* .......w \*/
404. 0x77, 0x77, 0x03, 0x69, 0x73, 0x63, 0x05, 0x65, /\* ww.isc.e \*/
405. 0x73, 0x63, 0x6f, 0x6d, 0x03, 0x69, 0x70, 0x6e, /\* scom.ipn \*/
406. 0x02, 0x6d, 0x78, 0x00, 0x00, 0x1c, 0x00, 0x01, /\* .mx..... \*/
407. 0xc0, 0x14, 0x00, 0x06, 0x00, 0x01, 0x00, 0x00, /\* ........ \*/
408. 0x0e, 0x10, 0x00, 0x21, 0x04, 0x64, 0x6e, 0x73, /\* ...!.dns \*/
409. 0x31, 0xc0, 0x1a, 0x03, 0x74, 0x69, 0x63, 0xc0, /\* 1...tic. \*/
410. 0x1a, 0x77, 0xec, 0xdf, 0x29, 0x00, 0x00, 0x2a, /\* .w..)..\* \*/
411. 0x30, 0x00, 0x00, 0x0e, 0x10, 0x00, 0x12, 0x75, /\* 0......u \*/
412. 0x00, 0x00, 0x00, 0x2a, 0x30                    /\* ...\*0 \*/
413. },
414. {
415. 0x00, 0x1f, 0x45, 0x9d, 0x1e, 0xa2, 0x00, 0x23, /\* ..E....# \*/
416. 0x8b, 0x46, 0xe9, 0xad, 0x08, 0x00, 0x45, 0x00, /\* .F....E. \*/
417. 0x00, 0x42, 0x04, 0x56, 0x00, 0x00, 0x80, 0x11, /\* .B.V.... \*/
418. 0x6b, 0xef, 0x94, 0xcc, 0x39, 0xcb, 0x94, 0xcc, /\* k...9... \*/
419. 0x67, 0x02, 0x04, 0x0c, 0x00, 0x35, 0x00, 0x2e, /\* g....5.. \*/
420. 0xff, 0x87, 0x68, 0x2a, 0x01, 0x00, 0x00, 0x01, /\* ..h\*.... \*/
421. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x03, 0x77, /\* .......w \*/
422. 0x77, 0x77, 0x03, 0x69, 0x73, 0x63, 0x05, 0x65, /\* ww.isc.e \*/
423. 0x73, 0x63, 0x6f, 0x6d, 0x03, 0x69, 0x70, 0x6e, /\* scom.ipn \*/
424. 0x02, 0x6d, 0x78, 0x00, 0x00, 0x01, 0x00, 0x01  /\* .mx..... \*/
425. },
426. {
427. 0x00, 0x1f, 0x45, 0x9d, 0x1e, 0xa2, 0x00, 0x23, /\* ..E....# \*/
428. 0x8b, 0x46, 0xe9, 0xad, 0x08, 0x00, 0x45, 0x00, /\* .F....E. \*/
429. 0x00, 0x3c, 0x04, 0x57, 0x00, 0x00, 0x80, 0x01, /\* .<.W.... \*/
430. 0x98, 0x25, 0x94, 0xcc, 0x39, 0xcb, 0x94, 0xcc, /\* .%..9... \*/
431. 0x3a, 0xe1, 0x08, 0x00, 0x49, 0x5c, 0x03, 0x00, /\* :...I\.. \*/
432. 0x01, 0x00, 0x61, 0x62, 0x63, 0x64, 0x65, 0x66, /\* ..abcdef \*/
433. 0x67, 0x68, 0x69, 0x6a, 0x6b, 0x6c, 0x6d, 0x6e, /\* ghijklmn \*/
434. 0x6f, 0x70, 0x71, 0x72, 0x73, 0x74, 0x75, 0x76, /\* opqrstuv \*/
435. 0x77, 0x61, 0x62, 0x63, 0x64, 0x65, 0x66, 0x67, /\* wabcdefg \*/
436. 0x68, 0x69                                      /\* hi \*/
437. },
438. {
439. 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x00, 0x1f, /\* .#.F.... \*/
440. 0x45, 0x9d, 0x1e, 0xa2, 0x08, 0x00, 0x45, 0x00, /\* E.....E. \*/
441. 0x00, 0x3c, 0x01, 0xb5, 0x00, 0x00, 0x3f, 0x01, /\* .<....?. \*/
442. 0xdb, 0xc7, 0x94, 0xcc, 0x3a, 0xe1, 0x94, 0xcc, /\* ....:... \*/
443. 0x39, 0xcb, 0x00, 0x00, 0x51, 0x5c, 0x03, 0x00, /\* 9...Q\.. \*/
444. 0x01, 0x00, 0x61, 0x62, 0x63, 0x64, 0x65, 0x66, /\* ..abcdef \*/
445. 0x67, 0x68, 0x69, 0x6a, 0x6b, 0x6c, 0x6d, 0x6e, /\* ghijklmn \*/
446. 0x6f, 0x70, 0x71, 0x72, 0x73, 0x74, 0x75, 0x76, /\* opqrstuv \*/
447. 0x77, 0x61, 0x62, 0x63, 0x64, 0x65, 0x66, 0x67, /\* wabcdefg \*/
448. 0x68, 0x69                                      /\* hi \*/
449. },
450. {
451. 0x00, 0x1f, 0x45, 0x9d, 0x1e, 0xa2, 0x00, 0x1c, /\* ..E..... \*/
452. 0xc0, 0x7b, 0x35, 0xa1, 0x08, 0x00, 0x48, 0x00, /\* .{5...H. \*/
453. 0x00, 0x48, 0x5c, 0x7d, 0x00, 0x00, 0x80, 0x01, /\* .H\}.... \*/
454. 0x6c, 0x88, 0x94, 0xcc, 0x39, 0xc3, 0x94, 0xcc, /\* l...9... \*/
455. 0x00, 0x49, 0x07, 0x0b, 0x04, 0x00, 0x00, 0x00, /\* .I...... \*/
456. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x08, 0x00, /\* ........ \*/
457. 0x3b, 0x5c, 0x02, 0x00, 0x10, 0x00, 0x61, 0x62, /\* ;\....ab \*/
458. 0x63, 0x64, 0x65, 0x66, 0x67, 0x68, 0x69, 0x6a, /\* cdefghij \*/
459. 0x6b, 0x6c, 0x6d, 0x6e, 0x6f, 0x70, 0x71, 0x72, /\* klmnopqr \*/
460. 0x73, 0x74, 0x75, 0x76, 0x77, 0x61, 0x62, 0x63, /\* stuvwabc \*/
461. 0x64, 0x65, 0x66, 0x67, 0x68, 0x69              /\* defghi \*/
462. },
463. {
464. 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x00, 0x1f, /\* .#.F.... \*/
465. 0x45, 0x9d, 0x1e, 0xa2, 0x80, 0x35, 0x00, 0x01, /\* E....5.. \*/
466. 0x08, 0x00, 0x06, 0x04, 0x00, 0x03, 0x00, 0x1f, /\* ........ \*/
467. 0x45, 0x9d, 0x1e, 0xa2, 0x94, 0xcc, 0x3a, 0xe1, /\* E.....:. \*/
468. 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x94, 0xcc, /\* .#.F.... \*/
469. 0x39, 0xcb, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* 9....... \*/
470. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
471. 0x00, 0x00, 0x00, 0x00, 0xd8, 0xee, 0xdf, 0xb0  /\* ........ \*/
472. },
473. {
474. 0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x02, /\* ........ \*/
475. 0xb3, 0x9c, 0xdf, 0x1b, 0x00, 0x03, 0xf0, 0xf0, /\* ........ \*/
476. 0x7f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
477. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
478. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
479. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
480. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
481. 0x00, 0x00, 0x00, 0x00, 0x43, 0x05, 0x90, 0x6d  /\* ....C..m \*/
482. },
483. {
484. 0x02, 0xff, 0x53, 0xc3, 0xe9, 0xab, 0x00, 0xff, /\* ..S..... \*/
485. 0x66, 0x7f, 0xd4, 0x3c, 0x08, 0x00, 0x45, 0x00, /\* f..<..E. \*/
486. 0x00, 0x30, 0x2c, 0x00, 0x40, 0x00, 0x80, 0x06, /\* .0,.@... \*/
487. 0x4b, 0x74, 0xc0, 0xa8, 0x01, 0x02, 0xc0, 0xa8, /\* Kt...... \*/
488. 0x01, 0x01, 0x04, 0x03, 0x00, 0x15, 0x00, 0x3b, /\* .......; \*/
489. 0xcf, 0x44, 0x00, 0x00, 0x00, 0x00, 0x70, 0x20, /\* .D....p  \*/
490. 0x20, 0x00, 0x0c, 0x34, 0x00, 0x00, 0x02, 0x04, /\*  ..4.... \*/
491. 0x05, 0xb4, 0x01, 0x01, 0x04, 0x02              /\* ...... \*/
492. },
493. {
494. 0x00, 0xff, 0x66, 0x7f, 0xd4, 0x3c, 0x02, 0xff, /\* ..f..<.. \*/
495. 0x53, 0xc3, 0xe9, 0xab, 0x08, 0x00, 0x45, 0x00, /\* S.....E. \*/
496. 0x00, 0x30, 0x05, 0xc4, 0x40, 0x00, 0x80, 0x06, /\* .0..@... \*/
497. 0x71, 0xb0, 0xc0, 0xa8, 0x01, 0x01, 0xc0, 0xa8, /\* q....... \*/
498. 0x01, 0x02, 0x00, 0x15, 0x04, 0x03, 0x21, 0x5d, /\* ......!] \*/
499. 0x3a, 0x44, 0x00, 0x3b, 0xcf, 0x45, 0x70, 0x12, /\* :D.;.Ep. \*/
500. 0x44, 0x70, 0x8c, 0x11, 0x00, 0x00, 0x02, 0x04, /\* Dp...... \*/
501. 0x05, 0xb4, 0x01, 0x01, 0x04, 0x02              /\* ...... \*/
502. },
503. {
504. 0x00, 0x02, 0xb3, 0x9c, 0xae, 0xba, 0x00, 0x02, /\* ........ \*/
505. 0xb3, 0x9c, 0xdf, 0x1b, 0x00, 0x12, 0xf0, 0xf0, /\* ........ \*/
506. 0x0a, 0x0b, 0x0e, 0x00, 0xff, 0xef, 0x14, 0x00, /\* ........ \*/
507. 0x00, 0x00, 0x28, 0x00, 0x00, 0x00, 0x7f, 0x23, /\* ..(....# \*/
508. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
509. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
510. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, /\* ........ \*/
511. 0x00, 0x00, 0x00, 0x00, 0x01, 0x99, 0x98, 0x6d  /\* .......m \*/
512. }};

515. /\*BYTE t[] = {0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x08, 0x06, 0x00, 0x01,
516. 0x08, 0x00, 0x06, 0x04, 0x00, 0x01, 0x00, 0x23, 0x8b, 0x46, 0xe9, 0xad, 0x94, 0xcc, 0x39, 0xcb,
517. 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x94, 0xcc, 0x39, 0xfe};\*/
518. //For IP
519. /\*
520. BYTE t[] = {0x13, 0x42, 0x18, 0x63, 0x96, 0x12, 0x13, 0x42,  0x18, 0x96, 0x35, 0x72, 0x08, 0x00, 0x46, 0x00,
521. 0x80, 0x42, 0x04, 0x55, 0x34, 0x11, 0x80, 0x11,  0x6b, 0xf0, 0x94, 0xcc, 0x39, 0xcb, 0x94, 0xcc,
522. 0x67, 0x02, 0xaa, 0xbb, 0xcc, 0xdd, 0x04, 0x0c,  0x00, 0x35, 0x00, 0x2e, 0x85, 0x7c, 0xe2, 0x1a,
523. 0x01, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x00,  0x00, 0x00, 0x03, 0x77, 0x77, 0x77, 0x03, 0x69,
524. 0x73, 0x63, 0x05, 0x65, 0x73, 0x63, 0x6f, 0x6d,  0x03, 0x69, 0x70, 0x6e, 0x02, 0x6d, 0x78, 0x00,
525. 0x00, 0x1c, 0x00, 0x01};
526. \*/
527. **BYTE** resultado[150];
529. //For just one frame
530. /\*
531. if(analizaTrama(t, resultado))
532. printf("%s\n", resultado);
533. else
534. printf("No es trama LLC\n");
535. \*/
537. //For LLC frames
538. /\*
539. for(BYTE i = 0; i < 33; i++)
540. if(analizaTrama(t[i], resultado))
541. printf("%d %s\n", i + 1, resultado);
542. else
543. printf("No es trama LLC\n");
544. \*/
546. //For ARP frames
548. **for**(**BYTE** i = 0; i < 20; i++)
549. **if**(analizaTrama(t[i], resultado))
550. printf("%d %s\n\n", i + 1, resultado);
551. **else**
552. printf("No es trama LLC\n");
554. exit(EXIT\_SUCCESS);
555. }
557. **int** analizaTrama(**BYTE** t[], **BYTE** resultado[])
558. {
559. **BYTE** supervision[][5] = {"RR", "RNR", "REJ", "SREJ"};
560. **BYTE** uc[][6] = {"UI", "SIM", "-", "SARN", "UP", "-", "-", "SABM", "DISC", "-", "-", "SARME", "-", "-", "-", "SABME",
561. "SNRM", "-", "-", "XID", "-", "-", "-", "-", "-", "-", "-", "SNRME"};
562. **BYTE** ur[][5] = {"UI", "RIM", "-", "DM", "-", "-", "-", "-", "RD", "-", "-", "-", "UA", "-", "-", "-", "-", "FRMR", "-",
563. "XID", "-", "-", "-", "-", "-", "-", "-", "-"};
564. **BYTE** icmp\_type[][45] = {"Echo\_Reply", "-", "-", "Destination\_Unreachable", "Source\_Quench", "Redirect", "-", "-", "Echo", "Router\_Advertisement",
565. "Router\_Selection", "Time Exceeded", "Parameter\_Problem", "Timestamp", "Timestamp\_Reply", "Information\_Request", "Information\_Reply",
566. "Address\_Mask\_Request", "Address\_Mask\_Reply", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "Traceroute"};
567. **BYTE** icmp\_code[][45] = {"Net\_Unreachable", "Host\_Reachable", "Protocol\_Unreachable", "Port\_Unreachable", "Fragmentation\_Needed\_&\_DF\_Set", "Source\_Route\_Failed",
568. "Destination\_Network\_Unknown", "Destination\_Host\_Unknown", "Source\_Host\_Isloated", "Network\_Administratively\_Prohibited", "Host\_Administratively\_Prohibited",
569. "Network\_Unreachable\_for\_TOS", "Host\_Unreachable\_for\_TOS", "Communication\_Administratively\_Prohibited"};
570. unsigned **short** **int** ToT = 0;
571. **BYTE** ihl = 0;
573. //Copiamos los bytes 12 y 13 de la trama en ToT (Tamaño o tipo)
574. **for**(**BYTE** j = 128; j > 0; j >>= 1)
575. **if**(t[12] & j)
576. ToT = (ToT << 1) | 1;
577. **else**
578. ToT <<= 1;
580. **for**(**BYTE** j = 128; j > 0; j >>= 1)
581. **if**(t[13] & j)
582. ToT = (ToT << 1) | 1;
583. **else**
584. ToT <<= 1;
586. //Tipo LLC
587. **if**(ToT <= 1500)
588. {
589. **switch**(t[16] & 3)
590. {
591. **case** 0: **case** 2:
592. sprintf(resultado, "T-I N(s) = %d, N(r) = %d", ((t[16] >> 1) & 255), ((t[17] >> 1) & 255));
593. **break**;
594. **case** 1:
595. sprintf(resultado, "T-S %s, N(r) = %d", supervision[(t[16] >> 2) & 3], ((t[17] >> 1) & 255));
596. **break**;
597. **case** 3:
598. **if**(((t[16] >> 4) & 1) && (t[15] & 1)) //Respuesta no numerada
599. sprintf(resultado, "T-U %s", ur[((t[16] >> 2) & 3) | ((t[16] >> 3) & 28)]);
600. **else** **if**(((t[16] >> 4) & 1) && !(t[15] & 0)) //Comando no numerado
601. sprintf(resultado, "T-U %s", uc[((t[16] >> 2) & 3) | ((t[16] >> 3) & 28)]);
602. **else** //?
603. sprintf(resultado, "Trama Intrusa");
604. **break**;
605. }
607. **return** 1;
608. }
609. **else** **if**(ToT == 2048) //Tipo IP
610. {
611. sprintf(resultado, "----------IP----------\n");
612. sprintf(resultado + strlen(resultado), "Version: %d\n", (t[14] >> 4));
613. sprintf(resultado + strlen(resultado), "IHL: %d bytes\n", (t[14] & 15) \* 4);
614. ihl = (t[14] & 15) \* 4;
615. sprintf(resultado + strlen(resultado), "Tipo Servico: ");
616. **if**(t[15] & 2)
617. sprintf(resultado + strlen(resultado), "| Maximize Delay | ");
618. **if**(t[15] & 4)
619. sprintf(resultado + strlen(resultado), "| Maximize Throughout | ");
620. **if**(t[15] & 8)
621. sprintf(resultado + strlen(resultado), "| Maximize Reliability | ");
622. **if**(t[15] & 16)
623. sprintf(resultado + strlen(resultado), "| Minimize Cost |");
624. sprintf(resultado + strlen(resultado), "\n");
626. sprintf(resultado + strlen(resultado), "Tamanio total: %d\n", (t[16] << 8) + t[17]);
627. sprintf(resultado + strlen(resultado), "Identificador: %d\n", (t[18] << 8) + t[19]);
628. sprintf(resultado + strlen(resultado), "Offset: %d saltos en bytes", (((t[20] & 31) << 8) + t[21]) \* 8);
629. **if**(t[20] & 64)
630. sprintf(resultado + strlen(resultado), "| Don't fragment ");
631. **else**
632. sprintf(resultado + strlen(resultado), "| Fragmented ");
633. **if**(t[20] & 64)
634. sprintf(resultado + strlen(resultado), "| More fragments ");
635. **else**
636. sprintf(resultado + strlen(resultado), "| No more fragments ");
637. sprintf(resultado + strlen(resultado), "\n");
639. sprintf(resultado + strlen(resultado), "TTL: %d\n", t[22]);
640. sprintf(resultado + strlen(resultado), "Protocolo: ");
641. **if**(t[23] & 1)
642. sprintf(resultado + strlen(resultado), "ICMP\n");
643. **else** **if**(t[23] & 6)
644. sprintf(resultado + strlen(resultado), "TCP\n");
645. **else** **if**(t[23] & 17)
646. sprintf(resultado + strlen(resultado), "UDP\n");
647. **else**
648. sprintf(resultado + strlen(resultado), "Otro\n");
650. sprintf(resultado + strlen(resultado), "Checksum Trama: %.4X\n", (t[24] << 8) + t[25]);
651. unsigned **short** checksum = 0;
652. unsigned **short** temp = 0;
654. **for**(**int** j = 128; j > 0; j >>= 1)
655. **if**(t[14] & j)
656. checksum = (checksum << 1) | 1;
657. **else**
658. checksum <<= 1;
659. **for**(**int** j = 128; j > 0; j >>= 1)
660. **if**(t[15] & j)
661. checksum = (checksum << 1) | 1;
662. **else**
663. checksum <<= 1;
665. //sprintf(resultado + strlen(resultado), "Checksum: %4x\n", checksum);
666. //No calculamos los t[24] ni t[25] porque corresponden al checksum previamente calculado
667. **for**(**int** i = 16; i < 14 + ihl; i++)
668. {
669. **if**(i != 24)
670. {
671. **for**(**int** j = 128; j > 0; j >>= 1)
672. **if**(t[i] & j)
673. temp = (temp << 1) | 1;
674. **else**
675. temp <<= 1;
676. i++;
677. **for**(**int** j = 128; j > 0; j >>= 1)
678. **if**(t[i] & j)
679. temp = (temp << 1) | 1;
680. **else**
681. temp <<= 1;
683. **if**(((checksum >> 12) + (temp >> 12)) > 15)
684. checksum += 1;
685. //sprintf(resultado + strlen(resultado), "Temp: %4x\n", temp);
686. checksum += temp;
687. //sprintf(resultado + strlen(resultado), "Checksum: %4x\n", checksum);
688. //sprintf(resultado + strlen(resultado), "\n");
689. }
690. **else**
691. i++;
692. }
693. checksum = ~checksum;
694. sprintf(resultado + strlen(resultado), "Checksum Calculado: %.4X ", checksum);
695. **if**(checksum != ((t[24] << 8) + t[25]))
696. sprintf(resultado + strlen(resultado), "Error! :(\n");
697. **else**
698. sprintf(resultado + strlen(resultado), "Correcto! :)\n");
700. sprintf(resultado + strlen(resultado), "IP Origen: ");
701. **for**(**int** i = 26; i < 26 + (t[14] >> 4); i++)
702. {
703. **if**(i == (25 + (t[14] >> 4)))
704. sprintf(resultado + strlen(resultado), "%d", t[i]);
705. **else**
706. sprintf(resultado + strlen(resultado), "%d.", t[i]);
707. }
708. sprintf(resultado + strlen(resultado), "\nIP Destino: ");
709. **for**(**int** i = 30; i < 30 + (t[14] >> 4); i++)
710. {
711. **if**(i == (29 + (t[14] >> 4)))
712. sprintf(resultado + strlen(resultado), "%d", t[i]);
713. **else**
714. sprintf(resultado + strlen(resultado), "%d.", t[i]);
715. }
716. **if**(ihl == 20)
717. sprintf(resultado + strlen(resultado), "\nNo hay opciones");
718. **else**
719. {
720. sprintf(resultado + strlen(resultado), "\nOpciones: ");
721. **for**(**int** i = 34; i < 34 + (ihl - 20); i++)
722. {
723. **if**(i == 33 + (ihl - 20))
724. sprintf(resultado + strlen(resultado), "%.2X", t[i]);
725. **else**
726. sprintf(resultado + strlen(resultado), "%.2X:", t[i]);
727. }
728. }
729. **if**(t[23] & 1) //ICMP
730. {
731. sprintf(resultado + strlen(resultado), "\n----------ICMP----------\n");
732. sprintf(resultado + strlen(resultado), "Type: %s\n", icmp\_type[t[14 + ihl] & 31]); //It is 14 beacuse of Ethernet Header
733. //if(t[34 + (ihl - 20)] == 3) Es lo mismo al parecer
734. **if**((t[14 + ihl] & 31) == 3)
735. sprintf(resultado + strlen(resultado), "Code: %s\n", icmp\_code[t[15 + ihl] & 15]);
736. sprintf(resultado + strlen(resultado), "Checksum Trama: %.4X\n", (t[16 + ihl] << 8) + t[17 + ihl]);
737. unsigned **short** checksum = 0;
739. **for**(**int** j = 128; j > 0; j >>= 1)
740. **if**(t[14 + ihl] & j)
741. checksum = (checksum << 1) | 1;
742. **else**
743. checksum <<= 1;
744. **for**(**int** j = 128; j > 0; j >>= 1)
745. **if**(t[15 + ihl] & j)
746. checksum = (checksum << 1) | 1;
747. **else**
748. checksum <<= 1;
750. checksum = ~checksum;
751. sprintf(resultado + strlen(resultado), "Checksum Calculado: %.4x\n", checksum);
752. **if**(checksum != ((t[16 + ihl] << 8) + t[17 + ihl]))
753. sprintf(resultado + strlen(resultado), "Error! :(\n");
754. **else**
755. sprintf(resultado + strlen(resultado), "Correcto! :)\n");
756. }
757. **else** **if**(t[23] & 6) //TCP
758. {
759. sprintf(resultado + strlen(resultado), "\n----------TCP----------\n");
760. sprintf(resultado + strlen(resultado), "Source Port: %d\n", (t[14 + ihl] << 8) + t[15 + ihl]);
761. sprintf(resultado + strlen(resultado), "Destination Port: %d\n", (t[16 + ihl] << 8) + t[17 + ihl]);
762. sprintf(resultado + strlen(resultado), "Sequence Number: %d\n", (t[18 + ihl] << 24) + (t[19 + ihl] << 16) + (t[20 + ihl] << 8) + t[21 + ihl]);
763. sprintf(resultado + strlen(resultado), "Acknowledgement Number: %d\n", (t[22 + ihl] << 24) + (t[23 + ihl] << 16) + (t[24 + ihl] << 8) + t[25 + ihl]);
764. unsigned **short** data\_offset = ((t[26 + ihl] >> 4) & 15) \* 4;
765. sprintf(resultado + strlen(resultado), "Data Offset: %d bytes\n", data\_offset);
766. sprintf(resultado + strlen(resultado), "Reserved: %d\n", t[26 + ihl] & 15);
767. **if**(t[27 + ihl] & 128)
768. sprintf(resultado + strlen(resultado), "CWR On\n");
769. **else**
770. sprintf(resultado + strlen(resultado), "CWR Off\n");
772. **if**(t[27 + ihl] & 64)
773. sprintf(resultado + strlen(resultado), "ECN-Echo On\n");
774. **else**
775. sprintf(resultado + strlen(resultado), "ECN-Echo Off\n");
777. **if**(t[27 + ihl] & 32)
778. sprintf(resultado + strlen(resultado), "U On\n");
779. **else**
780. sprintf(resultado + strlen(resultado), "U Off\n");
782. **if**(t[27 + ihl] & 16)
783. sprintf(resultado + strlen(resultado), "A On\n");
784. **else**
785. sprintf(resultado + strlen(resultado), "A Off\n");
787. **if**(t[27 + ihl] & 8)
788. sprintf(resultado + strlen(resultado), "P On\n");
789. **else**
790. sprintf(resultado + strlen(resultado), "P Off\n");
792. **if**(t[27 + ihl] & 4)
793. sprintf(resultado + strlen(resultado), "R On\n");
794. **else**
795. sprintf(resultado + strlen(resultado), "R Off\n");
797. **if**(t[27 + ihl] & 2)
798. sprintf(resultado + strlen(resultado), "S On\n");
799. **else**
800. sprintf(resultado + strlen(resultado), "S Off\n");
802. **if**(t[27 + ihl] & 1)
803. sprintf(resultado + strlen(resultado), "F On\n");
804. **else**
805. sprintf(resultado + strlen(resultado), "F Off\n");
807. sprintf(resultado + strlen(resultado), "Window: %d\n", (t[28 + ihl] << 8) + t[29 + ihl]);
808. sprintf(resultado + strlen(resultado), "Checksum: %.4X\n", (t[30 + ihl] << 8) + t[31 + ihl]);
809. **if**(t[27 + ihl] & 32) //If URG is on
810. sprintf(resultado + strlen(resultado), "Urgent Pointer: %d", (t[32 + ihl] << 8) + t[33 + ihl]);
811. **else**
812. sprintf(resultado + strlen(resultado), "No Urgent Pointer");
814. **if**(data\_offset == 20) //No options
815. sprintf(resultado + strlen(resultado), "\nNo Options");
816. **else**
817. {
818. sprintf(resultado + strlen(resultado), "\nOptions: ");
819. **for**(**BYTE** i = 34 + ihl; i < (34 + ihl) + (data\_offset - 20); i++)
820. {
821. **if**(i == (34 + ihl) + (data\_offset - 20) - 1)
822. sprintf(resultado + strlen(resultado), "%.2X", t[i]);
823. **else**
824. sprintf(resultado + strlen(resultado), "%.2X:", t[i]);
825. }
826. }
827. }
828. **else** **if**(t[23] & 17) //UDP
829. {
830. sprintf(resultado + strlen(resultado), "\n----------UDP----------\n");
831. sprintf(resultado + strlen(resultado), "Source Port: %d\n", (t[14 + ihl] << 8) + t[15 + ihl]);
832. sprintf(resultado + strlen(resultado), "Destination Port: %d\n", (t[16 + ihl] << 8) + t[17 + ihl]);
833. sprintf(resultado + strlen(resultado), "Length: %d\n", (t[18 + ihl] << 8) + t[19 + ihl]);
834. sprintf(resultado + strlen(resultado), "Checksum: %4X\n", (t[20 + ihl] << 8) + t[21 + ihl]);
835. }
836. **return** 1;
837. }
838. **else** **if**(ToT == 2054) //ARP
839. {
840. sprintf(resultado, "ARP\n");
841. **switch**(t[15])
842. {
843. **case** 1:
844. sprintf(resultado + strlen(resultado), "HW: Ethernet\n");
845. **break**;
846. **case** 6:
847. sprintf(resultado + strlen(resultado), "HW: Token Ring\n");
848. **break**;
849. **case** 15:
850. sprintf(resultado + strlen(resultado), "HW: Frame Relay\n");
851. **break**;
852. **case** 16:
853. sprintf(resultado + strlen(resultado), "HW: Atm\n");
854. **break**;
855. }
856. **if**(t[16] == 8)
857. sprintf(resultado + strlen(resultado), "Protocolo: IP\n");
858. **if**(t[18] == 6)
859. sprintf(resultado + strlen(resultado), "Tamanio de la direccion HW: 6 bytes\n");
860. **if**(t[19] == 4)
861. sprintf(resultado + strlen(resultado), "Tamanio de la direccion de Protocolo: 4 bytes\n");
862. **else** **if**(t[19] == 6)
863. sprintf(resultado + strlen(resultado), "Tamanio de la direccion de Protocolo: 6 bytes\n");
865. **switch**(t[21])
866. {
867. **case** 1:
868. sprintf(resultado + strlen(resultado), "Op Code: ARP request\n");
869. **break**;
870. **case** 2:
871. sprintf(resultado + strlen(resultado), "Op Code: ARP response\n");
872. **break**;
873. **case** 8:
874. sprintf(resultado + strlen(resultado), "Op Code: Inverse ARP request\n");
875. **break**;
876. **case** 9:
877. sprintf(resultado + strlen(resultado), "Op Code: Inverse ARP response\n");
878. **break**;
879. }
881. sprintf(resultado + strlen(resultado), "Sender HW address: ");
882. **for**(**BYTE** i = 0; i < t[18]; i++)
883. {
884. **if**(i == t[18] - 1)
885. sprintf(resultado + strlen(resultado), "%.2x", t[22 + i]);
886. **else**
887. sprintf(resultado + strlen(resultado), "%.2x:", t[22 + i]);
888. }
890. sprintf(resultado + strlen(resultado), "\nSender Protocol address: ");
891. **for**(**BYTE** i = 0; i < t[19]; i++)
892. {
893. **if**(i == t[19] - 1)
894. sprintf(resultado + strlen(resultado), "%d", t[22 + i + t[18]]);
895. **else**
896. sprintf(resultado + strlen(resultado), "%d.", t[22 + i + t[18]]);
898. }
900. sprintf(resultado + strlen(resultado), "\nTarget HW address: ");
901. **for**(**BYTE** i = 0; i < t[18]; i++)
902. {
903. **if**(i == t[18] - 1)
904. sprintf(resultado + strlen(resultado), "%.2x", t[22 + i + t[18] + t[19]]);
905. **else**
906. sprintf(resultado + strlen(resultado), "%.2x:", t[22 + i + t[18] + t[19]]);
907. }
909. sprintf(resultado + strlen(resultado), "\nTarget Protocol address: ");
910. **for**(**BYTE** i = 0; i < t[19]; i++)
911. {
912. **if**(i == t[19] - 1)
913. sprintf(resultado + strlen(resultado), "%d", t[22 + i + t[18] + t[19] + t[18]]);
914. **else**
915. sprintf(resultado + strlen(resultado), "%d.", t[22 + i + t[18] + t[19] + t[18]]);
916. }
918. **return** 1;
919. }
921. **return** 0;
922. }