windows样本高级静态分析之识别汇编中C代码结构四

yong夜 / 2019-10-17 09:29:28 / 浏览数 3261 安全技术 二进制安全 顶(0) 踩(0)

引言

通过将一条条指令组合成原始的数据类型完成汇编指令到高级语言结构

目标

掌握数组、链表、结构体等数据类型的汇编指令。

流程

1. 编写源代码，生成对应程序
2. 返汇编程序
3. 分析汇编代码，总结数据类型的特点
4. 小结

实践过程

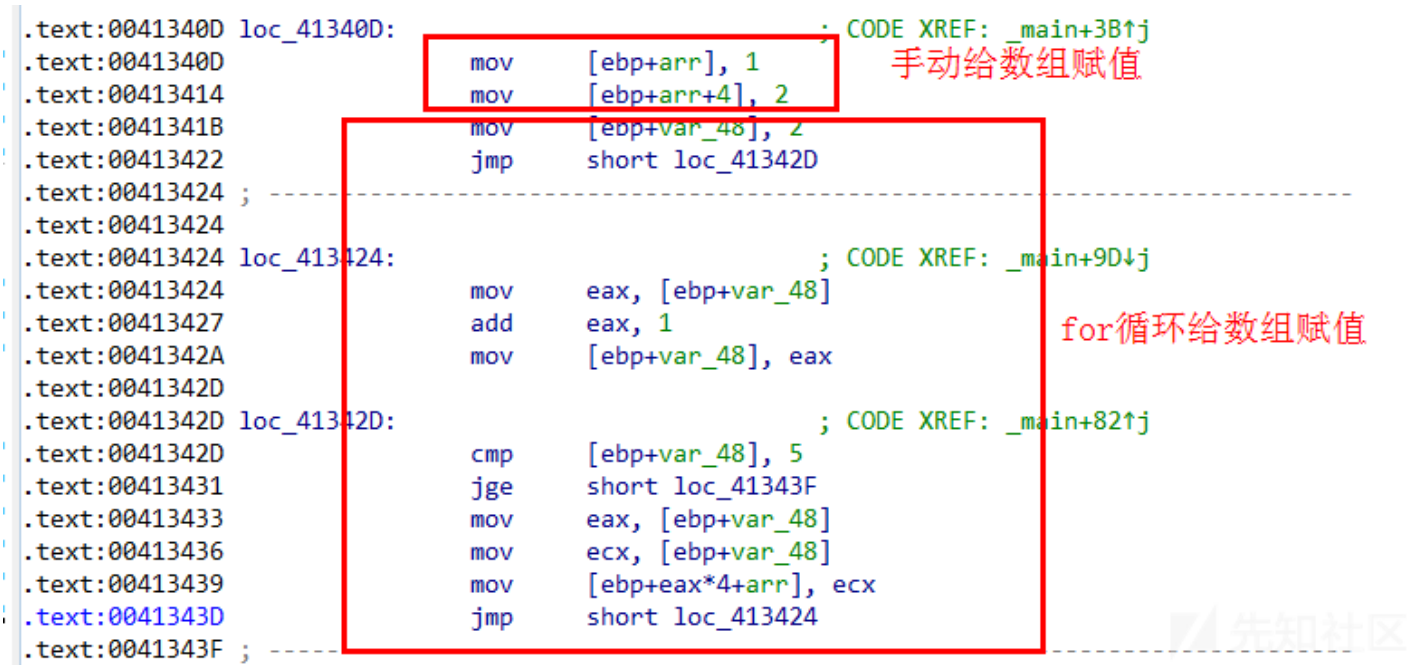
数组类型

- 源代码

```
#include <stdio.h>

void main()
{
   int arr[5];
   arr[0] = 1;
   arr[1] = 2;
   for(int i=2; i<5; i++)
   {
       arr[i] = i;
   }
}
```

- 汇编代码



- 数据类型特点

选区一块内存区域存放数组内容，这里选取的是栈上内存块并且从ebp+arr开始，然后将数据填充到这块内存里。

一块内存上的每个元素长度一致

小结

```
call    ds:__imp__malloc
...
mov     [ebp+eax*4+arr], ecx
```

给一段内存地址赋长度相同的值，看到类似上面这种指令的时候就可以浮现出一个对应数据类型的数组

结构体

- 源代码

```
#include <stdio.h>
#include <stdlib.h>

struct mystruct
{
   int x[5];
   char y;
};

struct mystruct *test;

void main()
{
   test = (struct mystruct *)malloc(sizeof(struct mystruct));
   for(int i=0; i<5; i++)
   {
       test->x[i]= i;
   }
   test->y = 'a';
}
```

- 汇编代码

```
1340D loc_41340D:                               ; CODE XREF: _main+3B↑j
1340D                 mov     esi, esp
1340F                 push    18h               ; Size
13411                 call    ds:__imp__malloc
13417                 add     esp, 4
1341A                 cmp     esi, esp
1341C                 call    j___RTC_CheckEsp
13421                 mov     ?test@@3PAUmystruct@@A, eax ; mystruct * test
13426                 mov     [ebp+var_2C], 0
1342D                 jmp     short loc_413438
1342F ; ---------------------------------------------------------------------------
1342F
1342F loc_41342F:                               ; CODE XREF: _main+AD↓j
1342F                 mov     eax, [ebp+var_2C]
13432                 add     eax, 1
13435                 mov     [ebp+var_2C], eax
13438
13438 loc_413438:                               ; CODE XREF: _main+8D↑j
13438                 cmp     [ebp+var_2C], 5
1343C                 jge     short loc_41344F
1343E                 mov     eax, [ebp+var_2C]
13441                 mov     ecx, ?test@@3PAUmystruct@@A ; mystruct * test
13447                 mov     edx, [ebp+var_2C]
1344A                 mov     [ecx+eax*4], edx
1344D                 jmp     short loc_41342F
1344F ; ---------------------------------------------------------------------------
1344F
1344F loc_41344F:                               ; CODE XREF: _main+9C↑j
1344F                 mov     eax, ?test@@3PAUmystruct@@A ; mystruct * test
13454                 mov     byte ptr [eax+14h], 'a'
13458                 xor     eax, eax
1345A                 pop     edi
1345B                 pop     esi
1345C                 pop     ebx
1345D                 add     esp, 0F0h
13463                 cmp     ebp, esp
13465                 call    j___RTC_CheckEsp
1346A                 mov     esp, ebp
1346C                 pop     ebp
1346D                 retn
1346D _main           endp
1346D
```

1. 申请一块内存

2. for循环给这个内存前20字节空间赋值

前20字节空间是数组类型

3. 给后四字节空间符一个字符

- 特点

malloc出一块内存，然后给这块内存赋不同类型的数据

一个内存上每个元素不全一致

小结

```
mov     ecx, ?test@@3PAUmystruct@@A ;
mov     edx, [ebp+var_2C]
mov     [ecx+eax*4], edx
...
mov     eax, ?test@@3PAUmystruct@@A ; mystruct * test
mov     byte ptr [eax+14h], 'a'
```

malloc得到一块内存后，给其赋不同长度或不同类型的数据

链表

- 源代码

```
#include <stdio.h>
#include <stdlib.h>

struct node
{
   int x;
   struct node * next;
```

```
};

typedef node pnode;

void main()
{
   pnode * curr, * head;
   int i;
   head = NULL;
   for(i = 1; i<=3; i++)
   {
       curr = (pnode *)malloc(sizeof(pnode));
       curr->x = i;
       curr->next = head;
       head = curr;
   }
}
```

- 汇编代码

```
text:004133BC                 rep stosd
text:004133BE                 mov     [ebp+head], 0
text:004133C5                 mov     [ebp+i], 1
text:004133CC                 jmp     short loc_4133D7
text:004133CE ; ---------------------------------------------------------------------------
text:004133CE
text:004133CE loc_4133CE:                             ; CODE XREF: _main+6B↓j
text:004133CE                 mov     eax, [ebp+i]
text:004133D1                 add     eax, 1
text:004133D4                 mov     [ebp+i], eax
text:004133D7
text:004133D7 loc_4133D7:                             ; CODE XREF: _main+2C↑j
text:004133D7                 cmp     [ebp+i], 3
text:004133DB                 jg      short loc_41340D
text:004133DD                 mov     esi, esp
text:004133DF                 push    8               ; Size
text:004133E1                 call    ds:__imp__malloc
text:004133E7                 add     esp, 4
text:004133EA                 cmp     esi, esp
text:004133EC                 call    j___RTC_CheckEsp
text:004133F1                 mov     [ebp+curr], eax
text:004133F4                 mov     eax, [ebp+curr]
text:004133F7                 mov     ecx, [ebp+i]
text:004133FA                 mov     [eax], ecx
text:004133FC                 mov     eax, [ebp+curr]
text:004133FF                 mov     ecx, [ebp+head]
text:00413402                 mov     [eax+4], ecx
text:00413405                 mov     eax, [ebp+curr]
text:00413408                 mov     [ebp+head], eax
text:0041340B                 jmp     short loc_4133CE
text:0041340D ; ---------------------------------------------------------------------------
text:0041340D
```

1. 给一块内存前4字节赋一个整型
2. 给这块内存后四字节赋head变量值
3. 将这块内存首地址赋给head变量

- 特点

malloc一块内存，给这块内存内赋任意元素数据和■■■■■■，这个内存地址指向另一块相同类型的内存。

1. 一个内存块里必须存在一个元素指向另一个相同类型的内存块

上一篇：Laravel框架RCE分析（CV... 下一篇：apk加固工具探究系列——advmp

1. 2 条回复

jzwxZZZ 2019-10-17 11:58:47

希望能合并下几篇的内容。。。。一篇文章的深度和内容感觉都不够，而且这个和样本早就没关系了吧。。。
感觉这一个windows样本分析系列需要浓缩一下，抱着学习的心态进来，有一点小小的失望

0 回复Ta

yong夜 2019-10-17 14:27:14

@jzwxZZZ 谢谢反馈，后面的动态分析的量很足，希望继续关注，一起学习~

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板