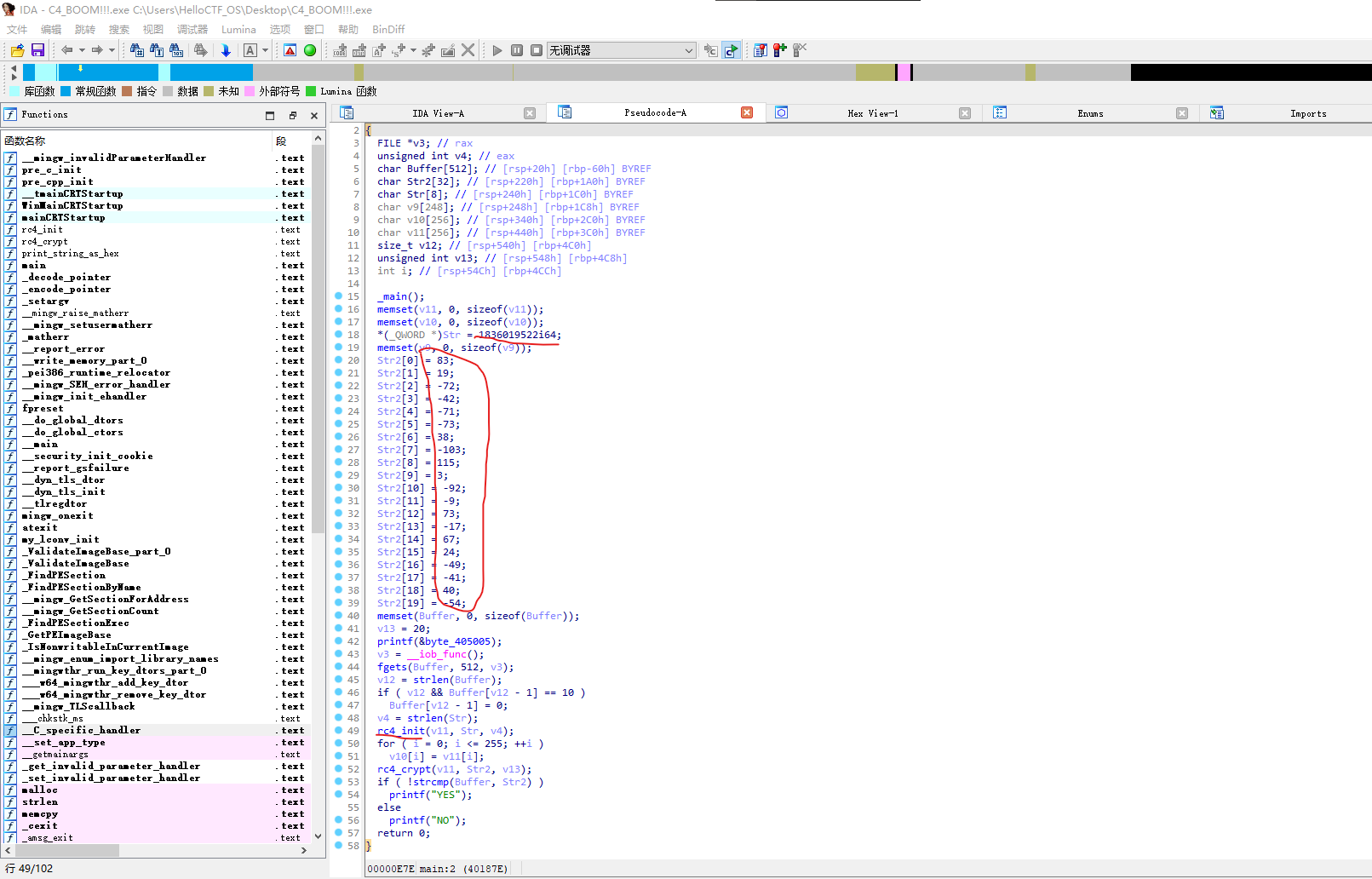
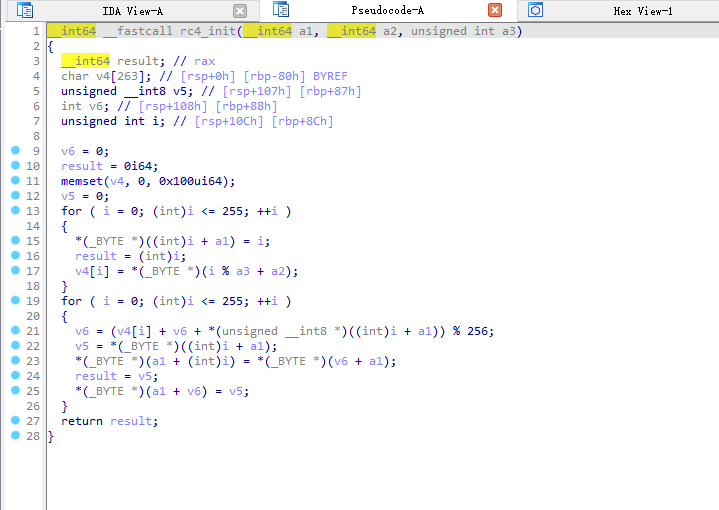
1. 使用DIE分析程序，64位，GCC编译器，无壳。

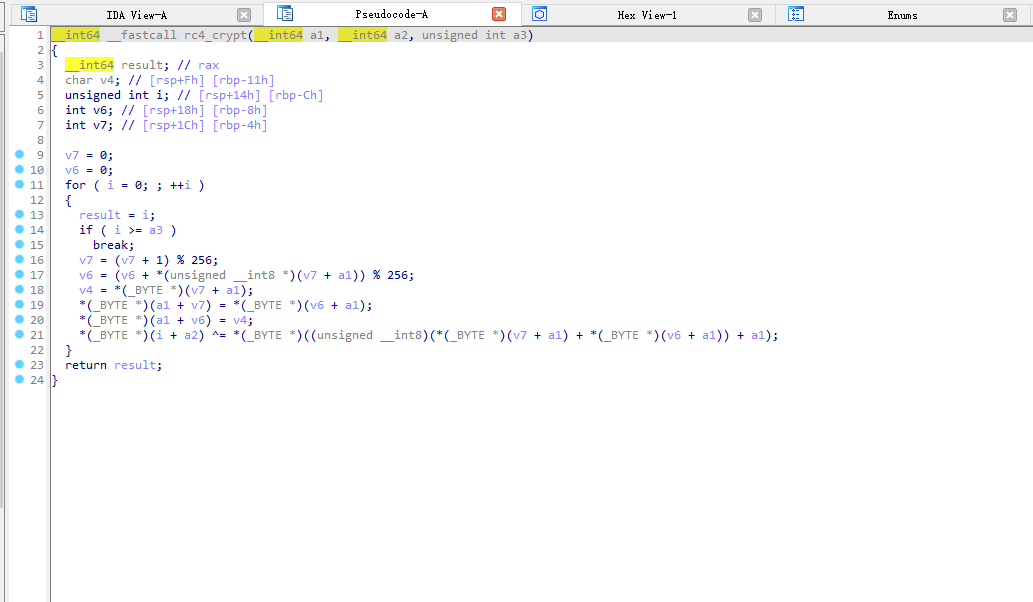


1. IDA静态分析，通过关键字判断是RC4，数组的数据可能是密文，str可能是密钥；

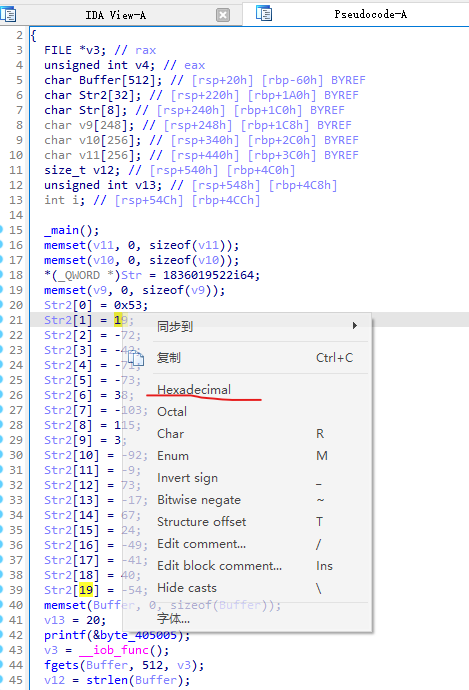


1. 分析出现在main函数中的所有函数

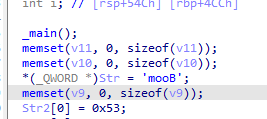




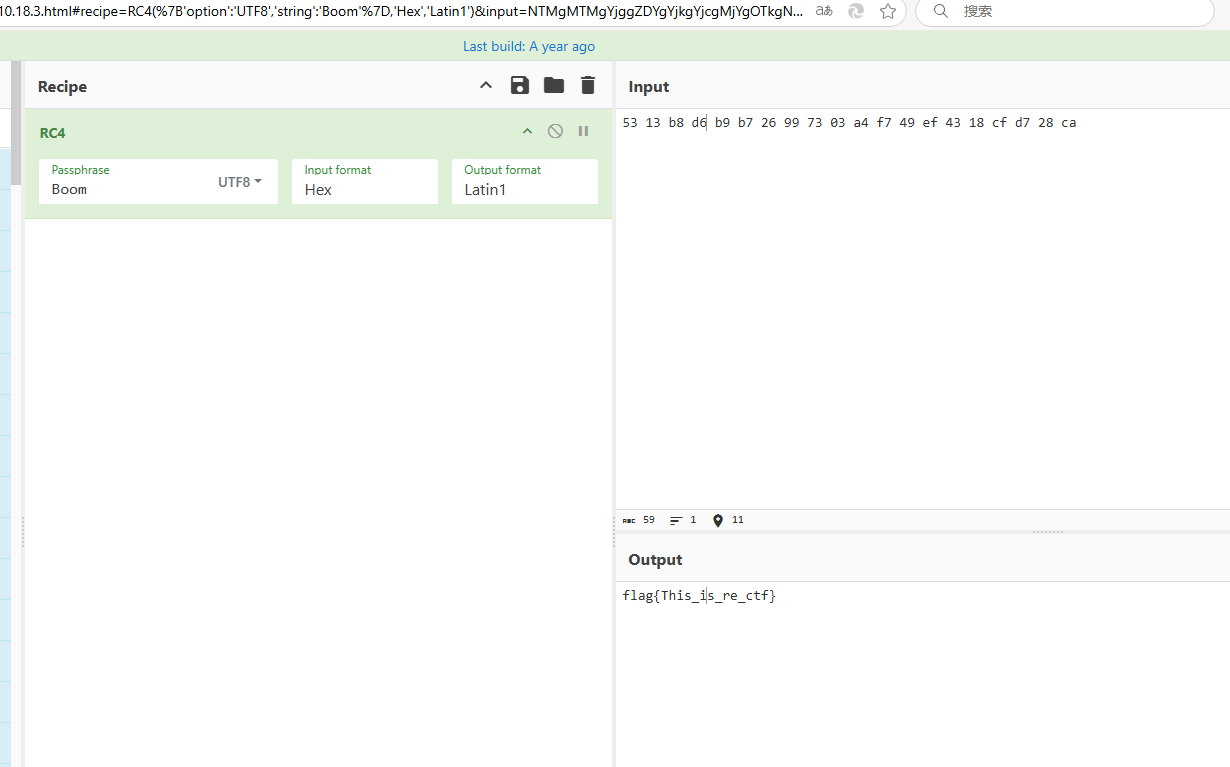
1. 判断是常规的RC4，将main函数中数组的数字转化为十六进制，然后提取即可；



1. 将密钥也转化为char型，发现是“mooB”，涉及大小端的问题，实际是“Boom”；



1. 由于是常规的RC4，这里直接工具一把梭哈，如果是魔改的RC4，就只有动态调试和写脚本的方法；



1. 也可以使用脚本，这里是C语言脚本，RC4本质是xor的一种变式；

#include <stdio.h>

#include <string.h>

typedef unsigned long ULONG;

/\* 初始化函数 \*/

void rc4\_init(unsigned char \*s, unsigned char \*key, unsigned long Len) {

int i = 0, j = 0;

char k[256] = {0};

unsigned char tmp = 0;

for (i = 0; i < 256; i++) {

s[i] = i;

k[i] = key[i % Len];

}

for (i = 0; i < 256; i++) {

j = (j + s[i] + k[i]) % 256;

tmp = s[i];

s[i] = s[j]; // 交换 s[i] 和 s[j]

s[j] = tmp;

}

}

/\* 加解密 \*/

void rc4\_crypt(unsigned char \*s, unsigned char \*Data, unsigned long Len) {

int i = 0, j = 0, t = 0;

unsigned long k = 0;

unsigned char tmp;

for (k = 0; k < Len; k++) {

i = (i + 1) % 256;

j = (j + s[i]) % 256;

tmp = s[i];

s[i] = s[j]; // 交换 s[x] 和 s[y]

s[j] = tmp;

t = (s[i] + s[j]) % 256;

Data[k] ^= s[t];

}

}

void print\_string\_as\_hex(const char \*str) {

size\_t len = strlen(str);

size\_t i;

for (i = 0; i < len; i++) {

// 以十六进制形式输出每个字符，并确保输出两位

printf("%02x", (unsigned char)str[i]);

if (i < len - 1) {

printf(" ");

}

}

printf("\n");

}

int main() {

unsigned char s[256] = {0}, s2[256] = {0}; // S-box

char key[256] = {"Boom"}; // 密钥

char pData[] = {0x53, 0x13, 0xb8, 0xd6, 0xb9, 0xb7, 0x26, 0x99, 0x73, 0x03, 0xa4, 0xf7, 0x49, 0xef, 0x43, 0x18, 0xcf, 0xd7, 0x28, 0xca}; // 密文

char arr[512] = {0}; // 假设这里存储预期结果，这里初始化为 0

unsigned long len = sizeof(pData) / sizeof(pData[0]); // 正确计算密文数据长度

int i;

size\_t arr\_len = strlen(arr);

if (arr\_len > 0 && arr[arr\_len - 1] == '\n') {

arr[arr\_len - 1] = '\0';

}

rc4\_init(s, (unsigned char \*)key, strlen(key)); // 已经完成了初始化

for (i = 0; i < 256; i++) {

s2[i] = s[i];

}

rc4\_crypt(s, (unsigned char \*)pData, len);

printf("%s",pData);

return 0;

}