

词法分析测试报告

陈童

2019211283

304班

```
tongchen@TongdeMacBook-Pro:~/Library/Mobile Documents/com~apple~CloudDocs/Docu... 1
> ./LSA>test1.txt -s CreateFileMapping.cpp
> ./LSA>test2.txt -s CreateRemoteThread.cpp
> ./LSA>test2.txt DeleteRecentFileCache.cpp
> ./LSA>test3.txt DeleteRecentFileCache.cpp
> ./LSA>test2.txt -s CreateRemoteThread.cpp
~/L/Mo/com~apple~C/Doc/C/Lexical_Syntax_Analysis/build master !18 77 > Py base 08:53:33
```

通过如下命令，对CreateFileMapping.cpp，CreateRemoteThread.cpp，DeleteRecentFileCache.cpp三个文件（随机选自Github代码库，来源<https://github.com/3gstudent/Homework-of-C-Language>），生成test1.txt, test2.txt, test3.txt三个输出文件。下对此进行分析。

CreateFileMapping.cpp

```
CreateFileMapping.cpp 9+, U x
Lexical_Syntax_Analysis > build > CreateFileMapping.cpp > main
1  #include <windows.h>
2  #include <stdio.h>
3  #include <conio.h>
4  #include <tchar.h>
5  #pragma comment(lib, "Advapi32.lib")
6
7  #define BUF_SIZE 256
8
9  int main(int argc, char *argv[])
10 {
11     if (argc != 2)
12     {
13         printf("\nCreateFileMapping\n\n");
14         printf("Usage:\n");
15         printf("%s <string>\n", argv[0]);
16         return 0;
17     }
18
19     HANDLE hMapFile1, hMapFile2;
20     char *pBuf;
21     char *pBuf2;
22     char szName1[] = "Global\\SharedMappingObject1";
23     char szName2[] = "Global\\SharedMappingObject2";
24     DWORD EventRecordID = 32;
25     DWORD offset = 0x11;
26
27     char szOffset[8];
28     sprintf_s(szOffset, "%d", offset);
29
30     printf("[*]Try to set SecurityDescriptor... ");
31
32     PSECURITY_DESCRIPTOR pSec = (PSECURITY_DESCRIPTOR)LocalAlloc(LMEM_FIXED, SECURITY_DESCRIPTOR_MIN_LENGTH);
33     if (!pSec)
34     {
35         return GetLastError();
36     }
37     if (!InitializeSecurityDescriptor(pSec, SECURITY_DESCRIPTOR_REVISION))
38     {
39         LocalFree(pSec);
40         return GetLastError();
41     }
```

通过LSA输出为

```

Usage ID
;
"
n ID
"
)
;
"
printf ID
(
"
s ID
relop LT
string ID
relop GT
n ID
"
;
argv ID
0 NUM
"
)
;
"
return KEYWORD
0 NUM
;
"
}
"
HANDLE ID
hMapFile1 ID
;
hMapFile2 ID
;
"
char ID
*
pBuf ID
;
"
char ID
*
pBuf2 ID
;
;
char ID
szName1 ID
"
"
relop EQ
"
Global ID
"
SharedMappingObject1 ID
"
;
"
char ID
szName2 ID
"
"
relop EQ
"
Global ID
"
"
SharedMappingObject2 ID
"
;
"
DWORD ID
EventRecordID ID
relop EQ
32 NUM
;
"

```

```

)
;
printf ID
(
"
"
+
+
You ID
can ID
input ID
something ID
to ID
stop ID
waiting ID
"
n ID
)
;
;
;
getch ID
(
)
;
;
printf ID
(
"
"
*
*
Free ID
"
n ID
"
)
;
;
;
LocalFree ID
{
pSec ID
}
;
;
;
UnmapViewOfFile ID
{
pBuf ID
}
;
;
;
CloseHandle ID
{
hMapFile1 ID
}
;
;
;
UnmapViewOfFile ID
{
pBuf2 ID
}
;
;
;
CloseHandle ID
{
hMapFile2 ID
}
;
;
;
return KEYWORD
0 NUM
;
;
}
Error: INVALID WORD: 0x11

NUM: 19
ID: 236
KEYWORD: 20
OP: 522
COMMENT: 0(Lines)
ERROR: 1

```

在对应代码块和输出部分可看出，LSA实现了一定的错误处理能力

```
;
"
DWORD ID
EventRecordID ID|
relop EQ
32 NUM
;
"
DWORD ID
offset ID
relop EQ
;
"
char ID
szOffset ID
"
8 NUM
"
;
"
sprintf_s ID
```

```
DWORD EventRecordID = 32;
DWORD offset = 0x11;
```

CreateRemoteThread.cpp

```
CreateRemoteThread.cpp 9+, U x
Lexical_Syntax_Analysis > build > CreateRemoteThread.cpp > ...
> 0x

1  #include <windows.h>
2  #include <stdio.h>
3  #include <tlhelp32.h>
4  #pragma comment(lib, "Advapi32.lib")
5
6  BOOL InjectDll(UINT32 ProcessId, char *DllPath)
7  {
8      if (strstr(DllPath, "\\") != 0)
9      {
10         printf("[!]Wrong Dll path\n");
11         return FALSE;
12     }
13     if (strstr(DllPath, "\\") == 0)
14     {
15         printf("[!]Need Dll full path\n");
16         return FALSE;
17     }
18
19     size_t len = strlen(DllPath) + 1;
20
21     LPVOID pThreadData = NULL;
22     HANDLE ProcessHandle = NULL;
23     HANDLE hThread = NULL;
24     BOOL bRet = FALSE;
25
26     try
27     {
28         ProcessHandle = OpenProcess(PROCESS_ALL_ACCESS, FALSE, ProcessId);
29         if (ProcessHandle == NULL)
30         {
31             printf("[!]OpenProcess error\n");
32             __leave;
33         }
34     }
```

DeleteRecentFileCache.cpp

```
Lexical_Syntax_Analysis > build > DeleteRecentFileCache.cpp > DeleteRecord

1  #include <windows.h>
2  #pragma pack(1)
3
4  typedef struct _BCF_HEADER {
5      ULONG64 Flag1;
6      ULONG64 Flag2;
7      ULONG Unknown;
8  } BCFHEADER, *PBCFHEADER;
9
10 typedef struct _BCF_RECORD {
11     ULONG Size;
12 } BCFRECORD, *PBCFRECORD;
13 #pragma pack()
14
15 int NewSize = 0;
16
17 char *DeleteRecord(PVOID mapAddress, char *TempBuf, int StopSize, WCHAR *FileName)
18 {
19     char flag[16] = { 0xFE, 0xFF, 0xEE, 0xFF, 0x11, 0x22, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00 };
20     if (!memcmp(mapAddress, flag, 16))
21     {
22         printf("[!]Maybe it's not RecentFileCache.bcf");
23         exit(0);
24     }
25     memcpy(TempBuf, mapAddress, 0x14);
26     PBCFRECORD currentRecordPtr = NULL;
27     PBCFRECORD nextRecordPtr = (PBCFRECORD)((PBYTE)mapAddress + 0x14);
28     int DeleteSize = 0;
29     int FlagSize = 0x14;
30     while (FlagSize + DeleteSize < StopSize)
31     {
32         currentRecordPtr = nextRecordPtr;
33
34         WCHAR *RecordName = new WCHAR[nextRecordPtr->Size + 1];
35         memcpy(RecordName, nextRecordPtr + 1, nextRecordPtr->Size * 2 + 2);
36         printf("%ws\n", RecordName);
37         if (wcsncmp(RecordName, FileName) == 0)
```

通过错误分析可看出，LSA对字符串支持的不是很好。

```
f("The new file will be saved as NewRec
f("Author:3gstudent\n");
f("Usage:\n");
f("      %s <file path of RecentFileCach
f("eg:\n");
f("      %s C:\\Windows\\AppCompat\\Prog
f("[!]Wrong parameter\n");
'n 0;

fp;
rr = fopen s(&fp, argv[1], "a+");

Error: INVALID WORD: 3gstudent
```



```
test2.txt U X
Lexical_Syntax_Analysis > build > test2.txt
1249 )
1250 ;
1251 "
1252 "
1253 return KEYWORD
1254 1 NUM
1255 ;
1256 "
1257 }
1258 "
1259 printf ID
1260 (
1261 "
1262 "
1263 +
1264 "
1265 InjectDll ID
1266 success ID
1267 "
1268 n ID
1269 "
1270 )
1271 ;
1272 "
1273 return KEYWORD
1274 0 NUM
1275 ;
1276 }
1277 "
1278
1279 NUM: 22
1280 ID: 360
1281 KEYWORD: 34
1282 OP: 861
1283 COMMENT: 0(lines)
1284 ERROR: 0
```