

2017年陕西省网络空间安全技术大赛——人民的名义-抓捕赵德汉2——Writeup

转载

baikeng3674 于 2017-04-23 21:48:00 发布 214 收藏

文章标签: [java](#) [开发工具](#)

原文链接: <http://www.cnblogs.com/WangAoBo/p/6754221.html>

版权

- 下载下来的文件是一个jar包, 用die和binwalk检查, 确实是一个纯正的jar包

java -jar FileName 运行jar包, 观察文件的外部特征, 发现也是判断password的题目

用查看jar包的工具jd-gui查看反编译的代码

大致浏览打码, 发现UnitTests中的main函数很可疑, 该段代码如下:

```
public static void main(String[] args)
{
    JFrame frame = new JFrame("Key check");
    JButton button = new JButton("Click to activate");

    button.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent ae)
        {
            String str = JOptionPane.showInputDialog(null, "Enter the product key: ",
                "xxxx-xxxx-xxxx-xxxx", 1);
            if (XXXXXXXXXXXXXXXX.M(str)) {
                JOptionPane.showMessageDialog(null, "Well done that was the correct key",
                    "Key check", 1);
            } else {
                JOptionPane.showMessageDialog(null, "                Sorry that was the incorrect key \nRemember i
                    "Key check", 1);
            }
        }
    });
}
```

虽然我不懂java, 但也大致能看出这是突破点, str为输入的字符串, 且应为xxxx-xxxx-xxxx-xxxx形式, 只需要让XXXXXXXXXXXXXXXX.M(str)的返回值为1即可

- 跟进XXXXXXXXXXXXXXXX.M(str)函数

```

public static boolean M(String 和味)
{
    if ((和味 != null) && (和味.length() == 19))
    {
        a1_ = System.arraycopy(r^a, 0, a1_, 5, 5);

        boolean keyGuessWrong = true;
        int z = 0;
        for (int z = 0; z < 4; z++)
        {
            for (int z = 0; z < 4; z++) {
                if (和味.charAt(z + z) != a1_.charAt(Start.aaaaaaaaaaaaaaaa(z + z, a1_))) {
                    keyGuessWrong = false;
                }
            }
            z += 5;
        }
        return keyGuessWrong;
    }
    return false;
}

```

百度了charAt等函数的作用后，可以得到这段代码的逻辑

□

- 跟进**Start.aaaaaaaaaaaaaaaa(z + z, a1_)**，相关代码如下：

```

public static int aaaaaaaaaaaaaaaaaa(int \, String G)
{
    return \, □□(\) % G.length();
}

private static int \, □□(int \)
{
    if (\ > 2) {
        return 2 - \)□□\, + (1 - \)□□\,;
    }
    return 1;
}

```

可以看出这个函数的逻辑：

- \, □□返回num[0] = num[1] = num[2] = 1的斐波那契数列
- aaaaaaaaaaaaaaaaaa返回斐波那契数列模G.length()的值

- 于是再分析字符串**G**(即为传递的参数**a1_**)，发现**a1_**是由**a1_ = System.arraycopy(r^a, 0, a1, 5, 5)**产生的；

java中有名为System.arraycopy的函数，但跟进去System.arraycopy函数可以发现这里的System.arraycopy函数是出题者自己定义的，这是本题最大的坑点

- 跟进System.arraycopy函数

```
public static String arraycopy(Object src, int srcPos, Object dest, int destPos, int length)
{
    return Start.main(null);
}

-----分割线-----

public static String main(String... args)
{
    String x = "";
    for (int $ : "vÈ¼qÊÊ~ÆÆÊv¼Ê²Ê²Âî¼", "-".toCharArray()) {
        x = x + (char)(($ >> 1) + 15);
    }
    return x;
}
```

可以看出arraycopy函数是伪装成库函数的自定义函数，并且返回值与传递的参数无关，返回的x字符串是固定的

根据百度到的java语法规则分析上段代码逻辑：

x是由一段乱码vÈ¼qÊÊ~ÆÆÊv¼Ê²Ê²Âî¼, - 中的每两位经过(char) ((ch >> 1) + 15)操作得来的，这段乱码转化成unicode格式

为v\u00C8\u00BE\u00A4\u00CA\u00CA\u00AC\u00C6\u00C6\u00CAv\u00CC\u00A4\u00CA\u00B2\u00

□

Help -> preference 中转化为unicode

着重解释为什么是每次去了两位：

Java中的编码规则是utf-8,每个字符占两个字节，int占四个字节，因此每次循环中，取了这段字符串中的4/2=2位，然后按照小端存储的规则，将取出的两位代入运算

大小端存储参考资料

<http://www.cnblogs.com/WangAoBo/p/6369979.html>

如果直接分析的话，在字节转化这里会遇到问题，当然这个问题可以用一种很直接的方法来解决，请拉倒文末。

即可解题，由上述分析得到脚本：


```
public class test {
    //static String arr1 = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    //static String arr2 = "ZYXWVUTSRQPONMLKJIHGFEDCBA";
    public static void main(String args[]){
        String arr1 = "JsnatterrtJuaththovacke";
        for(int i=0;i<19;i++){
            if(i==4||i==9||i==14||i==19){
                System.out.print('-');
            }else{
                System.out.print(arr1.charAt(check(i,arr1)));
            }
        }
    }
    public static int check(int i,String arg){
        return te(i)%arg.length();
    }
    public static int te(int i){
        if(i>2){
            return te(i-1)+te(i-2);
        }
        return 1;
    }
}
```

最后得到flag为**flag{sssn-trtk-tcea-akJr}**

□

转载于:<https://www.cnblogs.com/WangAoBo/p/6754221.html>