ctf 抓捕赵德汉_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...

原创

weixin_39678531 于 2020-12-20 11:05:35 发布
文章标签： ctf抓捕赵德汉

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 (₪₪₪₪₪₪₪₪₪₪₪₪.Ϻ(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 it is a crime to use software without paying for it",
"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๒๒๒๒๒๒๒๒๒๒๒.ᴍ(str)的返回值为1即可

跟进๒๒๒๒๒๒๒๒๒๒๒.ᴍ(str)函数

```
public static boolean  ᴍ(String 和咻)
{
if ((和咻 != null) && (和咻.length() == 19))
{
ªı_ = System.arraycopy(_ıª, 0, ªı_, 5, 5);
boolean keyGuessWrong = true;
int ƶ = 0;
for (int ƶ = 0; ƶ < 4; ƶ++)
{
for (int ẓ = 0; ẓ < 4; ẓ++) {
if (和咻.charAt(ƶ + ẓ) != ªı_.charAt(Start.๒๒๒๒๒๒๒๒๒๒๒(ƶ + ẓ, ªı_))) {
keyGuessWrong = false;
}
}
ƶ += 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.๒๒๒๒๒๒๒๒๒๒๒(ƶ + ẓ, ªı_) ，相关代码如下：

```
public static int ๒๒๒๒๒๒๒๒๒๒๒(int ＼, String Ꞓ)
{
return لريال□□(＼) % Ꞓ.length();
```

}

private static int ﷺ□□(int ヽ)

{

if (ヽ > 2) {

return 2 - ヽ)□□ﷺ + (1 - ヽ)□□ﷺ);

}

return 1;

}

可以看出这个函数的逻辑：

ﷺ□□返回num[0] = num[1] = num[2] = 1的斐波那契数列

₪₪₪₪₪₪₪₪₪₪返回斐波那契数列模₲.length()的值

于是再分析字符串₲(即为传递的参数ª_) , 发现ª_ 是由ª_ = System.arraycopy(ıª, 0, ª, 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 $: "vÈ¾¤ÊÊ¬ÆÆÊvÌ¤Ê²Ê²ÀÎ¤¨¸¬".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vÈ¾¤ÊÊ¬ÆÆÊvÌ¤Ê²Ê²ÀÎ¤¨¸¬ 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\u00CA\u00B

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

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：

```
import sys

key = 'JsnatterrtJuaththovacke'#unicode码经过处理后的字符串

num = [1, 1, 1]

for i in range(3,26):

num.append( num[i - 1] + num[i - 2] )

num[i] %= 23

#print len(key)

sys.stdout.write('flag{') #"flag{",

Z = 0

for a in range(4):

for b in range(4):

sys.stdout.write(key[ num[Z + b] ]) #key[ num[Z + b] ],

Z += 5

if Z != 20:

sys.stdout.write('-')# '-',

sys.stdout.write('}') #'}'
```

这个题更直接的做法是像官方的Writeup一样直接利用逆出的java代码写脚本，这样就不用考虑字节之间、编码之间的转换问题了。

同时可用JD—GUI的src导出功能，用eclipse导入sec文件方便分析

附官方writeup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}