

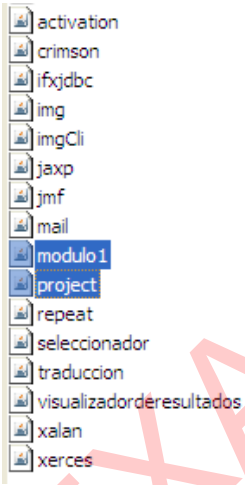
SOLO T'ARTELA

1. *Conociendo al enemigo* 2
2. *Analizando modulo1.jar* 2
3. *Analizando project.jar* 3
4. *La clase crypto* 5
5. *Buscando la clave* 6
6. *Conclusiones* 6
7. *Archivos adjuntos incluidos* 6

1. Conociendo al enemigo

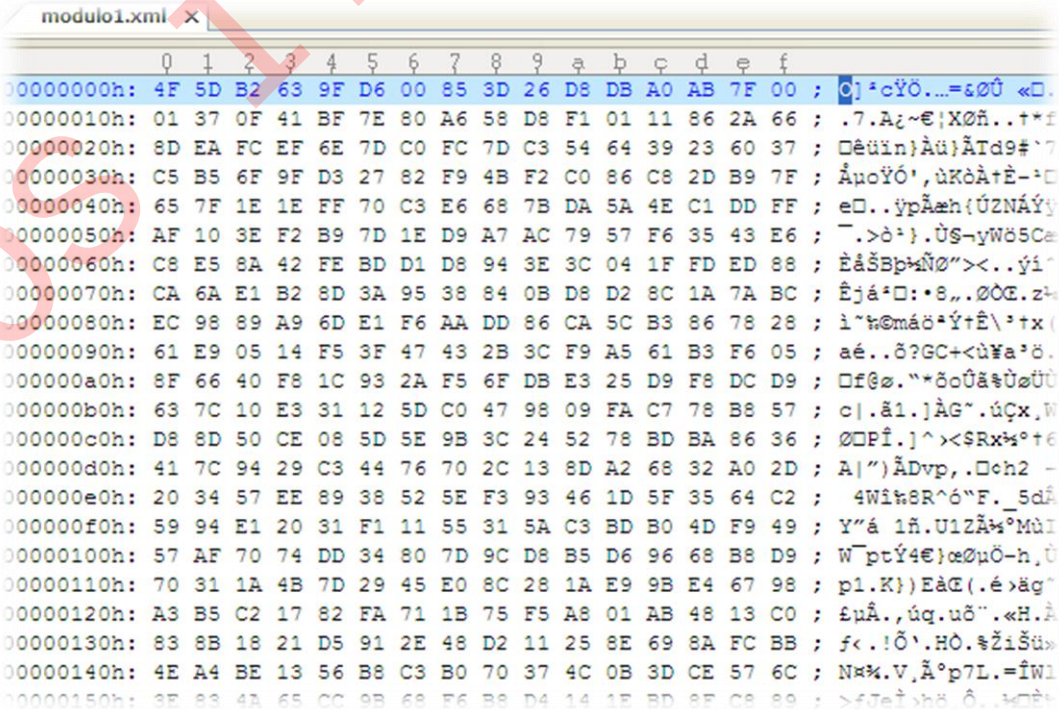
La demo en cuestión está programada en java. Utiliza una serie de librerías propias y trae una versión de java para hacer la demo portable.

Tras indagar un poco en sus entrañas se llega fácil a la conclusión de que las clases principales están dentro de "**project.jar**" y de que la demo encriptada es "**modulo1.jar**".



2. Analizando modulo1.jar

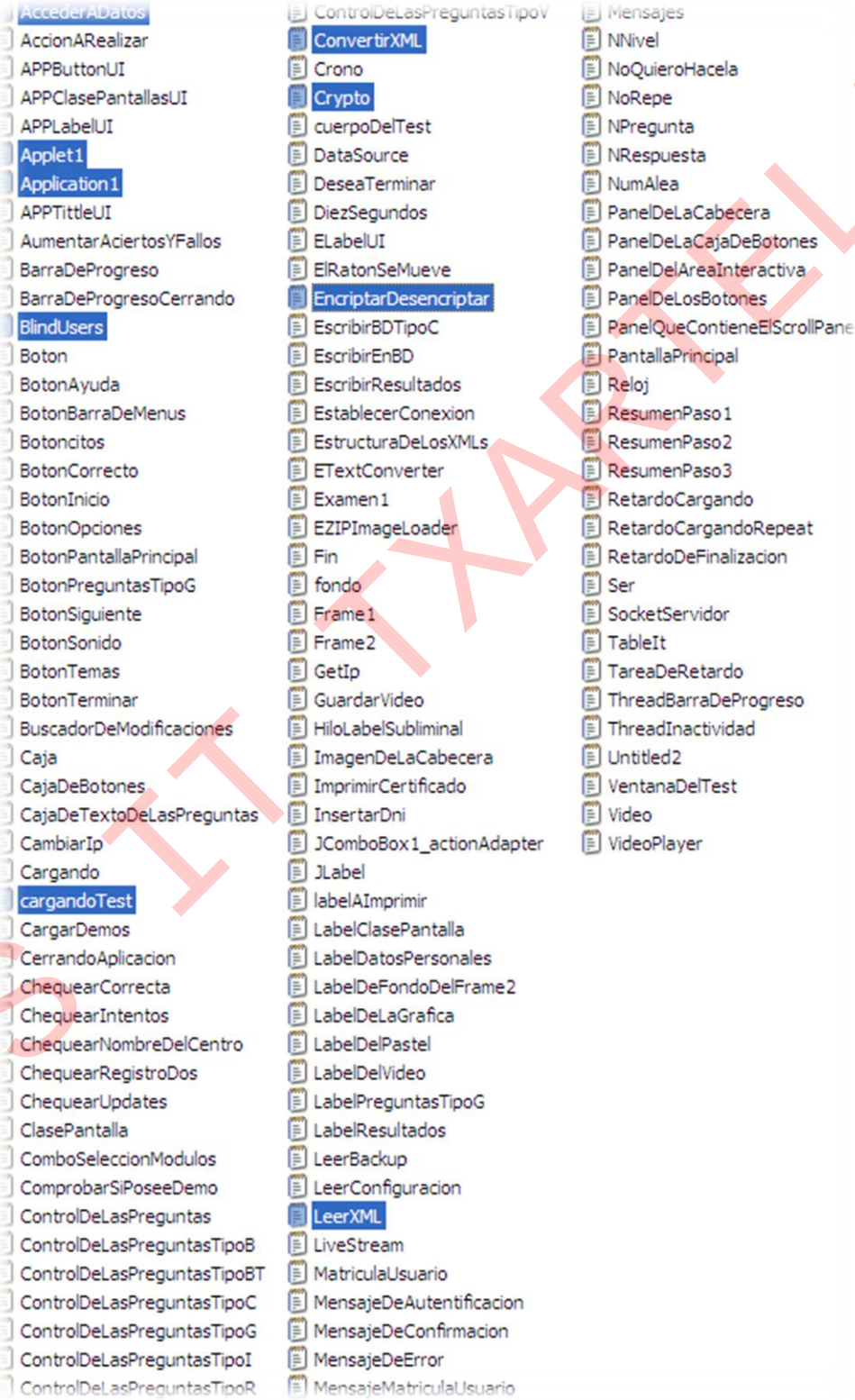
Descomprimimos **modulo1.jar** con 7zip (o similar) y observamos que la encriptación se realiza archivo por archivo, es decir, están codificados por separado tanto el archivo .xml con las preguntas, como las imágenes.



SOLO TXARTELA

3. *Analizando project.jar*

Se pueden ver 128 librerías principales de las que destacan a priori 8.



SOS IT TXARTELA

Antes de adentrarnos en el análisis de las clases veamos el interior del archivo "**application.properties**" que es el archivo de opciones.

Código de "application.properties"

```
h.idiomaPrimario=1
h.idiomaSecundario=
h.idiomas=1
h.controlarIdioma="No"
h.proyecto=CastillaLaMancha
h.tituloDateSelector=Seleccione una fecha
h.etituloDateSelector=Aukera ezazu egun bat
h.textoVersionC=x
h.textoVersionE=x
h.version=by deurus
h.demo=No
h.demos=Sí
h.controlIP=No
h.modulosDemo=1;2;17;18;21;22;24;25;26;39;58;
h.time=5
h.aleatorio=No
h.imprimir=Sí
h.repetir=Sí
h.camino=Sí
h.mostrarTodas=Sí
h.localizacion=Sí
h.dosIdimomas=No
h.seleccionDeIdiomas=Sí
h.repeat=Sí
h.direccionReclamacionRepeat=sendMail.jsp
h.toReclamacionRepeat=certifications@esi.es
h.apendiceReclamacion=IT Txartela
h.administrador=
h.porcentCompCon20030317=Sí
h.repeatIvap=Sí
h.nota=Sí
h.ventanaOpciones=Sí
h.controlarUpdates=Sí
h.controlarNombreCentro=Sí
h.checkBlindUsers=No
h.encriptar=No
h.desencriptarModulos=Sí

h.proxyAddress=http://guc.esi.es/guccmproxy/
h.repeatAddress=http://guc.esi.es/guccmadmin/

#h.proxyAddress=http://developguc1.esi.es/guccmproxy/
#h.repeatAddress=http://developguc1.esi.es/guccmadmin/

#h.proxyAddress=http://localhost/guccmproxy/
#h.repeatAddress=http://localhost/guccmadmin/

h.link=
h.GUCLink=http://guc.esi.es/guccmfull/
h.iconoInferiorPantallaPrincipal=No
h.proxySet=false
h.proxyHost=
h.proxyPort=
```

SOS IT TXARTELA

Ahora pasamos a analizar la clase principal "**Application1.class**". Lo primero la decompilamos con nuestro decompilador favorito, yo usé DJ java decompiler. En realidad decompilamos todas las clases por comodidad.

Tras echar un vistazo vemos algo interesante en "**LeerConfiguracion.class**". Vemos que se utiliza la clase "crypto" para desencriptar el módulo.

```
fichero = "/project/modulo" + modulo + ".xml";
System.out.println("fichero: " + fichero);
stream = Application1.crypto.decrypt((java.lang.Object.class).getResourceAsStream(fichero));
```

Función **decrypt** de la clase **crypto**:

```
public InputStream decrypt(InputStream in)
{
    if(AccederADatos.leer("h.desencriptarModulos").compareToIgnoreCase("S\355") == 0)
        try
        {
            in = new CipherInputStream(in, dcipher);
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    return in;
}
```

En sucesivos análisis del resto de código la clase "**crypto**" aparece recurrentemente, por lo que pasamos a analizarlo más detalladamente.

4. La clase crypto

En esta librería obtenemos información importante para poder revertir la encriptación. Destacamos los datos recopilados.

- Tipo de encriptación: **PBEWithMD5AndDES**
- Iteración: **19**
- Salt: **-87, -101, -56, 50, 86, 53, -29, 3**

Con estos datos tenemos casi todo lo importante, ahora solo necesitamos la clave que se usa para desencriptar.

SOS IT TXARTELA

5. Buscando la clave

Indagando en el código fuente vemos varias referencias a posibles contraseñas que destacamos a continuación.

Application1.class (crea la instancia)

- crypto = new Crypto(AccederADatos.decryptPassword);

AccederADatos.class (declara la variable)

- public static String decryptPassword = "";

Applet1.class (possible contraseña)

- AccederADatos.decryptPassword = "ESI";

Ser.class (possible contraseña)

- Crypto crypto = new Crypto("nunca sin mi permiso");

Como posteriormente comprobaremos, la clave correcta es "**nunca sin mi permiso**". Que por cierto es exactamente lo que dice "Leelo" (Milla Jovovich) en "El Quinto Elemento" a "Korben Dallas" (Bruce Willis).

6. Conclusiones

La clase "**Ser.class**" que contiene la clave correcta, no es referenciada por ninguna de las clases principales, por lo que deduzco que es una clase que no debería estar ya que no se utiliza y se olvido de borrar el creador.

En realidad la clave deberíamos de haberla conseguido por fuerza bruta ya que la desencriptación se realiza a través de una web especialmente creada "**decriptData.jsp**" alojada en el Proxy que vimos en el archivo de configuración, es decir, va desencriptando los caracteres desde **http://guc.esi.es/guccmproxy/decriptData.jsp**. Esta web es la madre del cordero ya que cuando la pusieron fuera de servicio, dejó de funcionar la demo.

7. Archivos adjuntos incluidos

- Conjunto de clases decompiladas.
- Código fuente del desencriptador en Java.
- Desencriptador de ficheros.
- Módulo encriptado.
- Módulo desencriptado.

Enlace DEMO operativa: http://ul.to/mlrc7hwq

By deurus

SOS IT TXARTELA