




java反逆向_Java逆向解密 WriteUp

原创

阿尔法查理  于 2021-02-25 10:47:31 发布  201  收藏

文章标签: [java反逆向](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/weixin_35589827/article/details/114622971

版权

题目地址

题解

工具

<http://java-decompiler.github.io/#jd-gui-download>

我下载的是zip版。解压完成, 右击需要反编译的Reverse.class文件, "打开方式"->"JD-GUI"打开直接就反编译完成了

```
import java.util.ArrayList;

import java.util.Scanner;

public class Reverse {

    public static void main(String[] args) {

        Scanner s = new Scanner(System.in);

        System.out.println("Please input the flag );

        String str = s.next();

        System.out.println("Your input is );

        System.out.println(str);

        char[] stringArr = str.toCharArray();

        Encrypt(stringArr);

    }

    public static void Encrypt(char[] arr) {

        ArrayList Resultlist = new ArrayList<>();

        for (int i = 0; i < arr.length; i++) {

            int result = arr[i] + 64 ^ 0x20;

            Resultlist.add(Integer.valueOf(result));

        }

        int[] KEY = {

            180, 136, 137, 147, 191, 137, 147, 191, 148, 136,
```

```
133, 191, 134, 140, 129, 135, 191, 65 };
```

```
ArrayList KEYList = new ArrayList<>();
```

```
for (int j = 0; j < KEY.length; j++)
```

```
KEYList.add(Integer.valueOf(KEY[j]));
```

```
System.out.println("Result:");
```

```
if (Resultlist.equals(KEYList)) {
```

```
System.out.println("Congratulations);
```

```
} else {
```

```
System.err.println("Error);
```

```
}
```

```
}
```

```
}
```

^的优先级比+低

读代码，感觉是很简单的加密，先计算 $64 \wedge 0x20$ 得到96，于是将KEY数组中的每一个整数减去96然后转成字符，发现报错

然后注意到KEY里的最后一个整数65小于96，怀疑我的解密方式有误，于是上网查询运算符优先级，发现^的优先级低于+，那么写出正确的脚本。

```
KEY = [180, 136, 137, 147, 191, 137, 147, 191, 148, 136, 133, 191, 134, 140, 129, 135, 191, 65]
```

```
for i in KEY:
```

```
print(chr((i^0x20)-64),end=")
```

参考