[原创]2019看雪CTF 晋级赛Q2 第10题-开启时间之轮 精

ODPan 5

举报

2019-6-24 12:06 981

写在前面

这是一道数论高配题。整数要用大整数，方程要用模方程，模方程要用二次模方程，二次模方程要用二次模合数方程，求逆要选离散对数。感谢作者大神，学习了。

该题选用了mbedtls大数库，可以对应着源代码查看分析。

程序的初步分析

作者一如既往的对话框程序，我们需要关心两个程序。sub_403B00 输入字符处理函数和sub_404270校验函数。校验函数做了抗F5小处理。将此处Nop掉即可F5分析。

.text:00404B1E pop ebp

输入字符处理

简单查看了一下。合法字符串使用“XXXX”进行分割，分为前后两个部分，并转化为string类型。

```
char __cdecl strProcess(char *key, ctf10_Str *strRetKeyBefore, ctf10_Str *strRetKeyAfter)
{
  char *pchar; // eax
  unsigned __int64 XXXXlen; // ST04_8
  char *index; // eax
  char *index_1; // esi
  ctf10_Str *strKeyBefore; // eax
  ctf10_Str *strTemp; // eax
  unsigned int keyAfterLen; // ecx
  _BYTE *pchar_keyAfter; // eax
  int v11; // edi
  char *pchar_keyAfter_1; // eax
  char v13; // al
  char v14; // al
  char v15; // al
  char result; // al
  char v17; // al
  ctf10_Str strXXXX; // [esp+Ch] [ebp-3Ch]
  ctf10_Str strKey; // [esp+1Ch] [ebp-2Ch]
  ctf10_Str strKeyAfter; // [esp+2Ch] [ebp-1Ch]
  int v21; // [esp+44h] [ebp-4h]

  LOBYTE(strKey.field_0) = (_BYTE)key;
  StrInitByFalg(&strKey, 0);
  strLoad(&strKey, key, strlen(key));
  LOBYTE(strXXXX.field_0) = (_BYTE)key;
  v21 = 0;
  StrInitByFalg(&strXXXX, 0);
  strLoadByInput(&strXXXX, 0, 4u, 'X');
  pchar = (char *)strXXXX.pchar;
  LOBYTE(v21) = 1;
  if ( !strXXXX.pchar )
    pchar = (char *)&unk_4100FC;
  HIDWORD(XXXXlen) = strXXXX.len;
  LODWORD(XXXXlen) = 0;
  index = str_find(&strKey, pchar, XXXXlen);    // find XXXX
  index_1 = index;
  if ( index == (char *)-1 )
    goto LABEL_39;
  strKeyBefore = str_substr(&strKey, &strKeyAfter, 0, (unsigned int)index);
  LOBYTE(v21) = 2;
  strCpy(strRetKeyBefore, strKeyBefore, 0, 0xFFFFFFFF);
  LOBYTE(v21) = 1;
  StrInitByFalg(&strKeyAfter, 1);
  strTemp = str_substr(&strKey, &strKeyAfter, (unsigned int)&index_1[strXXXX.len], 0xFFFFFFFF);
  LOBYTE(v21) = 3;
  strCpy(strRetKeyAfter, strTemp, 0, 0xFFFFFFFF);
  LOBYTE(v21) = 1;
  StrInitByFalg(&strKeyAfter, 1);
  if ( !strRetKeyBefore->len )
    goto LABEL_39;
  keyAfterLen = strRetKeyAfter->len;
  if ( !pchar_keyAfter )
    pchar_keyAfter = &unk_4100FC;
  if ( *pchar_keyAfter == '0' )
  {
LABEL_39:
    LOBYTE(v21) = 0;
    StrInitByFalg(&strXXXX, 1);
    v21 = -1;
    StrInitByFalg(&strKey, 1);
    return 0;
  }
  v11 = 0;
  if ( keyAfterLen > 0 )
  {
    while ( 1 )
    {
      pchar_keyAfter_1 = (char *)strRetKeyAfter->pchar;
      if ( !pchar_keyAfter_1 )
        pchar_keyAfter_1 = (char *)&unk_4100FC;
      if ( !isdigit(pchar_keyAfter_1[v11]) )    // 数字
        break;
      if ( (unsigned int)++v11 >= strRetKeyAfter->len )
        goto LABEL_14;
    }
    if ( strXXXX.pchar )
    {
      v14 = *(_BYTE *)(strXXXX.pchar - 1);
      if ( v14 && v14 != -1 )
        *(_BYTE *)(strXXXX.pchar - 1) = v14 - 1;
      else
        strFree((LPVOID)(strXXXX.pchar - 1));
    }
    strXXXX.pchar = 0;
    strXXXX.len = 0;
    strXXXX.field_C = 0;
    if ( strKey.pchar )
    {
      v15 = *(_BYTE *)(strKey.pchar - 1);
      if ( v15 && v15 != -1 )
      {
        *(_BYTE *)(strKey.pchar - 1) = v15 - 1;
        result = 0;
      }
      else
      {
        strFree((LPVOID)(strKey.pchar - 1));
        result = 0;
      }
      return result;
    }
    return 0;
  }
LABEL_14:
  if ( strXXXX.pchar )                          // 全是数字的处理
  {
    v13 = *(_BYTE *)(strXXXX.pchar - 1);
    if ( v13 && v13 != -1 )
      *(_BYTE *)(strXXXX.pchar - 1) = v13 - 1;
    else
      strFree((LPVOID)(strXXXX.pchar - 1));
  }
  strXXXX.pchar = 0;
  strXXXX.len = 0;
  strXXXX.field_C = 0;
  if ( strKey.pchar )
  {
    v17 = *(_BYTE *)(strKey.pchar - 1);
    if ( v17 && v17 != -1 )
    {
      *(_BYTE *)(strKey.pchar - 1) = v17 - 1;
      return 1;
    }
    strFree((LPVOID)(strKey.pchar - 1));
  }
  return 1;
}
```

校验函数

该函数要满足3个条件，才能返回正确结果。下面分别分析这三个条件。

函数初始化部分

大整数初始化

```
mbedtls_mpi_lset(&power0xff, v91);
BigInt_pow(&ret, &power0xff, v7);              // 2^0xff
mbedtls_mpi_add_int(&power0xff, &ret, v8);     // 2^0XFF - 0X13
mbedtls_mpi_sub_int(&powerSbu1, &power0xff, 1);// /2^0XFF - 0X13 - 1
mbedtls_mpi_lset(&A, a4);
```

将输入字串的两部分转化为大整数，我们假设a=keyBefore，b= keyAfter

```
mbedtls_mpi_copy(&keyBefore, &mytarget);
mbedtls_mpi_copy(&keyAfter, &srcNode);
```

第一个条件：4b < power0xff

```
mbedtls_mpi_add_mpi(&doubleAfter, &keyAfter, &keyAfter);
mbedtls_mpi_add_mpi(&node4b, &doubleAfter, &doubleAfter);
v12 = mbedtls_mpi_cmp_mpi(&node4b, &power0xff);  // 4b < power0xff
```

第一个条件是个限制条件，4*b < 2^0xff – 0x13。

第二个条件：模方程

```
    mbedtls_mpi_lset(&sum, v12 >= 0);         // v90 = 0
      v13 = 0;
      v63 = 0;
      v14 = v50;
      while ( v13 < v14 )                       // v14= 6
      {
        mbedtls_mpi_sub_mpi(&afterSunBefor, &keyAfter, &keyBefore); r=b-a
        powern(&desNode[v13], &afterSunBefor, *(&n + v13), &power0xff);// n=0,1,2,3,4,5
        addNum(&v58, *(&num + v13), &desNode[v13], &sum, &power0xff);// a*num+v90  num = 3,0,1,0,0x40,0,1
        mbedtls_mpi_copy(&sum, &v58);
        v63 = ++v13;
      }
      mbedtls_mpi_sub_mpi(&a1a, &keyBefore, &sum);
```

这里我们另r=b-a，循环6次的过程如下

sum0 = r^0/N   *3 =3

sum1 = r^1*0/N   +sum0/N = 3

sum2 = r^2*1/N   +sum1/N = (r^2+3)/N

sum3 = r^3*0/N   +sum2/N = (r^2+3)/N

sum4 = r^4*0x40/N +sum3/N = 0X40r^4/N + (r^2+3)/N

sum5 = r^5*0/N +sum4

最后得到的方程

(64r^4/N + (r^2 +3)/N)/n = a

这里我们另x=r^2，这里往回逆的时候要逆两次。化简后可得一个二次模方程。

64x^2 + x +3-a = 0(mod N)

从方程可知，求出a即可反推x，r最后求得b。那么我们继续分析，看如何求a值。

第三个条件：离散对数

```
mbedtls_mpi_lset(&E, 0);
maxTableData = getDataFormTable((char *)'X');// v15 = 0x19
maxTableData_1 = maxTableData;
maxTableData_2 = maxTableData;
index = 0;
index_1 = 0;
const0xA = *((_DWORD *)checkData + 4);
buf = 0;
memset(&v39, 0, 252u);
v40 = 0;
v41 = 0;
size = *((_DWORD *)checkData + 1);
pcharend = (char *)(size + *(_DWORD *)checkData - 1);
v36 = (char *)(size + *(_DWORD *)checkData - 1);
pbuf = &buf;
v42 = &buf;
while ( (unsigned int)pcharend >= *(_DWORD *)checkData )
{
  *pbuf++ = *pcharend;
  v42 = pbuf;
  v36 = --pcharend;
}
buf += checkData[12];                   // 输入字符串前半部分keyBefore，最后一个字符+0xA
while ( index < strlen(&buf) )
{
  keyChar1 = (char *)*(&buf + index++);
  index_1 = index;
  numFromTableByKey = getDataFormTable(keyChar1);
  numFromTableByKey_1 = numFromTableByKey;
  v43 = numFromTableByKey;
  if ( numFromTableByKey >= maxTableData_1 )
  {
    errorFlag = 1;
    break;
  }
  mbedtls_mpi_mul_int(&E, &E, maxTableData_1);
  mbedtls_mpi_add_int(&E, &E, numFromTableByKey_1);// 将keyBfore内容转换为0x19进制数
}
if ( index <= const0xA && !errorFlag )  // KEY 前半部分长度小于10
{
  sushuCnt = createSuShuByRand(randNum, &buf2);// 产生素数列表和随机检测个数
  randNum = sushuCnt;
  for ( j = 0; ; ++j )                  // 米勒罗宾素性测试
  {
    v63 = j;
    if ( j >= sushuCnt || !*(&buf2 + j) )
      break;
    sub_4025A0(&ret_1, &E, *(&buf2 + j));
    if ( !ret_1 )
      goto LABEL_35;
  }
  get_gccd(&v58, &powerSbu1, &E);       // gcd
  if ( node_getBitNum_0(&v58) <= 1 )
  {
    mbedtls_mpi_inv_mod(&D, &E, &powerSbu1);// D = E^-1 mod (N-1)
    mbedtls_mpi_exp_mod(&final, &A, &D, &power0xff, &a5);// X = A^D mod N
    mbedtls_mpi_sub_mpi(&v71, &nodeCheck, &final);
    v25 = node_getData(&v71);
    result3 = v25;
    mbedtls_mpi_exp_mod(&a2, &nodeCheck, &E, &power0xff, &a5);// A = X^E mode N
    mbedtls_mpi_sub_mpi(&v69, &a2, &A);
    if ( v25 )
      result3 = node_getData(&v69);
  }

```

代码几个关键步骤说明：

1、 将keyBefore（输入字符串得前半部分）最后一位+0x0A；

2、 将keyBfore内容转换为0x19进制数，记作E；

3、 米勒罗宾素性测试，确保E为素数；

4、 E与(N-1)最大公约数为1

5、 检测计算

D = E^-1 mod (N-1)

X = A^D mod N

A = X^E mode N

N=0x7FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFED

这里我们已知A和X得四组数据

```
100',0Ah
9230197858975018299629857977411527954550899478307510809210520967346958600039
'101',0Ah
5041422176735208376561349852467459084433382372025565643249055786677248860034
.102',0Ah
38377684164112914669201831650756813551072223314592288217929947158283532270268
'103',0Ah
13436195533519778671648120865743178010431697022400670384909515001970400645091
```

求D，求离散对数。利用作者dlp工具可以跑出D。

这样我们可以一路反推D->E->keyBefore->a

将a带回二次模方程，这里简单提一下方程化简过程

64r^4 + r^2 + 3-0x4B435446524541445955 = 0 mod N

64x^2 + x + 3-0x4B435446524541445955 = 0 mod N

转换为一般模方程

x^2 = A mode M　 A是奇数非素数　M是个合数　GCD（A,M）==1

其中

a = 64

b = 1

c =-0x4B435446524541445952

所以

A = b^2 - 4ac

M = 4aN

这样经过两次二次模方程求解可以解的r（b-a，b<4N。最终求得a，b

aXXXXb = KCTFREADYKXXXX1548396171915056368526513804948765619094392315806578461796159505215278288254

[公告]看雪.纽盾 KCTF 2019晋级赛Q3攻击方规则，9月10日开赛，华为P30 Pro、iPad、kindle等你来拿！

0 ☆ 收藏　0 赞　打赏　分享

最新回复 (0)

勇士小蓝

内容

回帖　表情　高级回复

返回

首页　论坛　专栏　课程　发现