Stangeapk

发表于 2021-03-05 分类于 Challenge， 2020， SWPU CTF 2020， Reverse
Challenge | 2020 | SWPU CTF 2020 | Reverse | Stangeapk

点击此处获得更好的阅读体验

WriteUp来源

官方WP

题目考点

- *APK逆向*

解题思路

使用JEB 3.0打开apk发现字符串被加密了

查看加密函数发现为异或+base64加密，编写解密函数

```
import java.util.Base64;
public class StringFog {
  private static byte[] xor(byte[] data, String key) {     //异或算法
      int len = data.length;
      int lenKey = key.length();
      int i = 0;
      int j = 0;
      while (i < len) {
          if (j >= lenKey) {
              j = 0;
          }
          data[i] = (byte) (data[i] ^ key.charAt(j));
          i++;
          j++;
      }
      return data;
  }
  public static String encode(String data, String key) {
      return new String(Base64.getEncoder().encode(xor(data.getBytes(), key)));
      //调用base64加密包
  }
  public static String decode(String data, String key) {
      return new String(xor(Base64.getDecoder().decode(data), key));
      //调用base64解密包
  }
```

得到关键字符串

```
if(!name.equals("WLLM")) {
                Toast.makeText(this, "please input your name", 0).show();
                return;
            }
if(!password.equals("welcome_to_SWPUCtf")) {
                Toast.makeText(this, "please input your password", 0).show();
                return;
            }
HttpURLConnection connection = (HttpURLConnection)new URL("http://121.196.219.16:8080/ForAndroid/mustLogin?logname=1&password=1").openConnection();
```

可以发现这是一个http请求，输入账号密码或者直接访问url均可达到目的 得到服务器返回的字符串：

根据url下载附件realapp

ps：JEB 3.9及以后会自动解密字符串

反编译apk，该apk验证流程为:

对输入内容进行凯撒加密，其中凯撒key被hook了，返回值为18，传入到native层，并对字符串进行异或操作，然后和值进行比较

hook代码：

其中几个重要参数：

```
比较字符串：{122,125,119,121,71,104,84,85,79,99,13,79,99,125,99,107,78,12,110,91,99,122,13,12,91,65}
异或的值：getInt()函数返回值，一个xtea算法的解密，其中v为{(unsigned int)-355481616,(unsigned int)1654711569}，key为{1,2,3,4}，返回值为v[0]
```

通过动态调试或者frida hook,或者观察log日志(出题人忘删除log了,难过.png) 可以得到 `v[0]=40`

脚本如下：

```
#include <iostream>
using namespace std;
#include <string>

char small_letter[26] = { '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' };
char big_letter[26] = { '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' };
char result[1000];
int p;

void decrypt(unsigned int* v, unsigned int* key) {
  unsigned int l = v[0], r = v[1], sum = 0, delta = 0x9e3779b9;
  sum = delta * 32;
  for (size_t i = 0; i < 32; i++) {
    r -= (((l << 4) ^ (l >> 5)) + l) ^ (sum + key[(sum >> 11) & 3]);
    sum -= delta;
    l -= (((r << 4) ^ (r >> 5)) + r) ^ (sum + key[sum & 3]);
  }
  v[0] = l;
  v[1] = r;
}

char* carse(char* estr,int move) {
  for (int i = 0; i < strlen(estr); i++)
  {
    if(estr[i]>='A'&&estr[i]<='Z')
        {
            p=((estr[i]-'A')- move);
            while(p<0)p+=26;
            result[i]=big_letter[p];
        }
        else if (estr[i]>='a'&&estr[i]<='z')
        {
            p=((estr[i]-'a')- move);
            while(p<0)p+=26;
            result[i]=small_letter[p];
        }
        else result[i]= estr[i];
  }
  return result;
}

int main(int argc, char const *argv[])
{
  //test
  unsigned int v[2] = { (unsigned int)-355481616,(unsigned int)1654711569 }, key[4] = { 1,2,3,4 };
  char *FK = (char*)malloc(50);
  char *b = (char*)malloc(50);
  int c[] = { 122,125,119,121,71,104,84,85,79,99,13,79,99,125,99,107,78,12,110,91,99,122,13,12,91,65 };
  decrypt(v, key);
  for (int i = 0; i < 26; i++) {
    FK[i] = c[i] ^ v[0];
  }
  b = carse(FK, 18);
  string eflag=b;
  cout << "flag is:" << eflag.substr(0,26);
  return 0;
}
```

Flag

```
flag{ZC_Yw@|}oS%oSCSKn$NaSZ%$aq}
```

- ***本文作者：*** *CTFHub*
- ***本文链接：*** *https://writeup.ctfhub.com/Challenge/2020/SWPU-CTF-2020/Reverse/34FHDKZ8z5H5cUb6yAdHBz.html*
- ***版权声明：*** *本博客所有文章除特别声明外，均采用 BY-NC-SA 许可协议。转载请注明出处！*

 # Challenge # 2020 # Reverse # SWPU CTF 2020
RealEzRE
where_is_temp