# 网鼎杯2020 玄武组部分writeup

原创

yusakul 于 2020-05-22 09:28:23 发布 3577 收藏 2

分类专栏： ctf

本文链接：https://blog.csdn.net/yusakul/article/details/106273003

版权

ctf 专栏收录该内容

6 篇文章 0 订阅

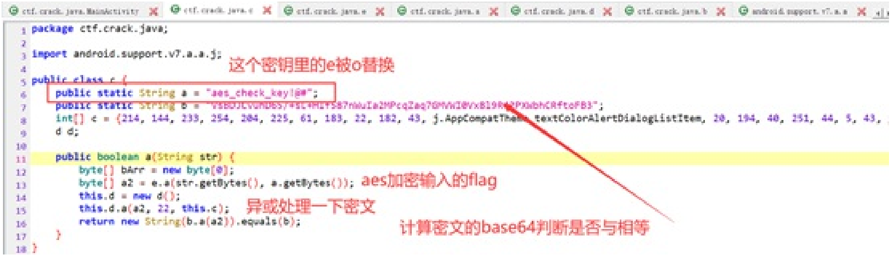
订阅专栏

> 太难了 太难了 先放出来俩re题花了大量时间没搞出来 如下是俩道简单的安卓题

## Reverse-java

解压题目得到一个java.apk，要求输入正确的flag。

主要函数：



```java
package ctf.crack.java;

import android.support.v7.a.a.j;

public class c {
    public static String a = "aes_check_key!@#";
    public static String b = "VsBDJCvuhD65/+sL+hlY5s7nWuIa2MPcqZaq7GMVWI0Vx8l9R[illegible]PXWbhCRftoFB3";
    int[] c = {214, 144, 233, 254, 204, 225, 61, 183, 22, 182, 43, j.AppCompatTheme_textColorAlertDialogListItem, 20, 194, 40, 251, 44, 5, 43,
    d d;

    public boolean a(String str) {
        byte[] bArr = new byte[0];
        byte[] a2 = e.a(str.getBytes(), a.getBytes());
        this.d = new d();
        this.d.a(a2, 22, this.c);
        return new String(b.a(a2)).equals(b);
    }
}
```

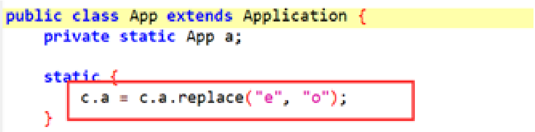
这个密钥里的e被o替换

aes加密输入的flag

异或处理一下密文

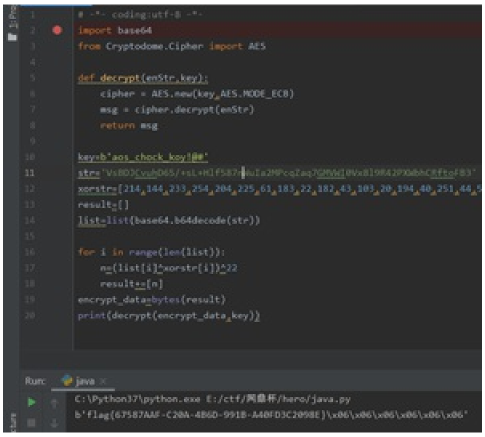
计算密文的base64判断是否与相等

坑：替换了密钥内容



```java
public class App extends Application {
    private static App a;

    static {
        c.a = c.a.replace("e", "o");
    }
```

解密代码：

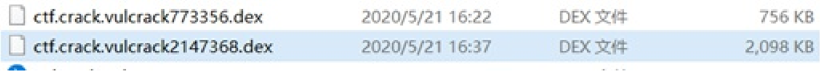


```
# -*- coding:utf-8 -*-
import base64
from Cryptodome.Cipher import AES

def decrypt(enStr,key):
    cipher = AES.new(key,AES.MODE_ECB)
    msg = cipher.decrypt(enStr)
    return msg

key=b'aos_chock_key!@#'
str='VsBD3CvuhD65/+sL+H1f587rWuIa2MPcqZaq7GMVWI0Vx819R42PXWbhC8ftoFB3'
xorstr=[214,144,233,254,204,225,61,183,22,182,43,103,20,194,40,251,44,5
result=[]
list=list(base64.b64decode(str))

for i in range(len(list)):
    n=(list[i]^xorstr[i])^22
    result+=[n]
encrypt_data=bytes(result)
print(decrypt(encrypt_data,key))
```

```
C:\Python37\python.exe E:/ctf/网鼎杯/hero/java.py
b'flag{67587AAF-C20A-4B6D-991B-A40FD3C2098E}\x06\x06\x06\x06\x06\x06'
```

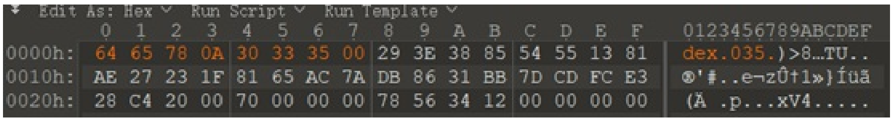
## misc –vulcrack

360加固，先脱壳，用fdex2即可：



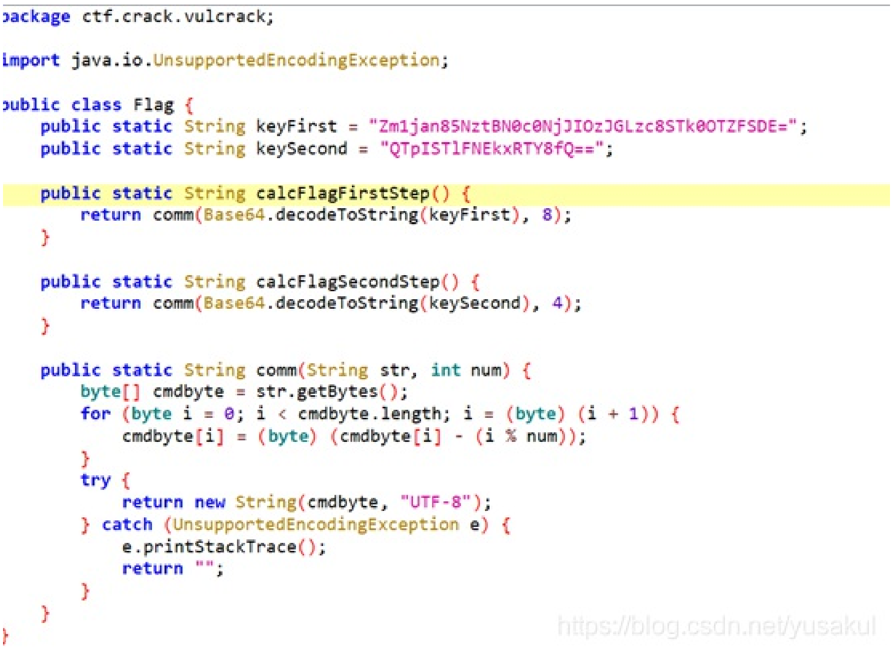
```
ctf.crack.vulcrack773356.dex     2020/5/21 16:22    DEX 文件      756 KB
ctf.crack.vulcrack2147368.dex    2020/5/21 16:37    DEX 文件    2,098 KB
```

拿到dex后修补一下文件头：



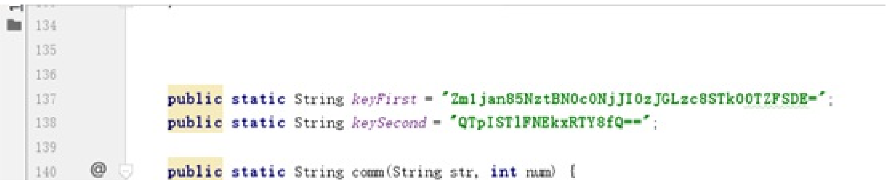
```
        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
0000h: 64 65 78 0A 30 33 35 00 29 3E 38 85 54 55 13 81   dex.035.)>8…TU..
0010h: AE 27 23 1F 81 65 AC 7A DB 86 31 BB 7D CD FC E3   ®'#..e¬zÛ†1»}Íüã
0020h: 28 C4 20 00 70 00 00 00 78 56 34 12 00 00 00 00   (Ä .p...xV4.....
```

反编译看源码：

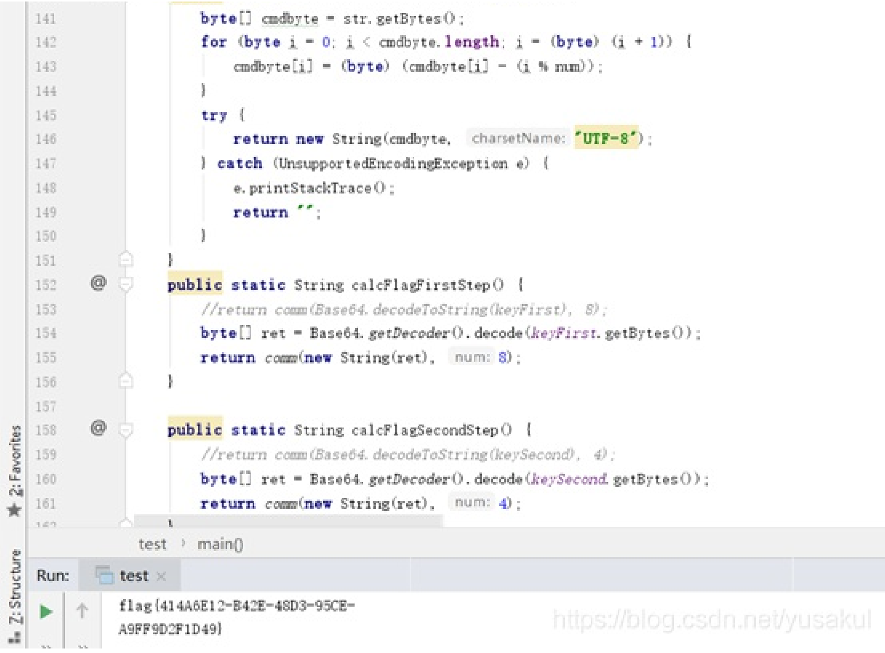


```
package ctf.crack.vulcrack;

import java.io.UnsupportedEncodingException;

public class Flag {
    public static String keyFirst = "Zm1jan85NztBN0c0NjJIOzJGLzc8STk0OTZFSDE=";
    public static String keySecond = "QTpISTlFNEkxRTY8fQ==";

    public static String calcFlagFirstStep() {
        return comm(Base64.decodeToString(keyFirst), 8);
    }

    public static String calcFlagSecondStep() {
        return comm(Base64.decodeToString(keySecond), 4);
    }

    public static String comm(String str, int num) {
        byte[] cmdbyte = str.getBytes();
        for (byte i = 0; i < cmdbyte.length; i = (byte) (i + 1)) {
            cmdbyte[i] = (byte) (cmdbyte[i] - (i % num));
        }
        try {
            return new String(cmdbyte, "UTF-8");
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
            return "";
        }
    }
}
```

抠出来用就可以了：



```
    public static String keyFirst = "Zm1jan85NztBN0c0NjJIOzJGLzc8STk0OTZFSDE=";
    public static String keySecond = "QTpISTlFNEkxRTY8fQ==";

    public static String comm(String str, int num) {
```



```
        byte[] cmdbyte = str.getBytes();
        for (byte i = 0; i < cmdbyte.length; i = (byte) (i + 1)) {
            cmdbyte[i] = (byte) (cmdbyte[i] - (i % num));
        }
        try {
            return new String(cmdbyte,  charsetName: "UTF-8");
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
            return "";
        }
    }
    public static String calcFlagFirstStep() {
        //return comm(Base64.decodeToString(keyFirst), 8);
        byte[] ret = Base64.getDecoder().decode(keyFirst.getBytes());
        return comm(new String(ret),  num: 8);
    }

    public static String calcFlagSecondStep() {
        //return comm(Base64.decodeToString(keySecond), 4);
        byte[] ret = Base64.getDecoder().decode(keySecond.getBytes());
        return comm(new String(ret),  num: 4);
    }
```

```
flag{414A6E12-B42E-48D3-95CE-
A9FF9D2F1D49}
```