[CTF] 2019SCTF 部分WP [复制链接]

yechen123 2019-6-24 18:38 楼主 电梯直达

本帖最后由 yechen123 于 2019-6-24 18:41 编辑

题目可在xctf社区获取

strange apk

一道安卓题，直接APKIDE打开。

通过阅读代码发现

[Asm] 纯文本查看 复制代码

```
public class c
    extends Application
{
    private String apkFileName;
    private String libPath;
    protected AssetManager mAssetManager;
    protected Resources mResources;
    protected Resources.Theme mTheme;
    private String odexPath;

    private void _(byte[] paramArrayOfByte)
        throws IOException
    {
        paramArrayOfByte = _0_(paramArrayOfByte);
        Object localObject = new File(this.apkFileName);
        try
        {
            localObject = new FileOutputStream((File)localObject);
            ((FileOutputStream)localObject).write(paramArrayOfByte);
            ((FileOutputStream)localObject).close();
            return;
        }
        catch (IOException paramArrayOfByte)
        {
            throw new RuntimeException(paramArrayOfByte);
        }
    }

    private byte[] _0_(byte[] paramArrayOfByte)
    {
        int i = 0;
        while (i < paramArrayOfByte.length)
        {
```

```
            paramArrayOfByte[i] = ((byte)("syclover".charAt(i % "syclover".length()) ^ param
            i += 1;
        }
        return paramArrayOfByte;
    }

    public byte[] __(String paramString)
        throws IOException
    {
        paramString = getResources().getAssets().open(paramString);
        byte[] arrayOfByte = new byte[paramString.available()];
        paramString.read(arrayOfByte);
        return arrayOfByte;
    }

    protected void attachBaseContext(Context paramContext)
    {
        super.attachBaseContext(paramContext);
        try
        {
            paramContext = getDir("sctf_odex", 0);
            localObject = getDir("sctf_lib", 0);
            this.odexPath = paramContext.getAbsolutePath();
            this.libPath = ((File)localObject).getAbsolutePath();
            localObject = new StringBuilder();
            ((StringBuilder)localObject).append(paramContext.getAbsolutePath());
            ((StringBuilder)localObject).append("/sctf.apk");
            this.apkFileName = ((StringBuilder)localObject).toString();
            paramContext = new File(this.apkFileName);
            localObject = new StringBuilder();
            ((StringBuilder)localObject).append("apk size:");
            ((StringBuilder)localObject).append(paramContext.length());
            Log.i("demo", ((StringBuilder)localObject).toString());
            if (!paramContext.exists())
            {
                paramContext.createNewFile();
                _(__("data"));
            }
            paramContext = s.invokeStaticMethod("android.app.ActivityThread", "currentActivi
            localObject = getPackageName();
            if (Build.VERSION.SDK_INT < 19) {
                paramContext = (WeakReference)((HashMap)s.getFieldOjbect("android.app.Activi
            } else {
                paramContext = (WeakReference)((ArrayMap)s.getFieldOjbect("android.app.Activ
            }
            localObject = new DexClassLoader(this.apkFileName, this.odexPath, this.libPath,
            s.setFieldOjbect("android.app.LoadedApk", "mClassLoader", paramContext.get(), lo
            paramContext = new StringBuilder();
            paramContext.append("classloader:");
            paramContext.append(localObject);
            Log.i("demo", paramContext.toString());
            return;
        }
        catch (Exception paramContext)
        {
            Object localObject = new StringBuilder();
            ((StringBuilder)localObject).append("error:");
            ((StringBuilder)localObject).append(Log.getStackTraceString(paramContext));
            Log.i("demo", ((StringBuilder)localObject).toString());
            paramContext.printStackTrace();
        }
    }
```

该APK中隐藏着一个data文件，直接修改后缀为zip打开，data解密之后就是一个APK。
上脚本解密

[Asm] 纯文本查看 复制代码

```
i = "syclover"

f = open("data", "rb")
q = open("datas","wb")

couts = 0
c = f.read(1)
qq = 0
while (1):
        uq = ord(c)^ord(i[qq%8])
        if (uq<=15):
                q.write(("0x0"+hex(uq)[2:])[2:])
        else:
                q.write((hex(uq))[2:])
        couts += 1
        qq += 1
        if (couts%2==0 and couts%16!=0):
                q.write(' ')

        if (couts%16==0):
                q.write('\n')
        c = f.read(1)
        if c==None:
                break

q.close()
f.close()
```

可以把datas里边的数据用winhex以二进制方式存入一个新文件，原来手动存的时候一直显示未预料到文件尾，还以为要手动修复zip，原来是自己的问题，可能是文件尾部多了一些数据，导致CRC校验出错。

最终得到一个新APK。
阅读代码。

[Asm] 纯文本查看 复制代码

```
protected void onCreate(Bundle paramBundle)
    {
        super.onCreate(paramBundle);
        setContentView(2131296285);
        paramBundle = (Button)findViewById(2131165218);
        findViewById(2131165322);
        paramBundle.setOnClickListener(new View.OnClickListener()
        {
            public void onClick(View paramAnonymousView)
            {
                paramAnonymousView = "";
                Object localObject1 = "";
                int i = 0;
                String str = this.val$ed.getText().toString();
                if (str.length() == 30)
                {
                    while (i < 12)
                    {
                        localObject2 = new StringBuilder();
                        ((StringBuilder)localObject2).append(paramAnonymousView);
                        ((StringBuilder)localObject2).append(str.charAt(i));
                        paramAnonymousView = ((StringBuilder)localObject2).toString();
```

```
                    i += 1;
                }
                Object localObject2 = f.sctf(paramAnonymousView);
                paramAnonymousView = (View)localObject1;
                while (i < 30)
                {
                    localObject1 = new StringBuilder();
                    ((StringBuilder)localObject1).append(paramAnonymousView);
                    ((StringBuilder)localObject1).append(str.charAt(i));
                    paramAnonymousView = ((StringBuilder)localObject1).toString();
                    i += 1;
                }
                if (((String)localObject2).equals("c2N0ZntXM2xjMG11"))
                {
                    localObject1 = new Intent();
                    ((Intent)localObject1).putExtra("data_return", paramAnonymousView);
                    s.this.setResult(-1, (Intent)localObject1);
                    s.this.finish();
                    return;
                }
                Toast.makeText(s.this.getApplicationContext(), "something wrong", 1).sho
                return;
            }
            Toast.makeText(s.this.getApplicationContext(), "something wrong", 1).show();
        }
    });
}
```

c2N0ZntXM2xjMG1l base64解密得到前半flag sctf{W3lc0me

[Asm] 纯文本查看 复制代码

```
protected void onActivityResult(int paramInt1, int paramInt2, Intent paramIntent)
  {
      TextView localTextView = (TextView)findViewById(2131165323);
      Button localButton = (Button)findViewById(2131165219);
      if (paramInt1 != 1) {
          return;
      }
      if (paramInt2 == -1)
      {
          Object localObject1 = "";
          try
          {
              Object localObject2 = MessageDigest.getInstance("MD5");
              ((MessageDigest)localObject2).update("syclover".getBytes());
              localObject2 = new BigInteger(1, ((MessageDigest)localObject2).digest()).toStr
              localObject1 = localObject2;
          }
          catch (Exception localException)
          {
              localException.printStackTrace();
          }
          if (f.encode(paramIntent.getStringExtra("data_return"), (String)localObject1).equa
          {
              localTextView.setVisibility(0);
              localButton.setVisibility(4);
          }
          else
          {
              Toast.makeText(getApplicationContext(), "one more step", 1).show();
          }
```

```
31          }}
```

[Asm] 纯文本查看 复制代码

```
public static String encode(String paramString1, String paramString2)
{
    int j = paramString1.length();
    int k = paramString2.length();
    StringBuilder localStringBuilder = new StringBuilder();
    int i = 0;
    while (i < j)
    {
        localStringBuilder.append(paramString1.charAt(i));
        localStringBuilder.append(paramString2.charAt(i / k));
        i += 1;
    }
    return localStringBuilder.toString();
}
```

~8t808_8A8n848r808i8d8-8w808r8l8d8}8 取奇数得到后半flag。
最终 sctf{W3lc0me~t0_An4r0id-w0rld}

creakme

IDA阅读代码

[Asm] 纯文本查看 复制代码

```
int sub_402540()
{
    HMODULE v0; // eax
    int v1; // eax
    _DWORD *v2; // eax
    unsigned int ser_len; // edx
    _DWORD *string; // ecx
    unsigned int string_len; // ebx
    char *ascc; // edi
    unsigned int v7; // esi
    unsigned int v8; // esi
    bool v9; // cf
    unsigned __int8 v10; // al
    unsigned __int8 v11; // al
    unsigned __int8 v12; // al
    signed int v13; // esi
    _BYTE *v14; // ecx
    _BYTE *v15; // ecx
    const char *v16; // edx
    int v17; // eax
    void *Memory; // [esp+10h] [ebp-70h]
    int v20; // [esp+20h] [ebp-60h]
    unsigned int v21; // [esp+24h] [ebp-5Ch]
    void *Dst; // [esp+28h] [ebp-58h]
    int v23; // [esp+38h] [ebp-48h]
    unsigned int v24; // [esp+3Ch] [ebp-44h]
    char Src; // [esp+40h] [ebp-40h]
    int v26; // [esp+7Ch] [ebp-4h]
```

```
029
030     v0 = GetModuleHandleW(0);
031     sub_402320(v0);
032     sub_4024A0();
033     v1 = sub_402870(std::cout, "welcome to 2019 sctf");
034     std::basic_ostream<char,std::char_traits<char>>::operator<<(v1, sub_402AC0);
035     sub_402870(std::cout, "please input your ticket:");
036     sub_402AF0(std::cin, &Src);
037     v23 = 0;
038     v24 = 15;
039     LOBYTE(Dst) = 0;
040     sub_401D30(&Dst, &Src, strlen(&Src));
041     v26 = 0;
042     v2 = aes(&Memory, (int)&Dst);
043     ser_len = strlen(aPvfqyc4ttc2uxr);
044     string = v2;
045     if ( v2[5] >= 16u )
046         string = (_DWORD *)*v2;
047     string_len = v2[4];
048     ascc = aPvfqyc4ttc2uxr;
049     v7 = v2[4];
050     if ( ser_len < string_len )
051         v7 = ser_len;
052     v9 = v7 < 4;                                                    // len -
053     v8 = v7 - 4;
054     if ( v9 )
055     {
056 LABEL_8:
057         if ( v8 == -4 )
058             goto LABEL_17;
059     }
060     else
061     {
062         while ( *string == *(_DWORD *)ascc )
063         {
064             ++string;                                               // +=4
065             ascc += 4;
066             v9 = v8 < 4;
067             v8 -= 4;
068             if ( v9 )
069                 goto LABEL_8;
070         }
071     }
072     v9 = *(_BYTE *)string < (unsigned __int8)*ascc;
073     if ( *(_BYTE *)string != *ascc
074         || v8 != -3
075         && ((v10 = *((_BYTE *)string + 1), v9 = v10 < (unsigned __int8)ascc[1], v10 != ascc[
076           || v8 != -2
077           && ((v11 = *((_BYTE *)string + 2), v9 = v11 < (unsigned __int8)ascc[2], v11 != asc
078             || v8 != -1 && (v12 = *((_BYTE *)string + 3), v9 = v12 < (unsigned __int8)ascc[3
079     {
080         v13 = -v9 | 1;
081         goto LABEL_18;
082     }
083 LABEL_17:
084     v13 = 0;
085 LABEL_18:
086     if ( !v13 )                                                     // v13要
087     {
088         if ( ser_len <= string_len )
089             v13 = ser_len < string_len;
090         else
091             v13 = -1;
092     }
093     if ( v21 >= 0x10 )
094     {
095         v14 = Memory;
096         if ( v21 + 1 >= 0x1000 )
097         {
098             v14 = ( BYTE *)*(( DWORD *)Memory - 1);
```

```
099             if ( (unsigned int)((_BYTE *)Memory - v14 - 4) > 0x1F )
100                 invalid_parameter_noinfo_noreturn(v14, v21 + 36);
101         }
102         frees(v14);
103     }
104     v26 = -1;
105     v20 = 0;
106     v21 = 15;
107     LOBYTE(Memory) = 0;
108     if ( v24 >= 0x10 )
109     {
110         v15 = Dst;
111         if ( v24 + 1 >= 0x1000 )
112         {
113             v15 = (_BYTE *)*((_DWORD *)Dst - 1);
114             if ( (unsigned int)((_BYTE *)Dst - v15 - 4) > 0x1F )
115                 invalid_parameter_noinfo_noreturn(v15, v24 + 36);
116         }
117         frees(v15);
118     }
119     v16 = "Have fun!";
120     if ( v13 )
121         v16 = "A forged ticket!!";
122     v17 = sub_402870(std::cout, v16);
123     std::basic_ostream<char,std::char_traits<char>>::operator<<(v17, sub_402AC0);
124     system("pause");
125     return 0;
126 }
```

查看sub_402320函数的代码发现

[Asm] 纯文本查看 复制代码

```
01 void __thiscall sub_402320(_DWORD *this)
02 {
03     int v1; // eax
04     __int16 v2; // bx
05     const char *v3; // esi
06     signed int i; // edi
07     int v5; // eax
08
09     v1 = this[15];
10     v2 = *(_WORD *)((char *)this + v1 + 6);
11     v3 = (char *)this + v1 + 248;
12     for ( i = 0; i < v2; ++i )                    // 判断区段
13     {
14         v5 = strcmp(v3, ".SCTF");
15         if ( v5 )
16             v5 = -(v5 < 0) | 1;
17         if ( !v5 )
18         {
19             DebugBreak();
20             return;
21         }
22         v3 += 40;
23     }
24 }
```

读取区段，判断区段是否是.SCTF区段。
如果是，就会跳到loc_4023EF并在里边调用函数解密区段。

[Asm] 纯文本查看 复制代码

```
.text:004023EF loc_4023EF:                                             ; DATA X
.text:004023EF                 mov         esp, [ebp+ms_exc.old_esp]
.text:004023F2                 lea         eax, [ebp+pbDebuggerPresent]
.text:004023F5                 push        eax                         ; pbDeb
.text:004023F6                 call        ds:GetCurrentProcess
.text:004023FC                 push        eax                         ; hProc
.text:004023FD                 call        ds:CheckRemoteDebuggerPresent
.text:00402403                 call        ds:IsDebuggerPresent
.text:00402409                 test        eax, eax
.text:0040240B                 jnz         short loc_4023B9
.text:0040240D                 cmp         [ebp+pbDebuggerPresent], eax
.text:00402410                 jnz         short loc_4023B9
.text:00402412                 mov         eax, [ebp+var_24]
.text:00402415                 mov         edx, [eax+10h]
.text:00402418                 mov         ecx, [eax+0Ch]
.text:0040241B                 add         ecx, [ebp+var_28]
.text:0040241E                 mov         esi, [ebp+var_2C]
.text:00402421                 lea         edi, [esi+1]
.text:00402424
.text:00402424 loc_402424:                                             ; CODE XRE
.text:00402424                 mov         al, [esi]                   ; sycloversy
.text:00402426                 inc         esi
.text:00402427                 test        al, al
.text:00402429                 jnz         short loc_402424 ; sycloversyclov
.text:0040242B                 sub         esi, edi
.text:0040242D                 push        esi
.text:0040242E                 push        ecx
.text:0040242F                 call        sub_402450
.text:00402434                 add         esp, 8
.text:00402437                 jmp         short loc_4023B9
.text:00402439 ; ---------------------------------------------------------------------------
.text:00402439
.text:00402439 loc_402439:                                             ; CODE XRE
.text:00402439                 add         esi, 28h
.text:0040243C                 inc         edi
.text:0040243D                 jmp         loc_402372
.text:0040243D sub_402320      endp
```

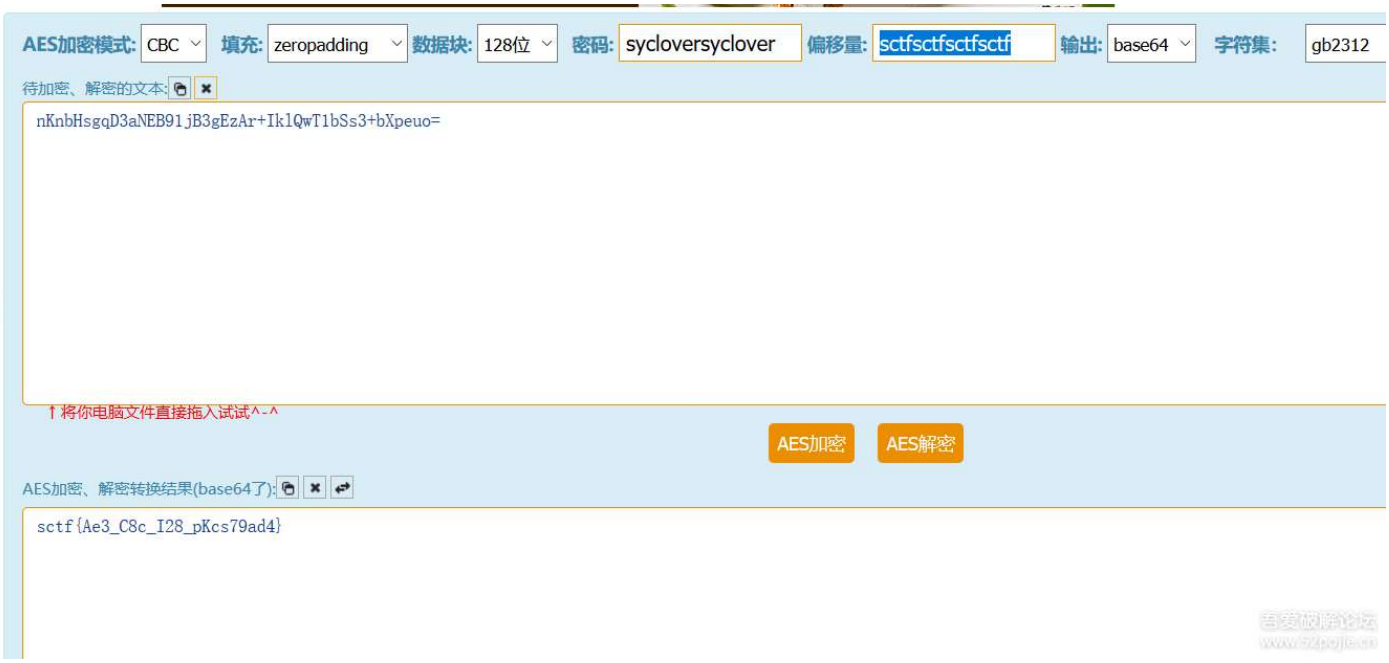
在sub_4024A0函数中，会进入.SCTF区段。
主要是解密>pvfqYc,4tTc2UxRmlJ,sB{Fh4Ck2:CFOb4ErhtIcoLo
解密成nKnbHsgqD3aNEB91jB3gEzAr+IklQwT1bSs3+bXpeuo=

sub_4020D0函数其实就是AES加密。

```
v19 = v3 + 269;
v33 = (int)(v3 + 10);
do
{
  if ( v6 > 0 )
  {
    v39 = v19;
    v20 = v8;
    v46 = (_DWORD *)v18;
    v21 = v45 - v8;
    v31 = v45 - v8;
    v32 = v41 - v8;
    v37 = v6;
    do
    {
      v22 = dword_4062E0[LOBYTE(v3[(v20 + v32) % v6 + 277])] ^ dword_4051E0[(unsigned __int8)(LOWORD(v3[(v20 + v21) % v6 + 277]) >> 8)];
      v23 = v20++;
      *v39 = *v46 ^ dword_4059E0[*((unsigned __int8 *)v39 + 35)] ^ dword_4055E0[(unsigned __int8)HIWORD(v3[(unsigned __int64)(v23 % v6) + 277])] ^ v2
      v17 = v37-- == 1;
      ++v46;
      v21 = v31;
      ++v39;
    }
    while ( !v17 );
    v19 = v3 + 269;
  }
  memcpy(v3 + 277, v19, 4 * v6);
  v8 = v43;
  v18 = v33 + 32;
  v25 = __OFSUB__(v35 + 1, v3[244]);
  v24 = v35++ + 1 - v3[244] < 0;
  v19 = v3 + 269;
  v33 += 32;
0000091A sub_4013E0:121 (40151A)
```

CBC模式，密码为sycloversyclover，偏移量为sctfsctfsctfsctf。

最终得到



Bybagame

一道游戏题，总共有三关。
第一关，

[Asm] 纯文本查看 复制代码

```
001 .text:00005593B1D94798 main:                                    ?
002 .text:00005593B1D94798 ; __unwind {
003 .text:00005593B1D94798                      push    rbp
004 .text:00005593B1D94799                      mov     rbp, rsp
005 .text:00005593B1D9479C                      sub     rsp, 160h
006 .text:00005593B1D947A3                      mov     rax, fs:28h
007 .text:00005593B1D947AC                      mov     [rbp-8], rax
```

```
.text:00005593B1D947B0                                  xor         eax, eax
.text:00005593B1D947B2                                  mov         dword ptr [rbp-150h], 0
.text:00005593B1D947BC                                  mov         qword ptr [rbp-110h], 0
.text:00005593B1D947C7                                  mov         qword ptr [rbp-108h], 0
.text:00005593B1D947D2                                  mov         qword ptr [rbp-100h], 0
.text:00005593B1D947DD                                  mov         byte ptr [rbp-0F8h], 0
.text:00005593B1D947E4                                  mov         qword ptr [rbp-0F0h], 0
.text:00005593B1D947EF                                  mov         qword ptr [rbp-0E8h], 0
.text:00005593B1D947FA                                  mov         qword ptr [rbp-0E0h], 0
.text:00005593B1D94805                                  mov         byte ptr [rbp-0D8h], 0
.text:00005593B1D9480C                                  mov         qword ptr [rbp-0D0h], 0
.text:00005593B1D94817                                  mov         qword ptr [rbp-0C8h], 0
.text:00005593B1D94822                                  mov         qword ptr [rbp-0C0h], 0
.text:00005593B1D9482D                                  mov         qword ptr [rbp-0B8h], 0
.text:00005593B1D94838                                  mov         qword ptr [rbp-0B0h], 0
.text:00005593B1D94843                                  mov         qword ptr [rbp-0A8h], 0
.text:00005593B1D9484E                                  mov         word ptr [rbp-0A0h], 0
.text:00005593B1D94857                                  mov         rax, '********'
.text:00005593B1D94861                                  mov         rdx, '*.******'
.text:00005593B1D9486B                                  mov         [rbp-90h], rax
.text:00005593B1D94872                                  mov         [rbp-88h], rdx
.text:00005593B1D94879                                  mov         rax, '.s**.***'
.text:00005593B1D94883                                  mov         rdx, '****..*.'
.text:00005593B1D9488D                                  mov         [rbp-80h], rax
.text:00005593B1D94891                                  mov         [rbp-78h], rdx
.text:00005593B1D94895                                  mov         rax, '.****.**'
.text:00005593B1D9489F                                  mov         rdx, '********'
.text:00005593B1D948A9                                  mov         [rbp-70h], rax
.text:00005593B1D948AD                                  mov         [rbp-68h], rdx
.text:00005593B1D948B1                                  mov         rax, '***..***'
.text:00005593B1D948BB                                  mov         rdx, '*#..**..'
.text:00005593B1D948C5                                  mov         [rbp-60h], rax
.text:00005593B1D948C9                                  mov         [rbp-58h], rdx
.text:00005593B1D948CD                                  mov         rax, '*..***..'
.text:00005593B1D948D7                                  mov         rdx, '*****.**'
.text:00005593B1D948E1                                  mov         [rbp-50h], rax
.text:00005593B1D948E5                                  mov         [rbp-48h], rdx
.text:00005593B1D948E9                                  mov         rax, '********'
.text:00005593B1D948F3                                  mov         rdx, '.*******'
.text:00005593B1D948FD                                  mov         [rbp-40h], rax
.text:00005593B1D94901                                  mov         [rbp-38h], rdx
.text:00005593B1D94905                                  mov         rax, '****..**'
.text:00005593B1D9490F                                  mov         rdx, '.**..***'
.text:00005593B1D94919                                  mov         [rbp-30h], rax
.text:00005593B1D9491D                                  mov         [rbp-28h], rdx
.text:00005593B1D94921                                  mov         rax, '*.*..*..'
.text:00005593B1D9492B                                  mov         [rbp-20h], rax
.text:00005593B1D9492F                                  mov         dword ptr [rbp-18h], '.**
.text:00005593B1D94936                                  mov         word ptr [rbp-14h], 2Ah ;
.text:00005593B1D9493C                                  mov         qword ptr [rbp-130h], 0
.text:00005593B1D94947                                  mov         qword ptr [rbp-128h], 0
.text:00005593B1D94952                                  mov         dword ptr [rbp-120h], 0
.text:00005593B1D9495C                                  lea         rax, [rbp-90h]
.text:00005593B1D94963                                  add         rax, 16h
.text:00005593B1D94967                                  mov         [rbp-148h], rax
.text:00005593B1D9496E                                  mov         rax, '019_ftcs' ; sctf_01
.text:00005593B1D94978                                  mov         [rbp-13Ah], rax
.text:00005593B1D9497F                                  mov         word ptr [rbp-132h], 32h
.text:00005593B1D94988                                  xor         rax, rax
.text:00005593B1D9498B                                  jb          short loc_5593B1D94990
.text:00005593B1D9498D                                  jnb         short loc_5593B1D94990
.text:00005593B1D9498D ; ---------------------------------------------------------------------------
.text:00005593B1D9498F                                  db 0E1h
.text:00005593B1D94990 ; ---------------------------------------------------------------------------
.text:00005593B1D94990
.text:00005593B1D94990 loc_5593B1D94990:                                        ; CODE X
.text:00005593B1D94990
.text:00005593B1D94990                                  lea         rdi, aPlzTellMeTheSh ; "p
.text:00005593B1D94997                                  call        _puts
.text:00005593B1D9499C                                  lea         rax, [rbp-0D0h]
```

```
078 .text:00005593B1D949A3                  mov     rsi, rax
079 .text:00005593B1D949A6                  lea     rdi, aS                 ;
080 .text:00005593B1D949AD                  mov     eax, 0
081 .text:00005593B1D949B2                  call    _scanf
082 .text:00005593B1D949B7                  mov     dword ptr [rbp-14Ch], 1
083 .text:00005593B1D949C1
084 .text:00005593B1D949C1 loc_5593B1D949C1:                                ; CODE X
085 .text:00005593B1D949C1                  cmp     dword ptr [rbp-14Ch], 0
086 .text:00005593B1D949C8                  jz      loc_5593B1D94AB1
087 .text:00005593B1D949CE                  mov     eax, [rbp-150h]
088 .text:00005593B1D949D4                  cdqe
089 .text:00005593B1D949D6
090 .text:00005593B1D949D6 loc_5593B1D949D6:
091 .text:00005593B1D949D6                  lea     rdx, [rbp-0D0h]
092 .text:00005593B1D949DD                  add     rax, rdx
093 .text:00005593B1D949E0                  movzx   eax, byte ptr [rax]
094 .text:00005593B1D949E3                  mov     [rbp-151h], al
095 .text:00005593B1D949E9                  cmp     byte ptr [rbp-151h], 77h
096 .text:00005593B1D949F0                  jnz     short loc_5593B1D949FC
097 .text:00005593B1D949F2                  sub     qword ptr [rbp-148h], 5
098 .text:00005593B1D949FA                  jmp     short loc_5593B1D94A66
099 .text:00005593B1D949FC ; ---------------------------------------------------------------------------
100 .text:00005593B1D949FC
101 .text:00005593B1D949FC loc_5593B1D949FC:                                ; CODE X
102 .text:00005593B1D949FC                  cmp     byte ptr [rbp-151h], 73h
103 .text:00005593B1D94A03                  jnz     short loc_5593B1D94A0F
104 .text:00005593B1D94A05                  add     qword ptr [rbp-148h], 5
105 .text:00005593B1D94A0D                  jmp     short loc_5593B1D94A66
106 .text:00005593B1D94A0F ; ---------------------------------------------------------------------------
107 .text:00005593B1D94A0F
108 .text:00005593B1D94A0F loc_5593B1D94A0F:                                ; CODE X
109 .text:00005593B1D94A0F                  cmp     byte ptr [rbp-151h], 64h
110 .text:00005593B1D94A16                  jnz     short loc_5593B1D94A22
111 .text:00005593B1D94A18                  add     qword ptr [rbp-148h], 1
112 .text:00005593B1D94A20                  jmp     short loc_5593B1D94A66
113 .text:00005593B1D94A22 ; ---------------------------------------------------------------------------
114 .text:00005593B1D94A22
115 .text:00005593B1D94A22 loc_5593B1D94A22:                                ; CODE X
116 .text:00005593B1D94A22                  cmp     byte ptr [rbp-151h], 61h
117 .text:00005593B1D94A29                  jnz     short loc_5593B1D94A35
118 .text:00005593B1D94A2B                  sub     qword ptr [rbp-148h], 1
119 .text:00005593B1D94A33                  jmp     short loc_5593B1D94A66
120 .text:00005593B1D94A35 ; ---------------------------------------------------------------------------
121 .text:00005593B1D94A35
122 .text:00005593B1D94A35 loc_5593B1D94A35:                                ; CODE X
123 .text:00005593B1D94A35                  cmp     byte ptr [rbp-151h], 78h
124 .text:00005593B1D94A3C                  jnz     short loc_5593B1D94A48
125 .text:00005593B1D94A3E                  add     qword ptr [rbp-148h], 19h
126 .text:00005593B1D94A46                  jmp     short loc_5593B1D94A66
127 .text:00005593B1D94A48 ; ---------------------------------------------------------------------------
128 .text:00005593B1D94A48
129 .text:00005593B1D94A48 loc_5593B1D94A48:                                ; CODE X
130 .text:00005593B1D94A48                  cmp     byte ptr [rbp-151h], 79h
131 .text:00005593B1D94A4F                  jnz     short loc_5593B1D94A5B
132 .text:00005593B1D94A51                  sub     qword ptr [rbp-148h], 19h
133 .text:00005593B1D94A59                  jmp     short loc_5593B1D94A66
134 .text:00005593B1D94A5B ; ---------------------------------------------------------------------------
135 .text:00005593B1D94A5B
136 .text:00005593B1D94A5B loc_5593B1D94A5B:                                ; CODE X
137 .text:00005593B1D94A5B                  mov     dword ptr [rbp-14Ch], 0
138 .text:00005593B1D94A65                  nop
139 .text:00005593B1D94A66
140 .text:00005593B1D94A66 loc_5593B1D94A66:                                ; CODE X
141 .text:00005593B1D94A66
142 .text:00005593B1D94A66                  add     dword ptr [rbp-150h], 1
143 .text:00005593B1D94A6D                  mov     rax, [rbp-148h]
144 .text:00005593B1D94A74                  movzx   eax, byte ptr [rax]
145 .text:00005593B1D94A77                  cmp     al, 2Eh ; '.'
146 .text:00005593B1D94A79                  jz      short loc_5593B1D94A93
```

```
.text:00005593B1D94A7B                  mov     rax, [rbp-148h]
.text:00005593B1D94A82                  movzx   eax, byte ptr [rax]
.text:00005593B1D94A85                  cmp     al, 23h ; '#'
.text:00005593B1D94A87                  jz      short loc_5593B1D94A93
.text:00005593B1D94A89                  mov     dword ptr [rbp-14Ch], 0
.text:00005593B1D94A93
.text:00005593B1D94A93 loc_5593B1D94A93:                                ; CODE X
.text:00005593B1D94A93
.text:00005593B1D94A93                  mov     rax, [rbp-148h]
.text:00005593B1D94A9A                  movzx   eax, byte ptr [rax]
.text:00005593B1D94A9D                  cmp     al, 23h ; '#'
.text:00005593B1D94A9F                  jnz     loc_5593B1D949C1
.text:00005593B1D94AA5                  lea     rdi, aGoodYouFindThe ; "g
.text:00005593B1D94AAC                  call    _puts
.text:00005593B1D94AB1
.text:00005593B1D94AB1 loc_5593B1D94AB1:                                ; CODE X
.text:00005593B1D94AB1                  cmp     dword ptr [rbp-14Ch], 0
.text:00005593B1D94AB8                  jnz     short loc_5593B1D94AD5
.text:00005593B1D94ABA                  lea     rdi, aSorryIsTNotARi ; "s
.text:00005593B1D94AC1                  mov     eax, 0
.text:00005593B1D94AC6                  call    _printf
.text:00005593B1D94ACB                  mov     eax, 0
.text:00005593B1D94AD0                  jmp     loc_5593B1D94C0C
```

先赋值迷宫，总共六个控制键，好像是三维的，但是可以化成一维。

w/s分别为后退/前进五格，a/d分别为后退/前进一格，y/x分别为后退/前进25格。

把迷宫换成一维，最终要使s和#相遇。

[Asm] 纯文本查看 复制代码

```
1 ***************.****.**s..*..******.****.************..***..**..#*..***..***.*****
```

得到sxss

第二关。
输入字符进入进入函数，最终和sctf_9102对比。
汇编看起来比较麻烦，不能f5的原因是中间多了一个0XE4字节，可以手动nop掉。

[Asm] 纯文本查看 复制代码

```
.text:0000000000000E11                  jnb     short loc_E14
.text:0000000000000E11 ; ---------------------------------------------------------------------------
.text:0000000000000E13                  db 0E4h
.text:0000000000000E14 ; ---------------------------------------------------------------------------
.text:0000000000000E14
.text:0000000000000E14 loc_E14:
.text:0000000000000E14
.text:0000000000000E14                  mov     dword ptr [rbp-244h], 0
.text:0000000000000E1E
.text:0000000000000E1E loc_E1E:
.text:0000000000000E1E                  cmp     dword ptr [rbp-238h], 0
```

nop之后

[Asm] 纯文本查看 复制代码

```
unsigned __int64 __fastcall sub_C22(const char *a1, __int64 a2)
{
    bool v2; // al
    int v3; // eax
    int v4; // eax
    int v5; // eax
    int v7; // [rsp+14h] [rbp-24Ch]
    signed int v8; // [rsp+18h] [rbp-248h]
    int v9; // [rsp+1Ch] [rbp-244h]
    int v10; // [rsp+20h] [rbp-240h]
    int v11; // [rsp+24h] [rbp-23Ch]
    int v12; // [rsp+28h] [rbp-238h]
    int v13; // [rsp+2Ch] [rbp-234h]
    char *v14; // [rsp+48h] [rbp-218h]
    int v15[130]; // [rsp+50h] [rbp-210h]
    unsigned __int64 v16; // [rsp+258h] [rbp-8h]

    v16 = __readfsqword(0x28u);
    qmemcpy(v15, &off_1740, 0x200uLL);
    v8 = 3;
    v7 = 0;
    v10 = 0;
    v11 = 0;
    v12 = strlen(a1);
    v14 = (char *)a1;
    while ( 1 )
    {
        v13 = 0;
        if ( v10 < v12 )
            break;
LABEL_13:
        if ( v10 >= v12 )
            goto LABEL_14;
    }
    do
    {
        if ( a1[v10] != 25 )
            break;
        ++v10;
        ++v13;
    }
    while ( v10 < v12 );
    if ( v10 != v12 )
    {
        if ( v12 - v10 > 1 )
        {
            v2 = v10 == 19 && a1[20] == 16;
            a1[v2];
        }
        ++v10;
        goto LABEL_13;
    }
LABEL_14:
    v9 = 0;
    while ( v12 > 0 )
    {
        v8 -= v15[*v14] == 64;
        v7 = v15[*v14] & 0x3F | (v7 << 6);
        if ( ++v9 == 4 )
        {
            v9 = 0;
            if ( v8 )
```

```
            {
                v3 = v11++;
                *(_BYTE *)(v3 + a2) = BYTE2(v7);
            }
            if ( v8 > 1 )
            {
                v4 = v11++;
                *(_BYTE *)(v4 + a2) = BYTE1(v7);
            }
            if ( v8 > 2 )
            {
                v5 = v11++;
                *(_BYTE *)(v5 + a2) = v7;
            }
        }
        ++v14;
        --v12;
    }
    return __readfsqword(0x28u) ^ v16;
}
```

其实这代码的原理就是，输入字符分成四个一组，每次取8位二进制索引字符，取字符低6位，一组可以得到4个6位二进制，最终一组得到3个字符。

可以逆推得到password2:c2N0Zl85MTAy

第三关。

[Asm] 纯文本查看 复制代码

```
signed __int64 __fastcall sub_5593B1D94FFA(char *a1)
{
    int v1; // ST24_4
    int v2; // ST28_4
    int v3; // ST2C_4
    signed int v5; // [rsp+18h] [rbp-158h]
    signed int i; // [rsp+18h] [rbp-158h]
    int cout; // [rsp+1Ch] [rbp-154h]
    int v8[16]; // [rsp+30h] [rbp-140h]
    int v9[16]; // [rsp+70h] [rbp-100h]
    int v10[29]; // [rsp+B0h] [rbp-C0h]
    unsigned int v11; // [rsp+124h] [rbp-4Ch]
    unsigned __int64 v12; // [rsp+168h] [rbp-8h]

    v12 = __readfsqword(0x28u);
    v8[0] = 0xBE;
    v8[1] = '\x04';
    v8[2] = '\x06';
    v8[3] = 0x80;
    v8[4] = 0xC5;
    v8[5] = 0xAF;
    v8[6] = 0x76;
    v8[7] = 0x47;
    v8[8] = 0x9F;
    v8[9] = 0xCC;
    v8[10] = 0x40;
    v8[11] = 0x1F;
    v8[12] = 0xD8;
    v8[13] = 0xBF;
    v8[14] = 0x92;
    v8[15] = 0xEF;
    v1 = (a1[6] << 8) | (a1[5] << 16) | (a1[4] << 24) | a1[7];
    v2 = (a1[10] << 8) | (a1[9] << 16) | (a1[8] << 24) | a1[11];
```

```
    v3 = (a1[14] << 8) | (a1[13] << 16) | (a1[12] << 24) | a1[15];// 16位
    cout = 0;
    v5 = 4;
    v10[0] = byteswap_ulong((a1[2] << 8) | (a1[1] << 16) | (*a1 << 24) | (unsigned int)a1[3]
    v10[1] = byteswap_ulong(v1);
    v10[2] = byteswap_ulong(v2);
    v10[3] = byteswap_ulong(v3);
    do
    {
        v10[v5] = sub_5593B1D9543B(v10[cout], v10[cout + 1], v10[cout + 2], v10[cout + 3]);
        ++cout;
        ++v5;
    }
    while ( v5 <= 29 );
    v9[0] = (unsigned int)v10[26] >> 24;
    v9[1] = BYTE2(v10[26]);
    v9[2] = BYTE1(v10[26]);
    v9[3] = LOBYTE(v10[26]);
    v9[4] = (unsigned int)v10[27] >> 24;
    v9[5] = BYTE2(v10[27]);
    v9[6] = BYTE1(v10[27]);
    v9[7] = LOBYTE(v10[27]);
    v9[8] = (unsigned int)v10[28] >> 24;
    v9[9] = BYTE2(v10[28]);
    v9[10] = BYTE1(v10[28]);
    v9[11] = LOBYTE(v10[28]);
    v9[12] = v11 >> 24;
    v9[13] = BYTE2(v11);
    v9[14] = BYTE1(v11);
    v9[15] = (unsigned __int8)v11;
    for ( i = 0; i <= 15; ++i )
    {
        if ( v9[i] != v8[i] )
            return 0xFFFFFFFFLL;
    }
    return 1LL;
}
```

[Asm] 纯文本查看 复制代码

```
__int64 __fastcall sub_5593B1D9543B(int a1, int a2, int a3, unsigned int a4)
{
    return a1 ^ (unsigned int)sub_5593B1D95464(a2 ^ a3 ^ a4);
}
__int64 __fastcall sub_5593B1D95464(unsigned int a1)
{
    int v1; // ST18_4
    int v3[290]; // [rsp+20h] [rbp-490h]
    unsigned __int64 v4; // [rsp+4A8h] [rbp-8h]

    v4 = __readfsqword(0x28u);
    qmemcpy(v3, &byte_5593B1D95940, 1152uLL);
    v1 = (v3[BYTE2(a1)] << 16) | v3[(unsigned __int8)a1] | (v3[BYTE1(a1)] << 8) | (v3[a1 >>
    return __ROL4__(v1, 12) ^ (unsigned int)(__ROL4__(v1, 8) ^ __ROR4__(v1, 2)) ^ __ROR4__(v
}
```

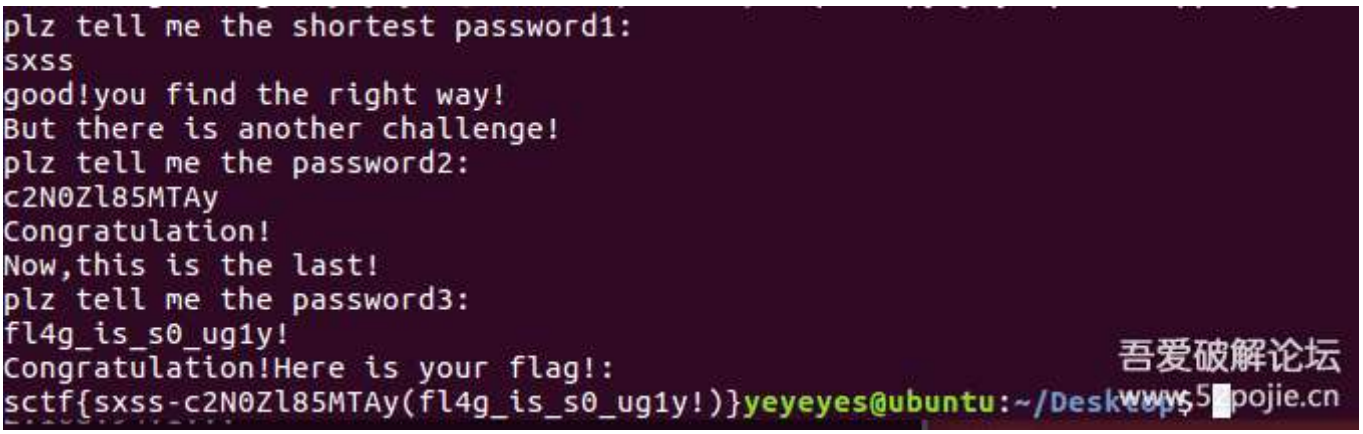
输入字符，长度为16,分为四位一组，组成4个int元素，再大小端颠倒。然后第2,3,4元素异或进入函数索引字符再

循环异或再跟第一个元素异或得到第五个元素，再用2,3,4,5重复相同步骤得到第六个元素，以此类推。
循环26次，取最高四个元素跟程序的值对比，可以写脚本逆推。

[Asm] 纯文本查看 复制代码

```
i   = [0xd8bf92ef,0x9fcc401f,0xc5af7647,0xbe040680]

asc = [0xD6,0x90,0xE9,0xFE,0xCC,0xE1,0x3D,0xB7,0x16,0xB6,0x14,0xC2,0x28,0xFB,0x2C,0x05,0x2B,

def xors(xor_mix):
        temp1 = (xor_mix >> 24)&0xff
        temp2 = (xor_mix >> 16)&0xff
        temp3 = (xor_mix >> 8)&0xff
        temp = (xor_mix)&0xff

        temp_q = asc[temp] | (asc[temp1]<<24)|(asc[temp2]<<16)|(asc[temp3]<<8)
        print ("%x"%(temp_q))
        temp_i = ((temp_q <<12|temp_q>>20)&0xffffffff) ^ ((temp_q <<8|temp_q>>24)&0xffffffff
        return temp_i

print ("%x"%(xors(0x6011F432)))
print (len(asc))

for q in range(26):
        i.append(i[q]^xors(i[q+1]^i[q+2]^i[q+3]))

print (i)
print ("%x"%(i[-1]))
print ("%x"%(i[-2]))
print ("%x"%(i[-3]))
print ("%x"%(i[-4]))
# fl4g_is_s0_ug1y!
```

得到fl4g_is_s0_ug1y!



后面几道题的平台没研究过，有时间再研究。

免费评分

	吾爱币	热心值	理由	收起
红哥	+ 1	+ 1	谢谢@Thanks!	

瑟瑟发抖小菜虾	+ 1	+ 1	我很赞同!
zhukai055	+ 1	+ 1	我很赞同!

查看全部评分

收藏 2 淘帖

发帖前要善用【论坛搜索】功能，那里可能会有你要找的答案或者已经有人发布过相同内容了，请勿重复发帖。

回复 举报

瑟瑟发抖小菜虾 2019-6-24 21:25 沙发

大佬写的博客太好了。。。。。 想前排请问一下大佬 第一个题的脚本有没有 别的方法（感觉这个有点麻烦鸭）

【吾爱破解论坛总版规】 - [让你充分了解吾爱破解论坛行为规则]

回复 支持 举报

yechen123 2019-6-25 00:03 3#

吾爱破解论坛没有任何官方QQ群，禁止留联系方式，禁止任何商业交易。

> 瑟瑟发抖小菜虾 发表于 2019-6-24 21:25
> 大佬写的博客太好了。。。。。 想前排请问一下大佬 第一个题的脚本有没有 别的方法（感觉这个有点麻烦鸭）

应该有吧，只不过我对安卓平台接触不多不知道。

如何升级？如何获得积分？积分对应解释说明！

回复 支持 举报

返回列表

高级模式

您需要登录后才可以回帖 登录 | 注册[Register] 用QQ帐号登录

发表回复 警告：禁止回复与主题无关内容，违者重罚！ 回帖并转播 回帖后跳转到最后一页