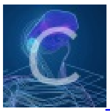
CUIT CTF WriteUp-最简单的题目

原创

RickGray 于 2014-05-22 16:20:57 发布 1667 收藏
分类专栏： CTF纪实

本文链接：https://blog.csdn.net/u013565525/article/details/26599949
版权

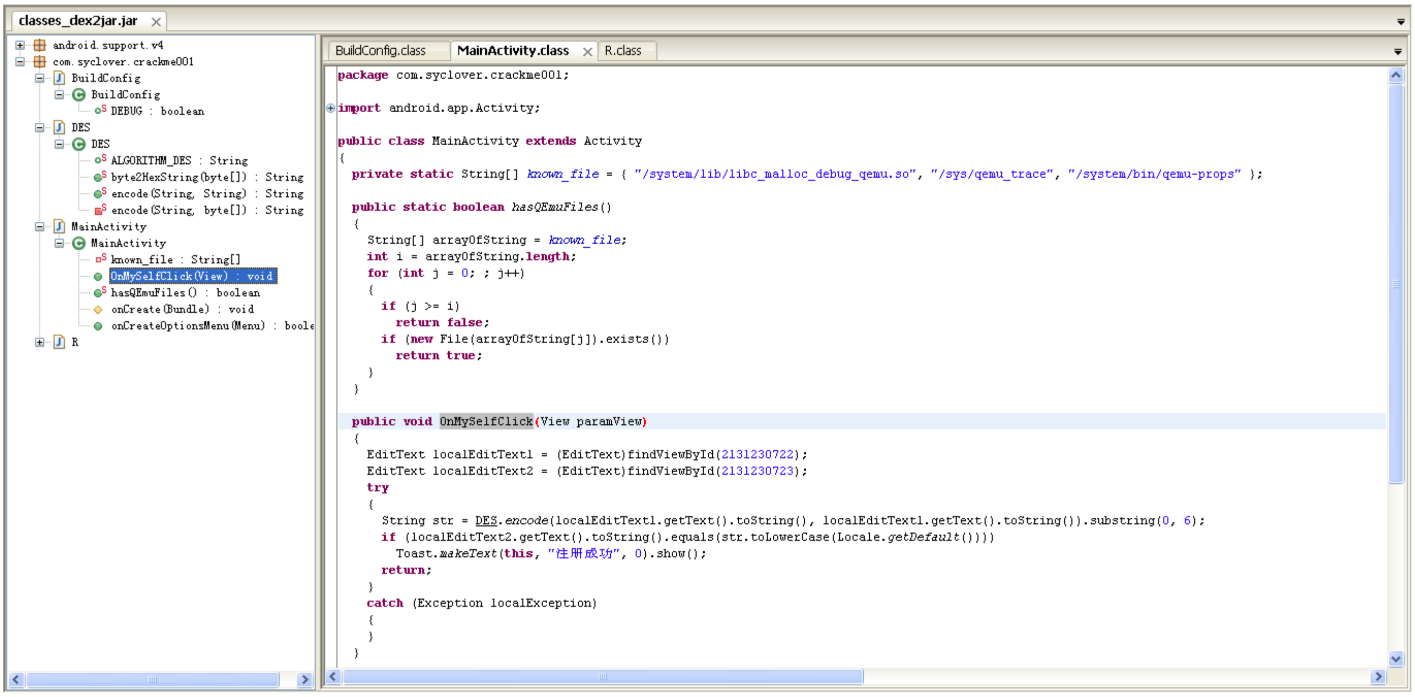


CTF纪实 专栏收录该内容
10 篇文章 0 订阅
订阅专栏

首先这是一道Android逆向，Down下apk文件后，果断进行反编译（dex2jar）

反编译处jar后，使用jd-gui.exe打开可以清晰的看到整个程序的运行流程

在MainActivity类中，有OnMySelfClick()方法



```java
package com.syclover.crackme001;

import android.app.Activity;

public class MainActivity extends Activity
{
  private static String[] known_file = { "/system/lib/libc_malloc_debug_qemu.so", "/sys/qemu_trace", "/system/bin/qemu-props" };

  public static boolean hasQEmuFiles()
  {
    String[] arrayOfString = known_file;
    int i = arrayOfString.length;
    for (int j = 0; ; j++)
    {
      if (j >= i)
        return false;
      if (new File(arrayOfString[j]).exists())
        return true;
    }
  }

  public void OnMySelfClick(View paramView)
  {
    EditText localEditText1 = (EditText)findViewById(2131230722);
    EditText localEditText2 = (EditText)findViewById(2131230723);
    try
    {
      String str = DES.encode(localEditText1.getText().toString(), localEditText1.getText().toString()).substring(0, 6);
      if (localEditText2.getText().toString().equals(str.toLowerCase(Locale.getDefault())))
        Toast.makeText(this, "注册成功", 0).show();
      return;
    }
    catch (Exception localException)
    {
    }
  }
```

简单地说，就是将用户名当作DES.encode()的参数算出密文，取密文的前6位作为注册码

ok，直接复制源码拷贝进Ecplise，处理一番后，直接运行，得到flag：uynvo4

```java
package com.syclover.crackme001;

import javax.crypto.Cipher;
import javax.crypto.SecretKey;
import javax.crypto.SecretKeyFactory;
import javax.crypto.spec.DESKeySpec;
import javax.crypto.spec.IvParameterSpec;

import com.sun.org.apache.xml.internal.security.utils.Base64;


public class DES {
    public static final String ALGORITHM_DES = "DES/CBC/PKCS5Padding";

    public static void main(String args[]) {
        String user = "syclover";
        try {
            String key = DES.encode(user, user).substring(0, 6).toLowerCase();
            System.out.println(key);
        }
        catch(Exception localException) {
        }
    }

    public static String encode(String paramString1, String paramString2)
            throws Exception {
        return encode(paramString1, paramString2.getBytes());
    }

    private static String encode(String paramString, byte[] paramArrayOfByte)
            throws Exception {
        try {
            DESKeySpec localDESKeySpec = new DESKeySpec(paramString.getBytes());
            SecretKey localSecretKey = SecretKeyFactory.getInstance("DES")
                    .generateSecret(localDESKeySpec);
            Cipher localCipher = Cipher.getInstance("DES/CBC/PKCS5Padding");
            localCipher.init(1, localSecretKey,
                    new IvParameterSpec("JoyChou ".getBytes()));
            String str = Base64.encode(
                    localCipher.doFinal(paramArrayOfByte), 0);
            return str;
        } catch (Exception localException) {
        }
        String localException = "";
        throw new Exception(localException);
    }
}
```