



URL	http://www.screamingfrog.co.uk/
Victima	ScreamingFrogSEOSpider-5.1
Herramientas	JdGui , Netbeans , (Ctrl-c , Ctrl-v)
Cracker	Alberto Fernández
Dificultad	Un poco complicado
Fecha	27 – 10 - 2015

Hola, como hace tanto tiempo que no escribo nada, haremos un Keygen.

Lo primero que hacemos, es abrirlo con el jdGui, cuando lo tengamos abierto, buscamos la referencia a licence o algo parecido.

Lo más parecido que encontramos es LicenceFileUI.class, que está en el paquete :

```
package seo.spider.ui;
```

LicenceFileUI.class : Es la clase que construye el formulario.

Las variables que nos interesan de esta clase son:

```
private final b mLicenceFileValidator; clase que comprueba la licencia
```

```
private final JTextField mUsername; ↔ Usuario
```

```
private final JTextField mLicenceKey; ↔ N° licencia
```

Tanto la clase **b** como la **c** están en el paquete [seo.spider.i](#)

```
c localc = new c(this.mUsername.getText(), this.mLicenceKey.getText());
```

La clase: **c** , descompone en dos cadenas el username y LicenceKey.

mUsername lo coloca en **a** y **mLicenceKey** en **b**.

En la clase : **b**, en jdGui, seleccionamos todo, pulsamos Ctrl-c y en la siguiente linea de este documento Ctrl-v.

```
package seo.spider.i;
```

```
import java.util.HashSet;
```

```
import java.util.Set;
```

```

import org.apache.log4j.Logger;

public class b
{
    private static final Logger a = Logger.getLogger(b.class.getName());

    public static long a()
    {
        Calcula la fecha actual
        long l = System.currentTimeMillis() / 1000L;
        Si la fecha actual es menor de 1439277087 devuelve este último valor
        if (l < 1439277087L)
        {
            a.warn("Clock out, correcting from: " + l);
            l = 1439277087L;
        }

        return l;
    }

    public final d b()
    {
        a locala = new a();
        c localc = locala.b();
        return a(localc);
    }
Comprueba que la fecha actual sea menor que la fecha fija
    private boolean a(long paramLong1, long paramLong2)
    {
        return paramLong2 < paramLong1;
    }
Comprueba que la fecha actual sea menor que la suma de (1439277087 + 34128000)
    private boolean b(long paramLong1, long paramLong2)
    {
        return paramLong1 < paramLong2 + 34128000L;
    }
Comprueba que la clave que introduces no sean idénticas a las que hay aquí, ya que si es idéntica es invalida.
    private boolean a(String paramString)
    {
        HashSet localHashSet = new HashSet();
        localHashSet.add(new StringBuffer("0C4796AEF7-5765006731-5BAA830553").reverse().toString());
        localHashSet.add(new StringBuffer("EFEF0B830F-2644397731-25EE7C3152").reverse().toString());
        localHashSet.add(new StringBuffer("4F1E936AD2-0040991931-3760F5BBAB").reverse().toString());
        localHashSet.add(new StringBuffer("711F946071-5921512831-E812E9FE98").reverse().toString());
        localHashSet.add(new StringBuffer("A6476EE9D3-9059895041-52BDEDCAC7").reverse().toString());
        localHashSet.add(new StringBuffer("2B7C4F8485-1976016141-355C74D7DD").reverse().toString());

        localHashSet.add(new StringBuffer("D30D5095C7-4884318241-4A59B064E5").reverse().toString());
        localHashSet.add(new StringBuffer("7169C1E896-3098218241-53A119ABE5").reverse().toString());
        localHashSet.add(new StringBuffer("A2894D6D56-0294318241-CE93DABB36").reverse().toString());
        localHashSet.add(new StringBuffer("FE4A3AF8F5-2994318241-64B1C74F4F").reverse().toString());
        localHashSet.add(new StringBuffer("BE63EE6B00-4205318241-96BA2C9D32").reverse().toString());
        localHashSet.add(new StringBuffer("FF6348A736-8505318241-D44F2AC703").reverse().toString());
        localHashSet.add(new StringBuffer("67E7E11B8D-6415318241-E57D3BD13A").reverse().toString());
        localHashSet.add(new StringBuffer("BF40BB3141-4715318241-3E31E9A48C").reverse().toString());
        localHashSet.add(new StringBuffer("86579C58A9-6135318241-E818C00461").reverse().toString());
        localHashSet.add(new StringBuffer("2592D3526E-6298218241-693E014DEF").reverse().toString());
    }

```

```

localHashSet.add(new StringBuffer("75842B1B52-8465318241-B806CC499A").reverse().toString());
localHashSet.add(new StringBuffer("40031DC7E4-8625847541-FF590F74FD").reverse().toString());

localHashSet.add(new StringBuffer("5827655FAC-1153179241-6E5E7AFB19").reverse().toString());
localHashSet.add(new StringBuffer("2EACCC13A1-5743179241-0CEA9003DB").reverse().toString());
localHashSet.add(new StringBuffer("91BFA7E5E6-1429713511-CAF5567285").reverse().toString());

localHashSet.add(new StringBuffer("583A28F78F-0046061541-B9FC1A365A").reverse().toString());

localHashSet.add(new StringBuffer("DACE3ADF0D-9661045641-A94D890914").reverse().toString());
localHashSet.add(new StringBuffer("A850569D0C-4581045641-7B1D710522").reverse().toString());
localHashSet.add(new StringBuffer("13EF2C2BD4-3791045641-3DF4BEB175").reverse().toString());

localHashSet.add(new StringBuffer("A004F200D6-3181847541-A5D999F949").reverse().toString());

localHashSet.add(new StringBuffer("7C5BC19F63-2579041641-FA3D788AF6").reverse().toString());
localHashSet.add(new StringBuffer("890FBEDB9C-9690141641-63573EBF56").reverse().toString());
localHashSet.add(new StringBuffer("C704DAC7A0-6959041641-84DA736562").reverse().toString());
localHashSet.add(new StringBuffer("3CC314A794-2169041641-0240F533A9").reverse().toString());
localHashSet.add(new StringBuffer("FF79DE6942-2269041641-CFDF22F9F1").reverse().toString());
localHashSet.add(new StringBuffer("522B21CFAF-8369041641-E49F84A993").reverse().toString());
localHashSet.add(new StringBuffer("2C7350EE23-8469041641-59D3927834").reverse().toString());
localHashSet.add(new StringBuffer("1E165F75FF-2669041641-CC3C0A1501").reverse().toString());
localHashSet.add(new StringBuffer("5F416DDAC5-2869041641-51A5390877").reverse().toString());
localHashSet.add(new StringBuffer("CD3F98F437-8379041641-6A6DBD64BA").reverse().toString());

localHashSet.add(new StringBuffer("5E893CF2A1-8307531741-664A752AB3").reverse().toString());

return !localHashSet.contains(paramString);
}

```

Esta es la parte que nos interesa.

```

public final d a(c paramc) paramc = ("mUsername", "mLicenceKey")
{
    d locald = new d();

```

Descompone paramc.b() que es mLicenceKey en tres cadenas.

```

String[] arrayOfString = paramc.b().split("-");

```

```

if (arrayOfString.length == 3)
{
    String str1 = arrayOfString[0].toUpperCase(); Primera subcadena
    String str2 = arrayOfString[1]; Segunda subcadena que es la fecha
    String str3 = arrayOfString[2].toUpperCase(); Tercera subcadena

```

Cadena tomada del reverso

```

String str4 = new StringBuffer("F2sM2kCet8vxNtC0Pupk- 41a5paIIPf8zbm_8MF").reverse().toString();

```

Clase para calcular SHA-1

```

seo.spider.util.b localb = new seo.spider.util.b();

```

Calcula el SHA-1 de mUsername + str2 + str4

```

String str5 = localb.b(paramc.a() + str2 + str4);

```

Comprueba que el largo resultante no sea inferior a 40 y si lo es le añade un 0 al inicio

```

while (str5.length() < 40)
{
    str5 = "0" + str5;
}

```

Comprueba que el largo es igual a 40

```

if (str5.length() == 40)
{

```

Crea una subcadena con los 10 primeros caracteres de la cadena resultante de SHA-1, colocandolos en

mayúsculas

```
String str6 = str5.substring(0, 10).toUpperCase();
```

Crea una subcadena con los 10 últimos caracteres de la cadena resultante de SHA-1, colocandolos en mayúsculas

```
String str7 = str5.substring(30, 40).toUpperCase();
```

Convierte la cadena de la fecha en valor long

```
long l1 = b(str2);
```

Comprueba que la fecha actual sea menor a < 1439277087L , si es mayor devuelve 1439277087

```
long l2 = a();
```

Comprueba que la str6 es igual a str1 y str7 es igual a str3 y l1 es igual a l2 y fecha actual sea menor que el valor de l2 + 34128000L

```
int i = (str6.equals(str1)) && (str7.equals(str3)) && (b(l1, l2)) && (a(paramc.b())) ? 1 : 0;
```

```
if (i != 0)
```

```
{
```

```
    locald.a(l1);
```

```
    if (a(l1, l2))
```

```
    {
```

```
        locald.a();
```

```
    }
```

```
}
```

```
}
```

```
}
```

```
return locald;
```

```
}
```

Convierte la cadena de fecha en long

```
private long b(String paramString)
```

```
{
```

```
    long l = 0L;
```

```
    try
```

```
    {
```

```
        l = Long.parseLong(paramString);
```

```
    }
```

```
    catch (NumberFormatException localNumberFormatException)
```

```
    {
```

```
        a.warn(localNumberFormatException.toString(), localNumberFormatException);
```

```
        l = 0L;
```

```
    }
```

```
    return l;
```

```
}
```

```
}
```

Clase util.b que nos servirá para el keygen, con alguna variación.

```
package seo.spider.util;
```

```
import java.math.BigInteger;
```

```
import java.security.MessageDigest;
```

```
import java.security.NoSuchAlgorithmException;
```

```
import org.apache.log4j.Logger;
```

```
public class b
```

```
{
```

```
    private static final Logger a = Logger.getLogger(b.class.getName());
```

```
    public final String a(String paramString)
```

```

{
    return a(paramString, "MD5");
}

public final String b(String paramString)
{
    return a(paramString, "SHA-1");
}

private final String a(String paramString1, String paramString2)
{
    String str = "";
    try
    {
        MessageDigest localMessageDigest = MessageDigest.getInstance(paramString2);
        localMessageDigest.update(paramString1.getBytes(), 0, paramString1.getBytes().length);
        str = new BigInteger(1, localMessageDigest.digest()).toString(16);
    }
    catch (NoSuchAlgorithmException localNoSuchAlgorithmException)
    {
        a.warn(localNoSuchAlgorithmException.toString(), localNoSuchAlgorithmException);
    }
    return str;
}
}

```

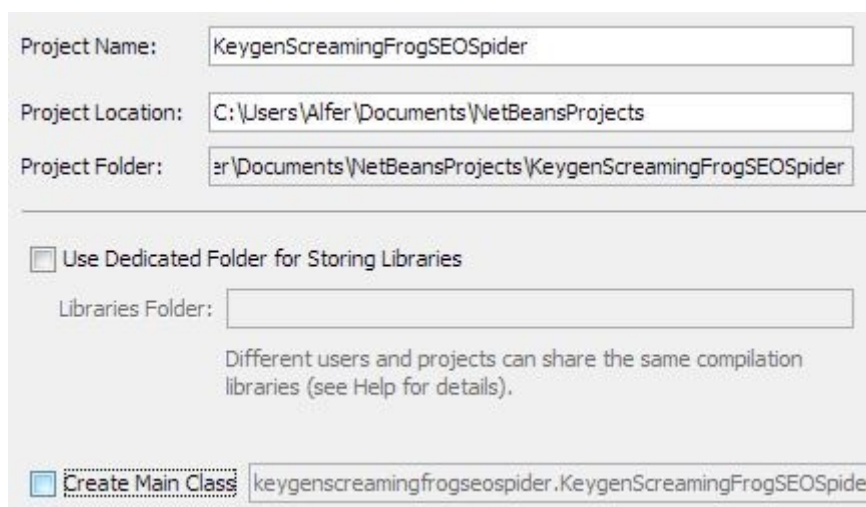
Y ahora construiremos el Keygen.

Lo primero de todo lanzar el Netbeans.

Cuando esté abierto creamos la nueva aplicación.

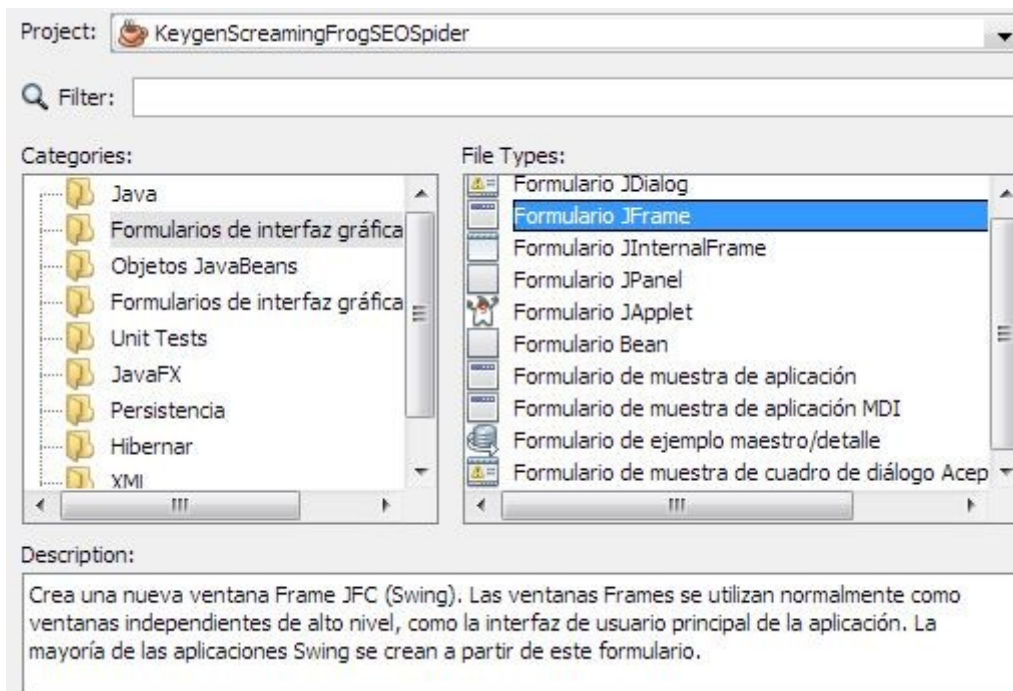


Le damos a siguiente:



Des tildamos Create Main Class, y pulsamos en Terminar.

En Paquete de fuentes, botón derecho de ratón / nuevo / Otro seleccionamos:



Pulsamos siguiente

Nombre de Clase:

Proyecto:

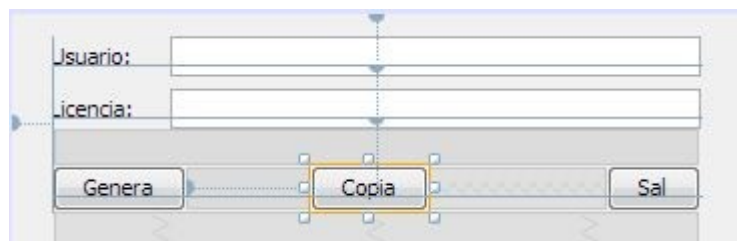
Ubicación:

Paquete:

Crear Archivo:

le damos a terminar.

En el frame que tenemos colocaremos 2 `javax.swing.JTextField`, 2 `javax.swing.JLabel`, 3 botones. Bueno la diferencia entre versiones anteriores a la 3 y esta, es que el valor 1439277087L es diferente.



A `javax.swing.JTextField`1 y `javax.swing.JTextField`2 le cambiaremos el nombre por `mUsername` y `mLicenceKey`, respectivamente. Para los botones les cambiamos a genera, copia y sal.

En botón Genera, lo seleccionamos y botón derecho del ratón, Eventos / Action / `actionPerformed`. Lo mismo para todos los botones.


```

private void generaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void copiaActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void salActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

```

En:

```

private void salActionPerformed(java.awt.event.ActionEvent evt) {
    // Cerrar la aplicación
    System.exit(0);
}

```

En:

```

private void copiaActionPerformed(java.awt.event.ActionEvent evt) {
    String serial = mLicenceKey.getText();
    StringSelection strSelection = new StringSelection(serial);
    Clipboard clpbd = Toolkit.getDefaultToolkit().getSystemClipboard();
    clpbd.setContents(strSelection, null);
}

```

En:

```

private void generaActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        String str = new StringBuffer("F2sM2kCet8vxNtC0Pupk- 41a5paIipF8zbm_8MF").reverse().toString();
        String texto = "";
        Long antesCadena2 = 0L;
        long l = 0L;
        long f = 1439277087L;

        // fecha actual en segundos
        l = System.currentTimeMillis() / 1000L;

        //Mayor de l y f, y le suma 34128000
        antesCadena2 = (Math.max(l, f) + 34128000L);
        // convierte a cadena cogiendo un máximo de 10 caracteres
        texto = String.valueOf(antesCadena2).substring(0, 10);
        //recoge el nombre del usuario
        String text2 = this.mUsername.getText();
        //crea una cadena con el usuario+texto+str
        String text31 = (text2 + texto + str);
        //comprueba que la cadena resultante no sea inferior a 40 de largo
        //si es inferior le añade un cero al principio
        while(text31.length() < 40){
            text31 = "0" + text31;
        }
        //convierte a sha-1
        util.b localb = new util.b();
        String text41 = localb.b(text31);
        //selecciona desde la posición 0 a 10
        String text51 = text41.substring(0, 10);
        //selecciona desde la posición 30 a 40
        String text61 = text41.substring(30, 40);
    }
}

```

```

        //crea el serial, text51 en mayúsculas - texto - text61 en mayúsculas
        this.mLicenceKey.setText(text51.toUpperCase() + '-' + texto + '-' + text61.toUpperCase());
    }catch(Exception e){
        e.getMessage();
    }
}

```

Crearemos, desde el paquete de fuentes, botón derecho del ratón nuevo paquete con nombre “util” y una nueva clase “b”. en la clase b colocaremos el contenido de la clase b del programa con alguna modificación que no la necesitamos. Y nos quedará de la siguiente manera.

package util;

```

import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

```

```

/**
 *
 * @author Alfer
 */
public class b {
    public final String a(String paramString){
        return a(paramString, "MD5");
    }

    public final String b(String paramString){
        return a(paramString, "SHA-1");
    }

    private final String a(String paramString1, String paramString2){
        String str = "";
        try{
            MessageDigest localMessageDigest = MessageDigest.getInstance(paramString2);
            localMessageDigest.update(paramString1.getBytes(), 0, paramString1.getBytes().length);
            str = new BigInteger(1, localMessageDigest.digest()).toString(16);
        }catch (NoSuchAlgorithmException localNoSuchAlgorithmException){
            localNoSuchAlgorithmException.getMessage();
        }
        return str;
    }
}

```

```

public long a(){
    long l = System.currentTimeMillis() / 1000L;
    if(l < 1439277087L){
        l = 1439277087L;
    }
    return l;
}

```

En Netbeans, Menú Run / Build Project o F11 , después a F6, cuando se ejecute, introducimos el nombre y le damos a generar.

Cuando ya se halla generado, tenemos dos opciones para registrarlo:

Opción 1 : ir a la ruta C:\Users\NombreUsuario\.ScreamingFrogSEOSpider seleccionar Licence.txt e

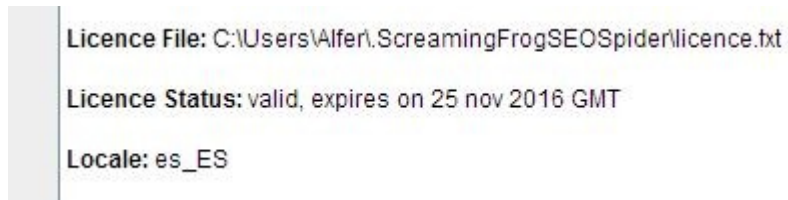
introducir los valores.

Primera linea: NombreUsuario

Segunda linea: Licencia.

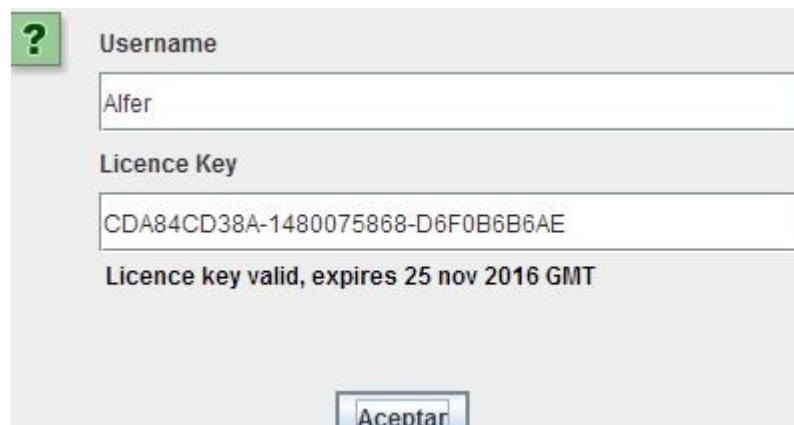
Opción 2 : ejecutar el programa e introducir los datos.

Para Comprobar si es correcto, ejecutamos el programa, en el menú vamos a Help y despues a debug.



Licence File: C:\Users\Alfer\ScreamingFrogSEOSpider\licence.txt
Licence Status: valid, expires on 25 nov 2016 GMT
Locale: es_ES

Licence y Enter Licence Key.



? Username
Alfer
Licence Key
CDA84CD38A-1480075868-D6F0B6B6AE
Licence key valid, expires 25 nov 2016 GMT
Aceptar

Y con esto es todo.

27 – 10 – 2015.

Alberto Fernández.