REVERSE ENGINEERING ASSIGNMENT

Thomas Raphel
tr1147@nyu.edu
N 17107219

WHAT IS REVERSE ENGINEERING:

Software reverse engineering is done to retrieve the source code of a program because the source code was lost, to study how the program performs certain operations, to improve the performance of a program, to fix a bug (correct an error in the program when the source code is not available), to identify malicious content in a program such as a virus or to adapt a program written for use with one microprocessor for use with another. Reverse engineering for the purpose of copying or duplicating programs may constitute a copyright violation. In some cases, the licensed use of software specifically prohibits reverse engineering.—-(http://searchsoftwarequality.techtarget.com/definition/reverse-engineering)

Reverse-engineering is especially important with computer hardware and software. Programs are written in a language, say C++ or Java, that's understandable by other programmers. But to run on a computer, they have to be translated by another program, called a compiler, into the ones and zeros of machine language. Compiled code is incomprehensible to most programmers, but there are ways to convert machine code back to a more human-friendly format, including a software tool called a decompiler.

WHAT IS DECOMPILING:

A decompiler is a computer program that performs the reverse operation to that of a compiler. That is, it translates program code at a relatively low level of abstraction (usually designed to be computer readable rather than human readable) into a form having a higher level of abstraction (usually designed to be human readable). Decompilers usually do not perfectly reconstruct the original source code, and can vary widely in the intelligibility of their outputs. Nonetheless, decompilers remain an important tool in software reverse engineering. A decompiler takes as input an executable file, and attempts to create a high level, compilable, possibly even maintainable source file that does the same thing. It is therefore the opposite of a compiler, which takes a source file and makes an executable. However, a general decompiler does not attempt to reverse every action of the compiler, rather it transforms the input program repeatedly until the result is high level source code.———WIKI DEFINITION.

SELECTION OF BINARY EXECUTABLE FILE:

Since Java was the language i was comfortable in,i had narrowed the search field accordingly and got the below mentioned one.

Search for crackme

search

Advanced search »

Recently discussed

borismilner: 4N006135 - Level 4

borismilner: 4N006135

Coderess: JCrackme#1

borismilner: 4N006135 - Level 5

monads: Facets

otmanov: CrackeMeby °Designer Shoes°

Kwisatz Haderach: berkeley

Various numbers

Users: 70060
Crackmes: 2927
Solutions: 4041

CRACKMES.DE ARCHIVE

- title/description of the crackme (or a part of it)
2 - Needs a little brain (or luck) - dificulty level
Any platform - what platform is the crackme running on?
Java - what language is it written in?
All crackmes - which crackmes are you interested in?

Most recent first - sorting order

Search

SEARCH RESULTS

[1..3 of 3]

JCrackme#1 by Coderess
Published: 13. Sep, 2015
Difficulty: 2 - Needs a little brain (or luck)
Platform: Windows
Language: Java

Solved by klefz, draww

GenMe by CRY971C
Published: 22. Jul, 2008
Difficulty: 2 - Needs a little brain (or luck)
Platform: Windows
Language: Java

Solved by obnoxious, xylitol

Java CrackMe #3 by vhly
Published: 22. Jan, 2006
Difficulty: 2 - Needs a little brain (or luck)
Platform: Unspecified/other
Language: Java

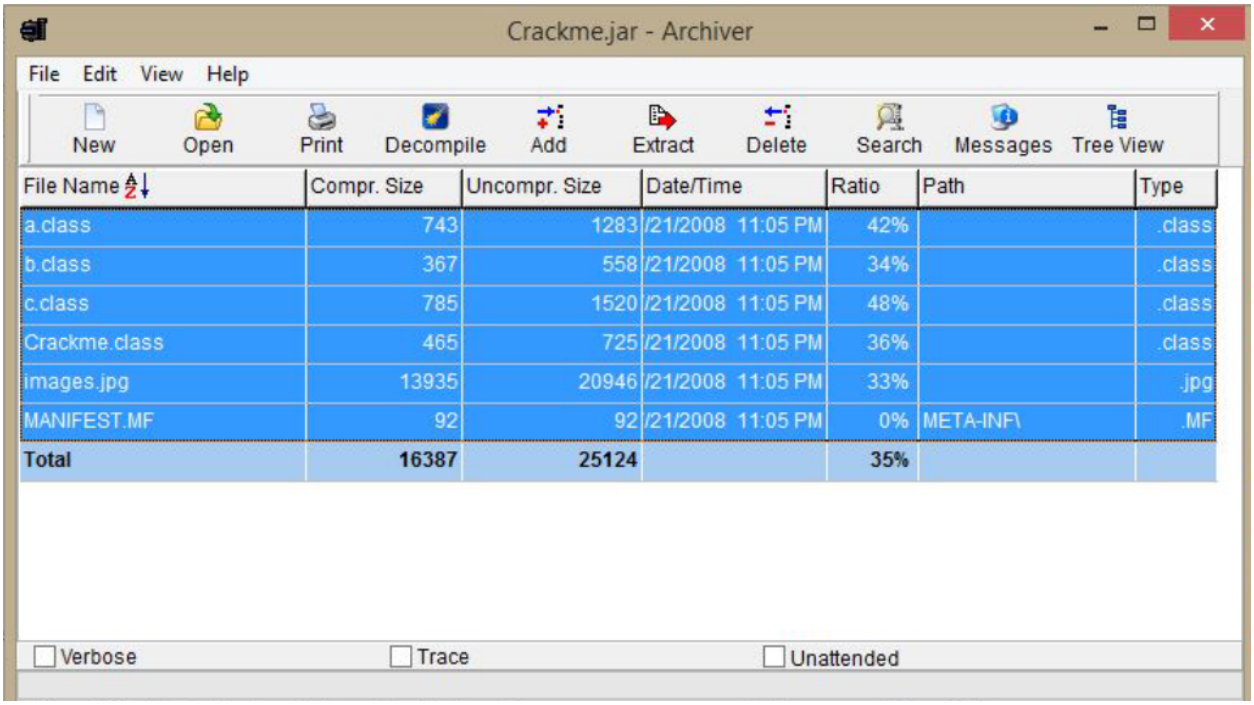
Solved by Kerberos

The binary file details is as given below:

GenMe by CRY971C
Publlished : July 22,2008
Platform : Windows
Language : Java

SELECTION OF DECOMPILER:

With DJ Java Decompiler you can decompile java class-files and save it in text or other format. It's simple and easy. Compatible with Windows 7 , Windows XP, Windows 2003, Windows Vista, Windows 7, Windows 8 decompiler and disassembler for Java that reconstructs the original source code from the compiled binary CLASS files (for example Java applets). DJ Java Decompiler is able to decompile complex Java applets and binaries, producing accurate source code. It lets you quickly obtain all essential information about the class files. It might be easy to decompile the Java files with DJ Java Decompiler.



File Name	Compr. Size	Uncompr. Size	Date/Time	Ratio	Path	Type
a.class	743	1283	/21/2008 11:05 PM	42%		.class
b.class	367	558	/21/2008 11:05 PM	34%		.class
c.class	785	1520	/21/2008 11:05 PM	48%		.class
Crackme.class	465	725	/21/2008 11:05 PM	36%		.class
images.jpg	13935	20946	/21/2008 11:05 PM	33%		.jpg
MANIFEST.MF	92	92	/21/2008 11:05 PM	0%	META-INF\	.MF
Total	**16387**	**25124**		**35%**		

After decompiling the files obtained were :
1)Crackle.jad
2)a.jad
3)b.jad
4)c.jad ,,,,,along with an image file that is used to set the background of the dialog box where key is being asked.

After looking into all the 4 java files that we got after decompiling,it was evident that c.java contained only the code for swing GUI,it had nothing related to the logic in it.

b.java had 2 strings and 1 string array declared inside it.These were used on the logic of password creation.

a.java was the file that contained the entire logic of the crack ,hence password creation.

The algorithm stated below was used for the password creation based on manipulations on the input string.

The crackle.java file contained the main function and initializes the code output.

The important portions of all the java files are shown as screenshots below.

IMPORTANT PART OF CODE FOR PASSWORD GENERATION:

The part of code in class a was crucial to find out how password was generated.

CODE:

```
public final void actionPerformed(ActionEvent actionevent)
   {
       actionevent = new b();
       a_c_fld.b_java_lang_String_fld = a_c_fld.a.getText();
       a_c_fld.c = a_c_fld.b_javax_swing_JTextField_fld.getText();
       a_c_fld.c;
       String s = a_c_fld.b_java_lang_String_fld;
       actionevent = actionevent;
       actionevent.key = Integer.toString(s.length() * 1337);
       actionevent.a_java_lang_String_fld = "";
       for(int i = 0; i <= 25; i++)
       {
           if(i == ((b) (actionevent)).key.length())
               break;
           actionevent.a_java_lang_String_fld = ((b)
(actionevent)).a_java_lang_String_fld.concat(((b)
(actionevent)).a_java_lang_String_array1d_fld[Integer.parseInt(((b)
(actionevent)).key.substring(i, i + 1))]);
       }

       break MISSING_BLOCK_LABEL_139;
_L1:
       ((b) (actionevent)).a_java_lang_String_fld;
       break MISSING_BLOCK_LABEL_146;
       JFrame jframe;
       jframe = a_javax_swing_JFrame_fld;
       String s1 = ((b) (actionevent)).a_java_lang_String_fld;
         goto _L1
       equals();
       JVM INSTR ifeq 162;
          goto _L2 _L3
_L2:
       break MISSING_BLOCK_LABEL_152;
```

```
_L3:
        break MISSING_BLOCK_LABEL_162;
        JOptionPane.showMessageDialog(jframe, "Good job, now write a guide and code a
keygen!!", "YES!!!", -1);
        return;
        JOptionPane.showMessageDialog(jframe, "Nope, try again!", "Try Again", 0);
        return;
    }

    private JFrame a_javax_swing_JFrame_fld;
    private c a_c_fld;
}
```

```java
final class b
{

    public b()
    {
    }

    String key;
    String a_java_lang_String_fld;
    String a_java_lang_String_array1d_fld[] = {
        "a", "b", "c", "d", "e", "f", "g", "h", "i", "j",
        "k", "l", "m", "n", "o", "p", "q", "r", "s", "t",
        "u", "v", "w", "x", "y", "z"
    };
}
```

```java
public final class c extends JPanel
{

    public c(JFrame jframe)
    {
        setLayout(new GridLayout(2, 1));
        a_javax_swing_ImageIcon_fld = new ImageIcon(getClass().getResource("images.jpg"));
        a_javax_swing_JLabel_fld = new JLabel(a_javax_swing_ImageIcon_fld);
        b_javax_swing_JLabel_fld = new JLabel("Enter Name:  ");
        c_javax_swing_JLabel_fld = new JLabel("Enter Serial:  ");
        b_javax_swing_JTextField_fld = new JTextField(12);
        a_javax_swing_JTextField_fld = new JTextField(12);
        button = new JButton("VERIFY");
        a_javax_swing_JPanel_fld = new JPanel();
        a_javax_swing_JPanel_fld.setPreferredSize(new Dimension(300, 150));
        button.setPreferredSize(new Dimension(300, 90));
        button.addActionListener(new a(this, jframe));
        a_javax_swing_JPanel_fld.add(b_javax_swing_JLabel_fld);
        a_javax_swing_JPanel_fld.add(a_javax_swing_JTextField_fld);
        a_javax_swing_JPanel_fld.add(c_javax_swing_JLabel_fld);
        a_javax_swing_JPanel_fld.add(b_javax_swing_JTextField_fld);
        a_javax_swing_JPanel_fld.add(button);
        add(a_javax_swing_JLabel_fld);
        add(a_javax_swing_JPanel_fld);
    }

    private JPanel a_javax_swing_JPanel_fld;
    private JLabel a_javax_swing_JLabel_fld;
    private JLabel b_javax_swing_JLabel_fld;
    private JLabel c_javax_swing_JLabel_fld;
    private ImageIcon a_javax_swing_ImageIcon_fld;
    JTextField a_javax_swing_JTextField_fld;
    JTextField b_javax_swing_JTextField_fld;
    private JButton button;
    String b_java_lang_String_fld;
    String c_java_lang_String_fld;
}
```

```
public final void actionPerformed(ActionEvent actionevent)
{
    actionevent = new b();
    a_c_fld.b_java_lang_String_fld = a_c_fld.a.getText();
    a_c_fld.c = a_c_fld.b_javax_swing_JTextField_fld.getText();
    a_c_fld.c;


            /*Key calculation starts here*/

    String s = a_c_fld.b_java_lang_String_fld; /*Takes the input string*/
    actionevent = actionevent;
    actionevent.key = Integer.toString(s.length() * 1337);/*key=name.length * 1337 */
    actionevent.a_java_lang_String_fld = "";
    for(int i = 0; i <= 25; i++)
    {
        if(i == ((b) (actionevent)).key.length())
            break;
        actionevent.a_java_lang_String_fld = ((b) (actionevent)).a_java_lang_String_fld.concat(((b) (actionevent)).a_java_lang_String_array1d_fld[In
    }
            /*calculation ligic ends here*/


    break MISSING_BLOCK_LABEL_139;
_L1:
    ((b) (actionevent)).a_java_lang_String_fld;
    break MISSING_BLOCK_LABEL_146;
    JFrame jframe;
    jframe = a_javax_swing_JFrame_fld;
    String s1 = ((b) (actionevent)).a_java_lang_String_fld;
      goto _L1
    equals();
    JVM INSTR ifeq 162;
       goto _L2 _L3
_L2:
    break MISSING_BLOCK_LABEL_152;
_L3:
    break MISSING_BLOCK_LABEL_162;
    JOptionPane.showMessageDialog(jframe, "Good job, now write a guide and code a keygen!!", "YES!!!", -1);
    return;
    JOptionPane.showMessageDialog(jframe, "Nope, try again!", "Try Again", 0);
    return;
}
```

ALGORITHM: The algorithm was deduced from the last screenshot.Its been explained further below

1)A String array was created to store all the 26 characters

```
String a_java_lang_String_array1d_fld[] = {
                                          "a", "b", "c", "d", "e", "f", "g", "h", "i", "j",
                                          "k", "l", "m", "n", "o", "p", "q", "r", "s", "t",
                                          "u", "v", "w", "x", "y", "z"
                                          };
```

2)the input data is stored in a string “name”.
3)counter=name.length() * 1337
4)another string variable is initialized to null to calculate the password or key.
5)for(i=0;i<counter.length();i++)

```
 {
    key= key + keybuilding function(counter[i]);
 }
```

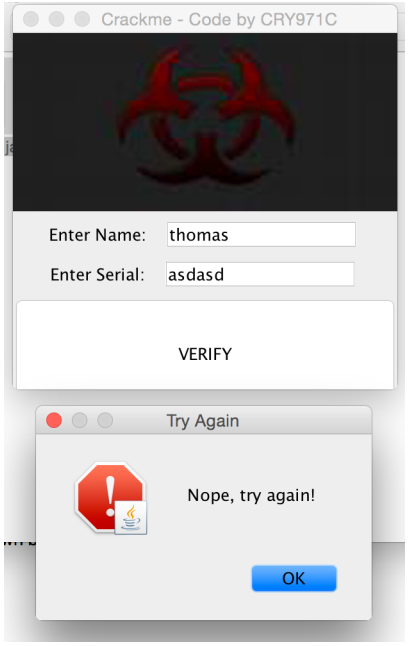
SOLUTION:

A Key generator was made using the above cracked logic.

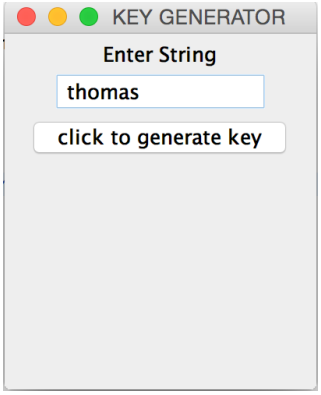
The screenshots of the keygen are placed below.

The whole process is being shown as screenshots down below

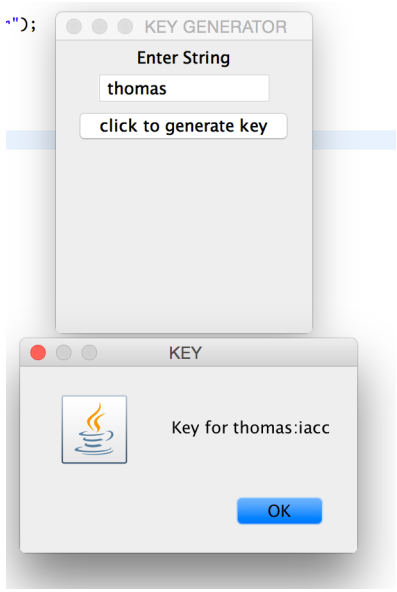
a)the crackme.jar when opened and a random key for the string is given,



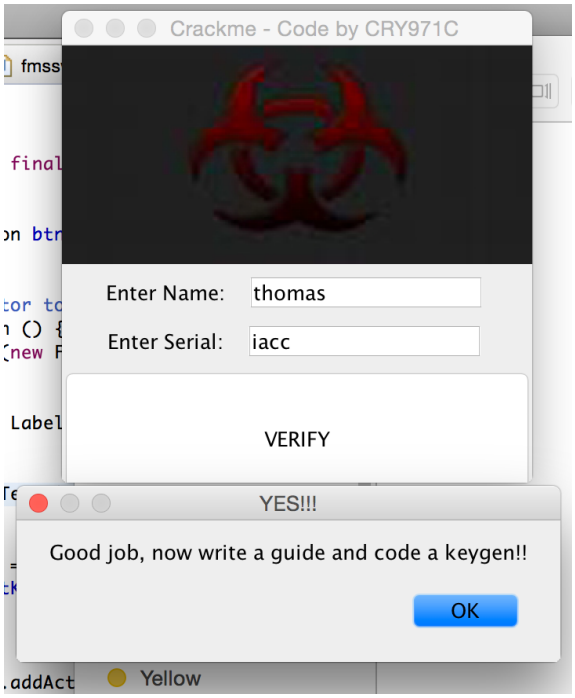
b)the keygen swing applet being run and input is given as the same string.



c)the key being created upon pressing the button



d)using the generated key, in the crackMe



DESCRIPTION OF SOLUTION:

The solution has to work, and it does work since the algorithm to create a key has been duplicated just like that and a key was generated.The crackle doesn't know that is a password generated by using another application.It just checks if the password matches with what it has calculated and created by itself.