WuCup WP -Xherlock

Web

> Sign

蚁剑连接即可，根目录找到flag

Misc

> Sign

十六进制转字符串即可

> 原神启动！

stegsolve打开图片可以找到第一层解压密码

解压后拿到word再解压，找到zip和图片，图片还是stegsolve找到第二层解压密码

解压还有一个加密压缩包，最后在word里删掉图片后面有最后一层的解压密码，拿到flag

Crypto

> easy

rc4解密下即可

```python
def KSA(key):
    """ Key-Scheduling Algorithm (KSA) 密钥调度算法"""
    S = list(range(256))
    j = 0
    for i in range(256):
        j = (j + S[i] + key[i % len(key)]) % 256
        S[i], S[j] = S[j], S[i]
    return S


def PRGA(S):
    """ Pseudo-Random Generation Algorithm (PRGA) 伪随机数生成算法"""
    i, j = 0, 0
    while True:
        i = (i + 1) % 256
        j = (j + S[i]) % 256
        S[i], S[j] = S[j], S[i]
```

```
        K = S[(S[i] + S[j]) % 256]
        yield K


def RC4(key, text):
    """ RC4 encryption/decryption """
    S = KSA(key)
    keystream = PRGA(S)
    res = []
    for char in text:
        res.append(char ^ next(keystream))
    return bytes(res)


if __name__ == "__main__":
    key = b"hello world"
    text = [0xd8, 0xd2, 0x96, 0x3e, 0x0d, 0x8a, 0xb8, 0x53, 0x3d, 0x2a, 0x7f, 0xe2, 0x96, 0xc5, 0x29, 0x23, 0x39, 0x24,
            0x6e, 0xba, 0x0d, 0x29, 0x2d, 0x57, 0x52, 0x57, 0x83, 0x59, 0x32, 0x2c, 0x3a, 0x77, 0x89, 0x2d, 0xfa, 0x72,
            0x61, 0xb8, 0x4f]
    print(len(text))
    print(RC4(key, text))
    print(RC4(key, text).decode())

```

Reverse

> HotDog

jeb反编译apk，找到关键代码

```
package net.wucup.hotdog;

import android.content.Context;
import dalvik.system.DexClassLoader;
import dalvik.system.PathClassLoader;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.lang.reflect.Array;
import java.lang.reflect.Field;
import java.net.HttpURLConnection;
import java.net.URL;

public class H {
    private static File DEX_FILE = null;
    private static final String DEX_SUFFIX = ".dex";
```

```
18      private static final String FIELD_DEX_ELEMENTS = "dexElements";
19      private static final String FIELD_PATH_LIST = "pathList";
20      private static final String NAME_BASE_DEX_CLASS_LOADER =
"dalvik.system.BaseDexClassLoader";
21      private static final String OPTIMIZE_DEX_DIR = "odex";
22      private static final String TAG = "HotDog";
23
24      static {
25          System.loadLibrary("hotdog");
26      }
27
28      private Object combineElementArray(Object pathElements, Object
dexElements) {
29          Class componentType =
pathElements.getClass().getComponentType();
30          int i = Array.getLength(pathElements);
31          int j = Array.getLength(dexElements);
32          Object result = Array.newInstance(componentType, i + j);
33          System.arraycopy(dexElements, 0, result, 0, j);
34          System.arraycopy(pathElements, 0, result, j, i);
35          return result;
36      }
37
38      public void doHotFix(Context context) throws IllegalAccessException,
NoSuchFieldException, ClassNotFoundException {
39          if(context == null) {
40              return;
41          }
42
43          File dexFile = context.getFilesDir();
44          if(!dexFile.exists()) {
45              dexFile.mkdir();
46          }
47
48          H.DEX_FILE = new File(dexFile.getAbsolutePath(), "hotdog.dex");
49          if(!H.DEX_FILE.exists()) {
50              this.down();
51          }
52
53          File odexFile = context.getDir("odex", 0);
54          if(!odexFile.exists()) {
55              odexFile.mkdir();
56          }
57
58          File[] listFiles = dexFile.listFiles();
59          if(listFiles != null && listFiles.length != 0) {
60              String dexPath = this.getPatchDexPath(listFiles);
61              String odexPath = odexFile.getAbsolutePath();
62              PathClassLoader pathClassLoader =
(PathClassLoader)context.getClassLoader();
```

```
            DexClassLoader dexClassLoader = new DexClassLoader(dexPath, odexPath, null, pathClassLoader);
            this.setDexElements(pathClassLoader, this.combineElementArray(this.getDexElements(pathClassLoader), this.getDexElements(dexClassLoader)));
            H.DEX_FILE.delete();
            return;
        }
    }

    private native void down() {
    }

    private void down(String downUrl) {
        try {
            HttpURLConnection connection = (HttpURLConnection)new URL(downUrl).openConnection();
            connection.setRequestMethod("GET");
            InputStream inputStream = connection.getInputStream();
            H.saveInputStreamToFile(inputStream, H.DEX_FILE);
            H.DEX_FILE.setReadOnly();
            inputStream.close();
            connection.disconnect();
        }
        catch(IOException v0) {
        }
    }

    private Object getDexElements(ClassLoader classLoader) throws ClassNotFoundException, NoSuchFieldException, IllegalAccessException {
        Field pathListField = Class.forName("dalvik.system.BaseDexClassLoader").getDeclaredField("pathList");
        pathListField.setAccessible(true);
        Object dexPathList = pathListField.get(classLoader);
        Field dexElementsField = dexPathList.getClass().getDeclaredField("dexElements");
        dexElementsField.setAccessible(true);
        return dexElementsField.get(dexPathList);
    }

    private String getPatchDexPath(File[] listFiles) {
        StringBuilder sb = new StringBuilder();
        int i;
        for(i = 0; i < listFiles.length; ++i) {
            File file = listFiles[i];
            if(file.getName().endsWith(".dex")) {
                if(i != 0 && i != listFiles.length - 1) {
                    sb.append(File.pathSeparator);
                }
```

```
105
106                 sb.append(file.getAbsolutePath());
107             }
108         }
109
110         return sb.toString();
111     }
112
113     public static void saveInputStreamToFile(InputStream inputStream,
    File outputFile) {
114         FileOutputStream fileOutputStream = null;
115         try {
116             File parentDir = outputFile.getParentFile();
117             if(parentDir != null && !parentDir.exists() &&
    !parentDir.mkdirs()) {
118                 throw new IOException("无法创建目标目录：" +
    parentDir.getAbsolutePath());
119             }
120
121             fileOutputStream = new FileOutputStream(outputFile);
122             byte[] buffer = new byte[0x2000];
123             while(true) {
124                 int v3 = inputStream.read(buffer);
125                 if(v3 == -1) {
126                     break;
127                 }
128
129                 fileOutputStream.write(buffer, 0, v3);
130             }
131         }
132         catch(IOException v1_1) {
133             if(fileOutputStream != null) {
134                 try {
135                     fileOutputStream.close();
136                 label_33:
137                     if(inputStream != null) {
138                         inputStream.close();
139                         return;
140                     }
141                 }
142                 catch(IOException v1_3) {
143                 }
144
145                 return;
146             }
147
148             goto label_33;
149         }
150         catch(Throwable v1) {
151             if(fileOutputStream == null) {
```

```
                goto label_40;
            }
            else {
                try {
                    fileOutputStream.close();
                label_40:
                    if(inputStream != null) {
                        inputStream.close();
                    }
                }
                catch(IOException v2_1) {
                }
            }

            throw v1;
        }

        try {
            fileOutputStream.close();
            if(inputStream != null) {
                inputStream.close();
                return;
            }
        }
        catch(IOException v1_3) {
            return;
        }
    }

    private void setDexElements(ClassLoader classLoader, Object value)
throws ClassNotFoundException, NoSuchFieldException,
IllegalAccessException {
        Field pathListField =
Class.forName("dalvik.system.BaseDexClassLoader").getDeclaredField("path
List");
        pathListField.setAccessible(true);
        Object dexPathList = pathListField.get(classLoader);
        Field dexElementsField =
dexPathList.getClass().getDeclaredField("dexElements");
        dexElementsField.setAccessible(true);
        dexElementsField.set(dexPathList, value);
    }
}
```

由该代码可知从so文件的down函数里获取了url下载dex，并调用里面的检查方法。因此查看so文件

```
__int64 __fastcall Java_net_wucup_hotdog_H_down(_JNIEnv *a1, __int64 a2)
{
  const char *v2; // x0
  __int64 MethodID; // x0
```

```
  __int64 Class; // [xsp+20h] [xbp-60h]
  __int64 v7; // [xsp+28h] [xbp-58h]
  char v10; // [xsp+47h] [xbp-39h] BYREF
  char v11[48]; // [xsp+48h] [xbp-38h] BYREF
  __int64 v12; // [xsp+78h] [xbp-8h]

  v12 = *(_QWORD *)(_ReadStatusReg(ARM64_SYSREG(3, 3, 13, 0, 2)) + 40);
  sub_1B08(&v10);
  v2 = (const char *)crypt::Xor_string<42u>::decrypt(v11);
  v7 = _JNIEnv::NewStringUTF(a1, v2);
  Class = _JNIEnv::FindClass(a1, "net/wucup/hotdog/H");
  MethodID = _JNIEnv::GetMethodID(a1, Class, "down", "
(Ljava/lang/String;)V");
  return _JNIEnv::CallVoidMethod(a1, a2, MethodID, v7);
}
unsigned int *__fastcall crypt::Xor_string<42u>::decrypt(unsigned int
*a1)
{
  unsigned int i; // [xsp+Ch] [xbp-14h]
  unsigned int *v3; // [xsp+10h] [xbp-10h]

  v3 = a1 + 1;
  for ( i = 0; i < *a1; ++i )
    *((_BYTE *)v3 + i) = (*((_BYTE *)a1 + i + 4) - *((_BYTE *)a1 + 47)) ^
*((_BYTE *)a1 + 46);
  *((_BYTE *)v3 + *a1) = 0;
  return a1 + 1;
}
void *__usercall sub_1B08@<X0>(void *a1@<X8>)
{
  return memcpy(a1, ")", 0x30u);    // 这里不完整，得双击进汇编查看
}
```

是一个简单的加密处理

```
c = [0x29, 0x00, 0x00, 0x00, 0xD5, 0xD1, 0xD1, 0xCD, 0xCA, 0x03, 0x16,
0x16, 0xCD, 0xDC, 0xD7, 0x17, 0xCE, 0xD0, 0xDA, 0xD0, 0xCD, 0x17, 0xDA,
0xD7, 0x16, 0xDA, 0xD5, 0xDC, 0xD9, 0xD9, 0xE0, 0xD7, 0xDE, 0xE0, 0x16,
0xD5, 0xD6, 0xD1, 0xE1, 0xD6, 0xDE, 0x17, 0xE1, 0xE0, 0xC5, 0x3D, 0x7B,
0xC2]
v = [0] * c[0]
for i in range(c[0]):
    v[i] = ((c[i+4]-c[47])^c[46]) & 0xff
print("".join(map(chr, v))) # https://pan.wucup.cn/challenge/hotdog.dex
```

下载dex反编译

```
package net.wucup.hotdog;

```

```java
import java.security.InvalidAlgorithmParameterException;
import java.security.InvalidKeyException;
import java.security.Key;
import java.security.NoSuchAlgorithmException;
import java.security.Security;
import java.security.spec.InvalidKeySpecException;
import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.SecretKeyFactory;
import javax.crypto.spec.DESedeKeySpec;
import javax.crypto.spec.IvParameterSpec;
import org.bouncycastle.jce.provider.BouncyCastleProvider;

public class T {
    public static enum Padding {
        NO_PADDING("NoPadding"),
        PKCS5_PADDING("PKCS5Padding");

        private String value;

        private Padding(String arg3) {
            this.value = arg3;
        }
    }

    private static final String ALGORITHM_3DES = "DESEDE";
    public IvParameterSpec IV_PARAMETER_SPEC;

    public byte[] encryptCbc(byte[] arg3, byte[] arg4, Padding arg5) {
        Security.addProvider(new BouncyCastleProvider());
        try {
            Key v3 = this.keyGenerator(arg3);
            Cipher v5 = Cipher.getInstance("DESEDE/CBC/" +
Padding.-$$Nest$fgetvalue(arg5));
            v5.init(1, v3, this.IV_PARAMETER_SPEC);
            return v5.doFinal(arg4);
        }
        catch(InvalidKeyException unused_ex) {
            throw new UnsupportedOperationException("Invalid Key");
        }
        catch(NoSuchAlgorithmException unused_ex) {
            throw new UnsupportedOperationException("No such algorithm");
        }
        catch(InvalidKeySpecException unused_ex) {
            throw new UnsupportedOperationException("Invalid key spec");
        }
        catch(NoSuchPaddingException unused_ex) {
            throw new UnsupportedOperationException("No such padding");
```

```
        }
        catch(BadPaddingException unused_ex) {
            throw new UnsupportedOperationException("Bad padding");
        }
        catch(IllegalBlockSizeException unused_ex) {
            throw new UnsupportedOperationException("Illegal block
size");
        }
        catch(InvalidAlgorithmParameterException unused_ex) {
            throw new UnsupportedOperationException("Illegal algorithm
parameter");
        }
    }

    private Key keyGenerator(byte[] arg1) throws InvalidKeyException,
NoSuchAlgorithmException, InvalidKeySpecException {
        DESedeKeySpec v0 = new DESedeKeySpec(arg1);
        return SecretKeyFactory.getInstance("DESEDE").generateSecret(v0);
    }
}

package net.wucup.hotdog;

import android.widget.Toast;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.nio.charset.StandardCharsets;
import java.util.Arrays;
import java.util.zip.InflaterInputStream;
import javax.crypto.spec.IvParameterSpec;

public class V {
    public static void verify(String arg4) throws IOException,
ClassNotFoundException {
        ObjectInputStream v0 = new ObjectInputStream(new
InflaterInputStream(A.getInstance().getAssets().open("data")));
        T v1 = new T();
        v1.IV_PARAMETER_SPEC = new
IvParameterSpec(((byte[])v0.readObject()));
        byte[] v2 = (byte[])v0.readObject();
        if(Arrays.equals(((byte[])v0.readObject()), v1.encryptCbc(v2,
arg4.getBytes(StandardCharsets.UTF_8), Padding.PKCS5_PADDING))) {
            Toast.makeText(A.getInstance(), "Congratulations!",
0).show();
            return;
        }

        Toast.makeText(A.getInstance(), "Wrong!", 0).show();
    }
}
```

可以看到从data里读取了3des的iv、密钥、密文，合并下java写解密

```java
import java.io.*;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.security.*;
import javax.crypto.*;
import javax.crypto.spec.*;
import java.security.spec.InvalidKeySpecException;
import java.util.Arrays;
import java.util.zip.InflaterInputStream;

public class Des3Utils {
    private static final String ALGORITHM_3DES = "DESEDE";
    public IvParameterSpec IV_PARAMETER_SPEC;

    public static void main(String[] args) throws IOException,
ClassNotFoundException {
        // 这里是验证字符串的调用示例
        String inputString = "test_string";  // 你想要验证的字符串
        verify(inputString);
    }

    public static void verify(String str) throws IOException,
ClassNotFoundException {
        // 修改为直接读取同目录下的data文件
        try (ObjectInputStream objectInputStream = new
ObjectInputStream(new
InflaterInputStream(Files.newInputStream(Paths.get("data"))))) {
            Des3Utils verifier = new Des3Utils();
            verifier.IV_PARAMETER_SPEC = new IvParameterSpec((byte[])
objectInputStream.readObject());
            byte[] decryptedData = verifier.decryptCbc(
                    (byte[]) objectInputStream.readObject(),
                    (byte[]) objectInputStream.readObject(),
                    Padding.PKCS5_PADDING);
            System.out.println("Decrypted Data (in bytes): " +
Arrays.toString(decryptedData));
            // 读取加密数据并进行验证
            if (Arrays.equals((byte[]) objectInputStream.readObject(),
verifier.encryptCbc(
                    (byte[]) objectInputStream.readObject(),
                    str.getBytes(StandardCharsets.UTF_8),
                    Padding.PKCS5_PADDING))) {
                System.out.println("Congratulations!");
            } else {
                System.out.println("Wrong!");
            }
        }
```

```
    }
    public byte[] decryptCbc(byte[] key, byte[] encryptedData, Padding padding) {
        try {
            Key keyGenerator = keyGenerator(key);
            Cipher cipher = Cipher.getInstance("DESEDE/CBC/" + padding.value);
            cipher.init(Cipher.DECRYPT_MODE, keyGenerator, this.IV_PARAMETER_SPEC);
            return cipher.doFinal(encryptedData);
        } catch (Exception e) {
            throw new UnsupportedOperationException("Decryption failed: " + e.getMessage());
        }
    }
    public byte[] encryptCbc(byte[] key, byte[] data, Padding padding) {
        try {
            Key keyGenerator = keyGenerator(key);
            Cipher cipher = Cipher.getInstance("DESEDE/CBC/" + padding.value);
            cipher.init(Cipher.ENCRYPT_MODE, keyGenerator, this.IV_PARAMETER_SPEC);
            return cipher.doFinal(data);
        } catch (Exception e) {
            throw new UnsupportedOperationException("Encryption failed: " + e.getMessage());
        }
    }

    private Key keyGenerator(byte[] key) throws InvalidKeyException, NoSuchAlgorithmException, InvalidKeySpecException, InvalidKeySpecException {
        return SecretKeyFactory.getInstance(ALGORITHM_3DES).generateSecret(new DESedeKeySpec(key));
    }

    public enum Padding {
        NO_PADDING("NoPadding"),
        PKCS5_PADDING("PKCS5Padding");

        private String value;

        Padding(String value) {
            this.value = value;
        }
    }
}
// WuCup{3DES_also_known_as_TDEA}
```

> I love CTF

smc，两处wucup段一个异或0x43、一个0x44，还原下即可

因为有个左移是负数没搞懂所以直接用c++爆破

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

int main() {
  size_t v1; // rax
  unsigned __int8 v2; // al
  __int64 v3; // rcx
  unsigned __int8 v4; // al
  unsigned __int64 v7; // [rsp+20h] [rbp-10h]
  int i; // [rsp+28h] [rbp-8h]
  char v9; // [rsp+2Eh] [rbp-2h]
  char v10; // [rsp+2Fh] [rbp-1h]
  long long l[20] = {
    0x027627626F09D86D, 0x0276276267BB1378, 0x027627626F09D85C,
0x027627626F3B1366,
    0x02762762634EC49B, 0x027627626EA7621A, 0x0276276267BB134B,
0x02762762656C4E53,
    0x027627626F6C4E48, 0x027627626662759E, 0x0276276267BB131B,
0x02762762634EC449,
    0x027627626EA761C5, 0x027627626289D7DF, 0x0276276261313A47,
0x0276276267BB12D6,
    0x027627625EE2753D, 0x02762762637FFF04, 0x027627625F44EB43,
0x027627625E4EC3CF
  };
  v1 = 20;
  v7 = v1;
  unsigned long long v6[20] = {0};
  v10 = -1;
  v9 = 103;
  int tmp_v9 = v9, tmp_v10 = v10;
  printf("WuCup{");
  for ( i = 0; i < v7; ++i )
  {
    for (int j = 32; j < 127; j++) {
        v9 = tmp_v9;
        v10 = tmp_v10;
        v6[i] = 0;
        v6[i] = v7 ^ j;
        v6[i] *= v7;
        v6[i] <<= (v10 - v9)&0xff;
        v6[i] ^= (unsigned __int8)(v10 + 1) - v7 * (unsigned __int8)(v9 +
1);
        v6[i] += ((unsigned __int8)(v10 + 2) - v7) ^ ((unsigned __int8)
(v9 + 2) + v7);
```

```
        v6[i] -= ((unsigned __int8)(v9 + 3) + v7) * ((unsigned __int8)(v10 + 3) + v7);
        v2 = v9 + 4;
        v9 += 5;
        v3 = v2;
        v4 = v10 + 4;
        v10 += 5;
        v6[i] = v6[i] / (v3 - (unsigned __int64)v4);
        if (v6[i] == l[i]) {
            printf("%c", j);
            tmp_v9 = v9;
            tmp_v10 = v10;
            break;
        }
    }
  }
  printf("}");
  return 0;
}   // WuCup{1_10v3_C7F_v3ry_much}
```

> If you know

还是爆破，但是第一位不知道为啥一直不对，好在可以猜到是i的变体1

```
cmp = [0x000000F5, 0x00000200, 0x00000208, 0x000001EF, 0x00000235, 0x00000274, 0x0000023A, 0x00000276, 0x000002B7, 0x00000306, 0x000002B2, 0x00000313, 0x000002E2, 0x0000032F, 0x00000371, 0x00000440, 0x00000338, 0x000003E9, 0x000003E2, 0x000003B6, 0x00000407, 0x0000043E, 0x000003BA, 0x000003F4, 0x00000415, 0x00000473, 0x000004DA]
# bruteforce
for k in range(27):
    for i in range(1, 128):
        data = i
        for j in range(27):
            if (j & 1) != 0:
                data = k + j + 2 + (k ^ data)
            else:
                data = k + j + 1 + (k ^ data)
        # print(data)
        if data == cmp[k]:
            print(chr(i), end="")    # _10v3_yOu_d34r_1f_yOu_knOw
```