

二进制漏洞挖掘之Fuzzing

代涛

关于我

- 安全研究员 @ 开源网安研究院
 - 二进制漏洞挖掘、模糊测试等
- 研究成果
 - [Microsoft Windows CVE-2019-1468](#)
 - [Microsoft Windows CVE-2020-0607](#)
 - [Microsoft Windows CVE-2020-0744](#)
 - [Microsoft Windows CVE-2020-0821](#)
 - [Microsoft Windows CVE-2020-0879](#)
 - [Microsoft Windows CVE-2020-1007](#)
 - [Microsoft Windows CVE-2020-1351](#)
 - CVE-2021-21493/CVE-2021-27584/CVE-2021-21461/CVE-2021-21464
 - (以及多个未公开漏洞)
- 刚入门的二进制菜鸡/fuzz、rust语言爱好者
- 个人博客: <https://github.com/xinali/articles>

目录

- 漏洞挖掘
- 模糊测试
- 模糊测试实战案例

漏洞挖掘

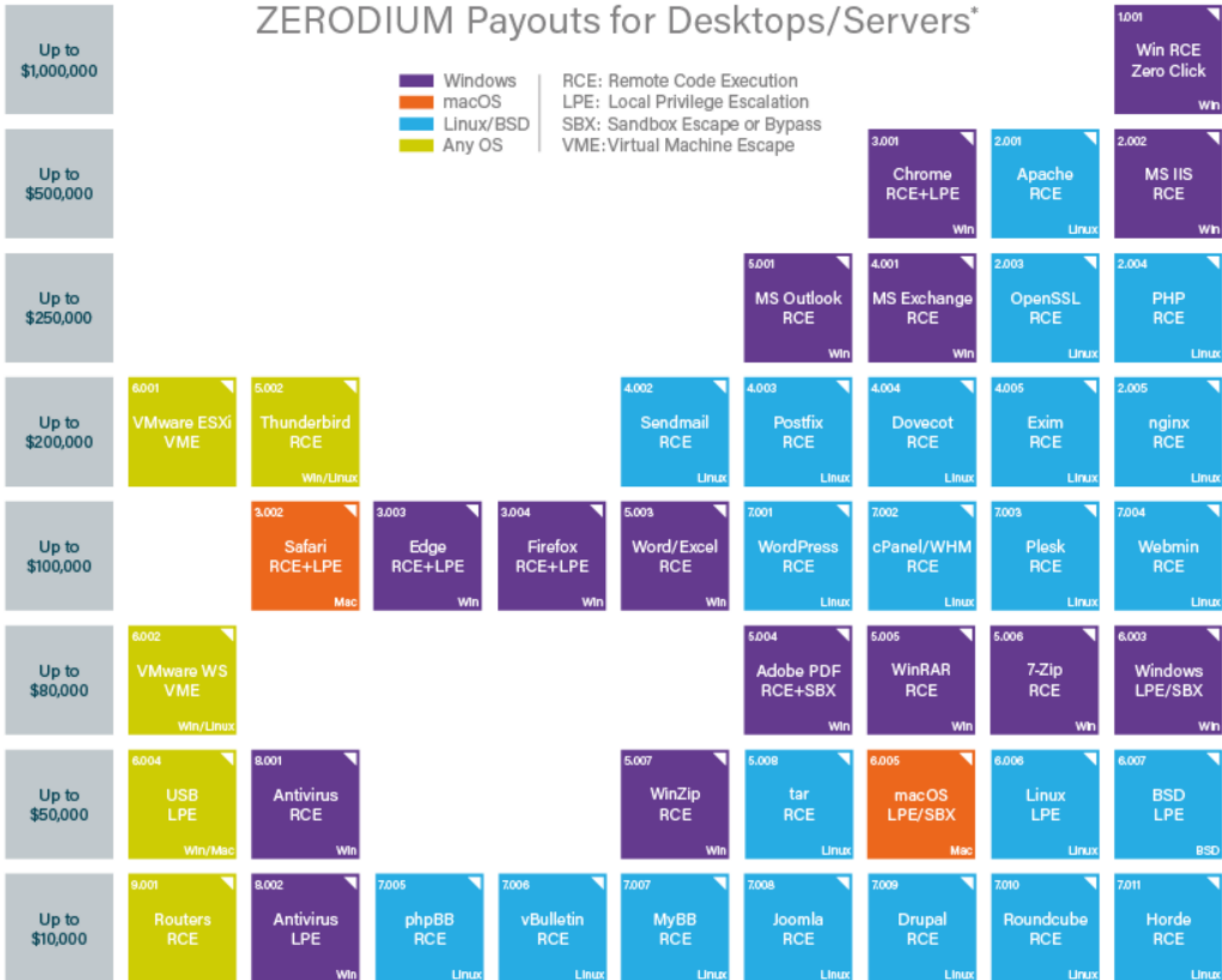
二进制漏洞挖掘why?

- 门槛高，难度大，有意思
- 个人荣誉（CVE，公开致谢）
 - tomkeeper(tk教主)/yuange/heige等等
- 奖金（大笔dollar）

ZERODIUM Payouts for Desktops/Servers*

- Windows
- macOS
- Linux/BSD
- Any OS

RCE: Remote Code Execution
 LPE: Local Privilege Escalation
 SBX: Sandbox Escape or Bypass
 VME: Virtual Machine Escape



二进制漏洞主要类型

- Out-of-bound Read/Write
- Use after free (UAF)
- Double Free
- Integer Overflow
- ...

漏洞挖掘的方式

- 静态分析
 - 白盒源码(strcpy/memcpy等关键函数)
 - 黑盒逆向(汇编/arm等等)
 - IDA pro
- 动态分析
 - Windbg(cdb)/lldb/gdb等等
 - IDA pro
 - Qemu

模糊测试

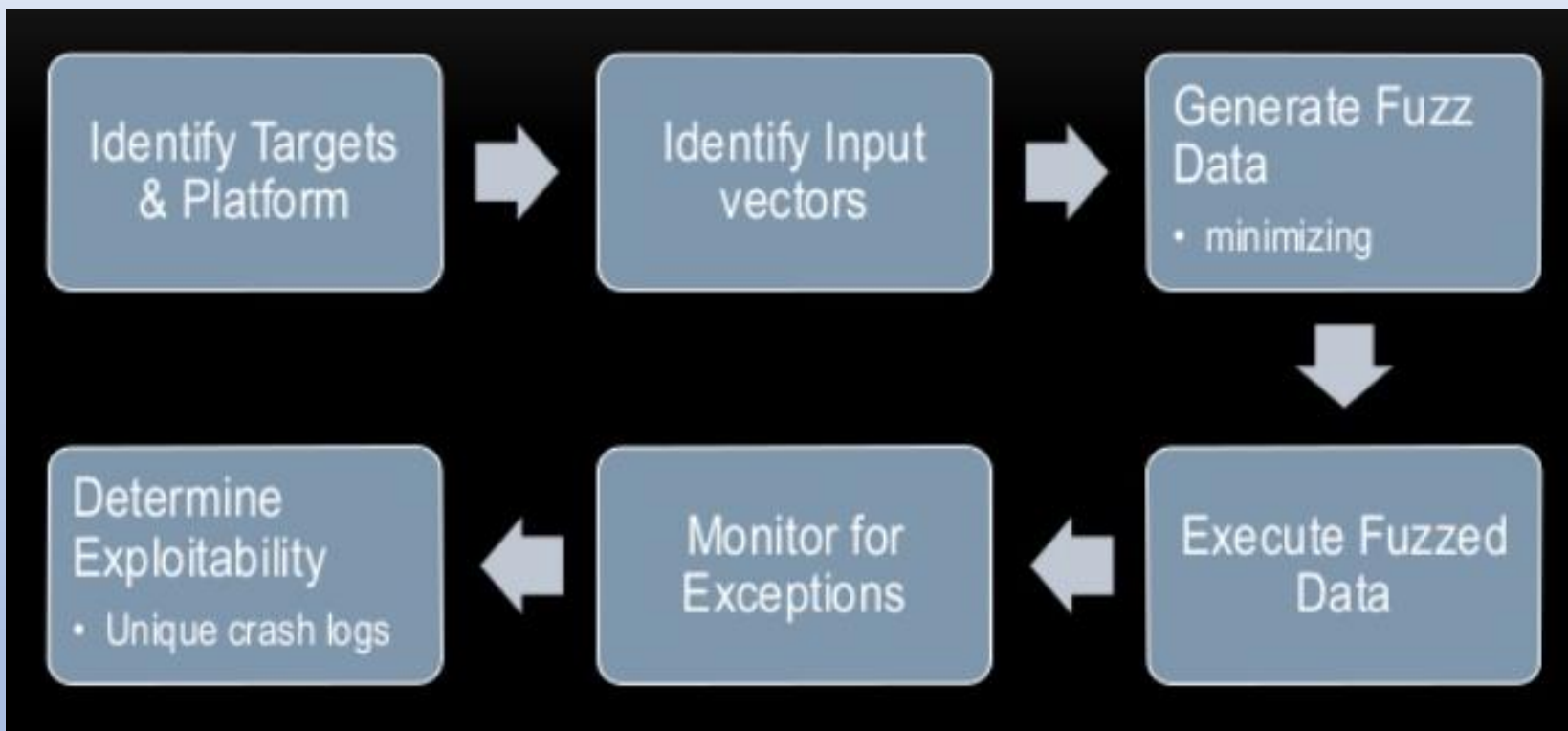
模糊测试why?

- 测试成本低(企业/个人)
- 技术门槛相对较低
- 目前二进制领域漏洞挖掘最有效方式

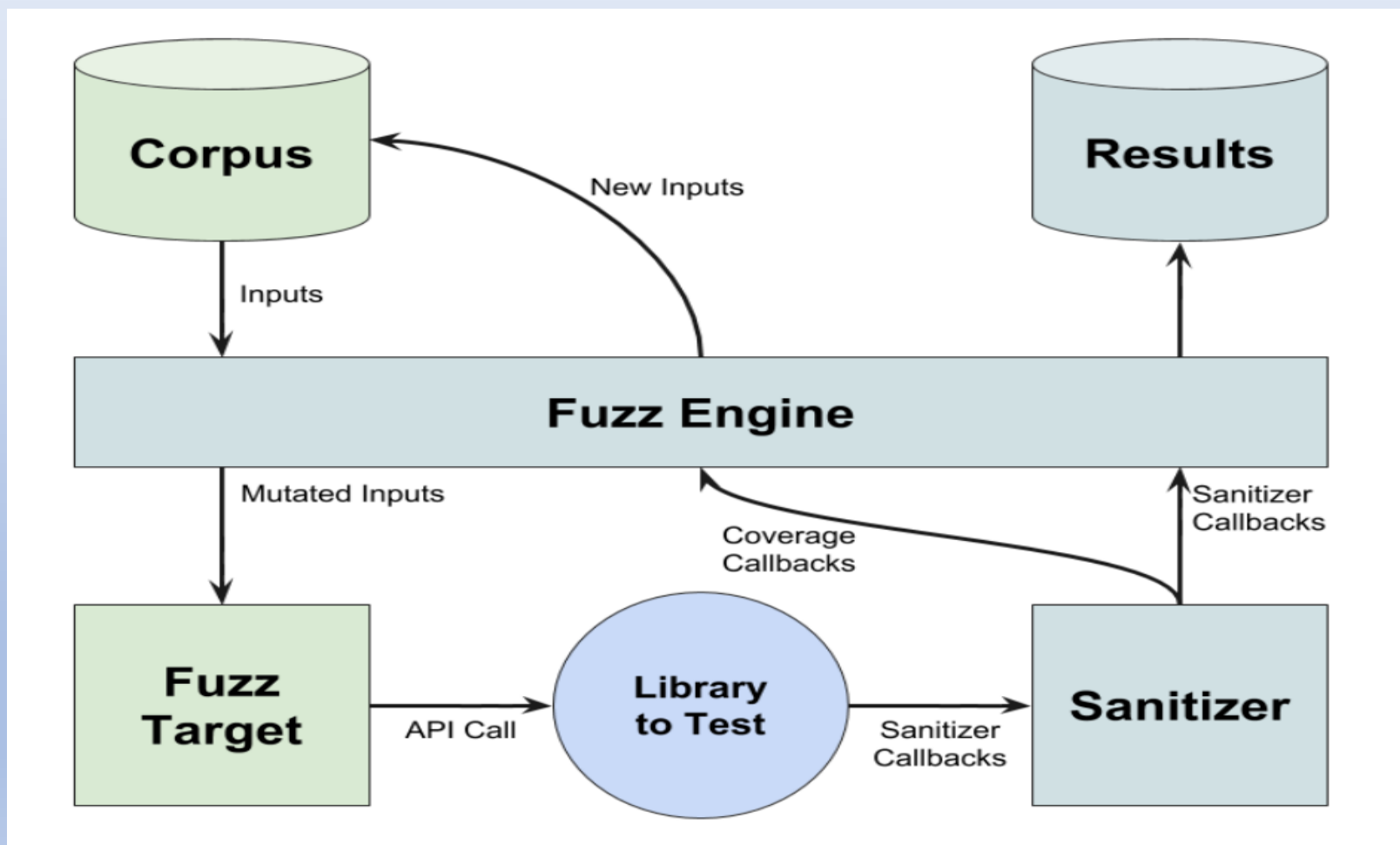
模糊测试

- 模糊测试 (fuzz testing, fuzzing) 是一种软件测试技术。其核心思想是将自动或半自动生成的随机数据输入到一个程序中，并监视程序异常，如崩溃，断言 (assertion) 失败，以发现可能的程序错误，比如内存泄漏。
- 主要测试方式：变异测试 (mutation-based) 以及生成测试 (generation-based)
- 主要测试目标：文件格式与网络协议等
- 简单分类：开源/闭源模糊测试

生成测试



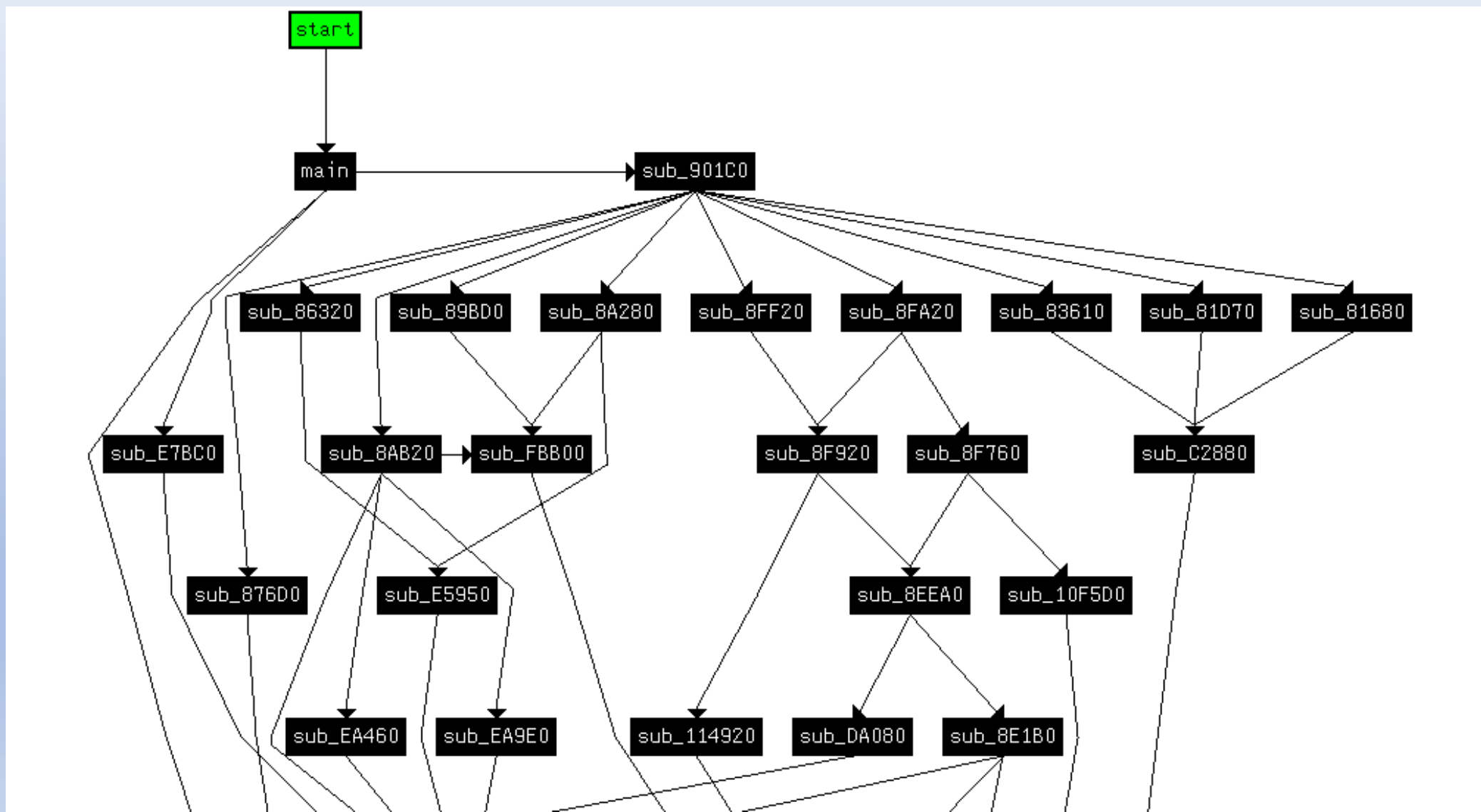
变异测试



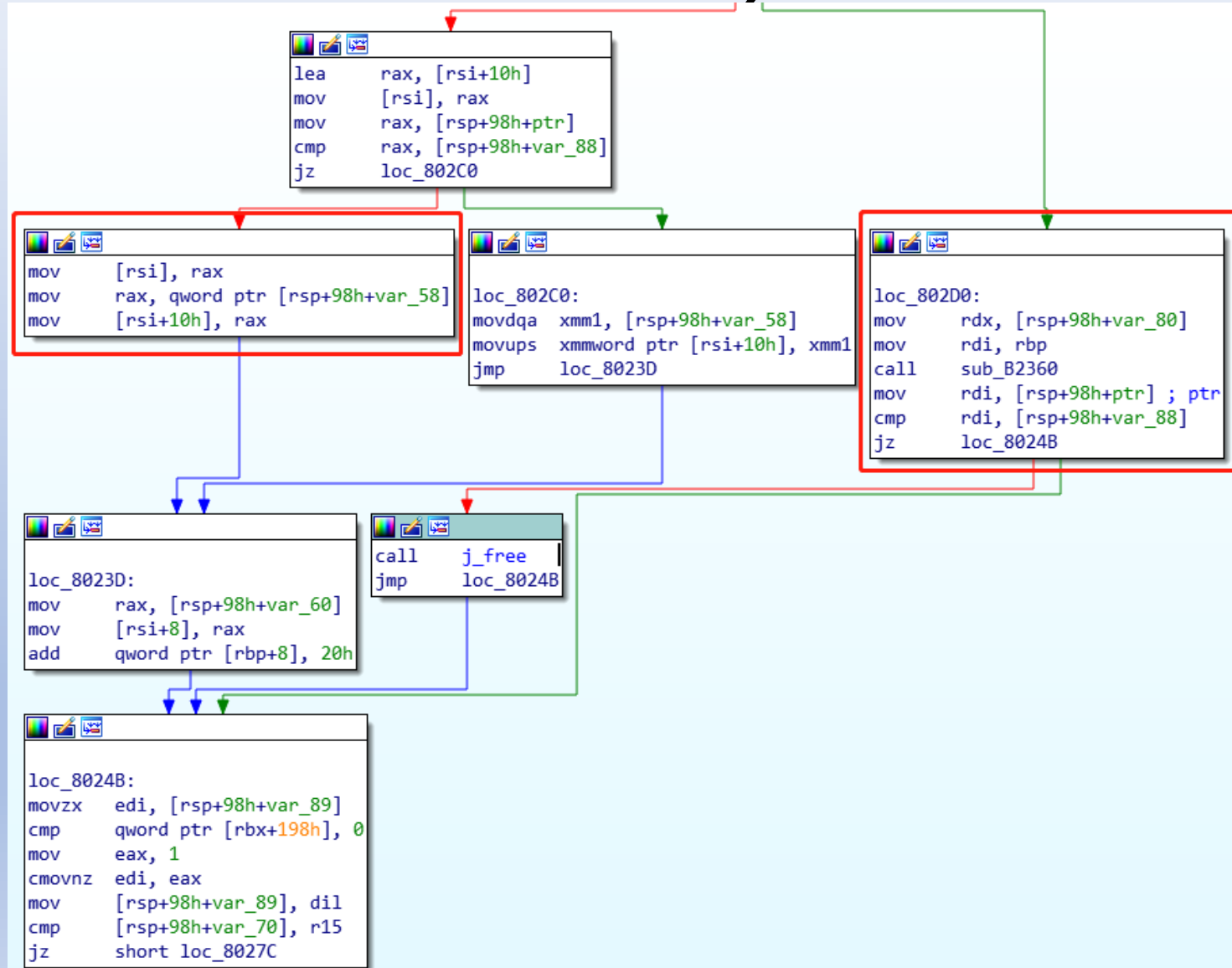
Code Coverage(代码覆盖)

- 函数 (Function-Level)
- 基本块 (BasicBlock-Level)
- 边界 (Edge-Level)

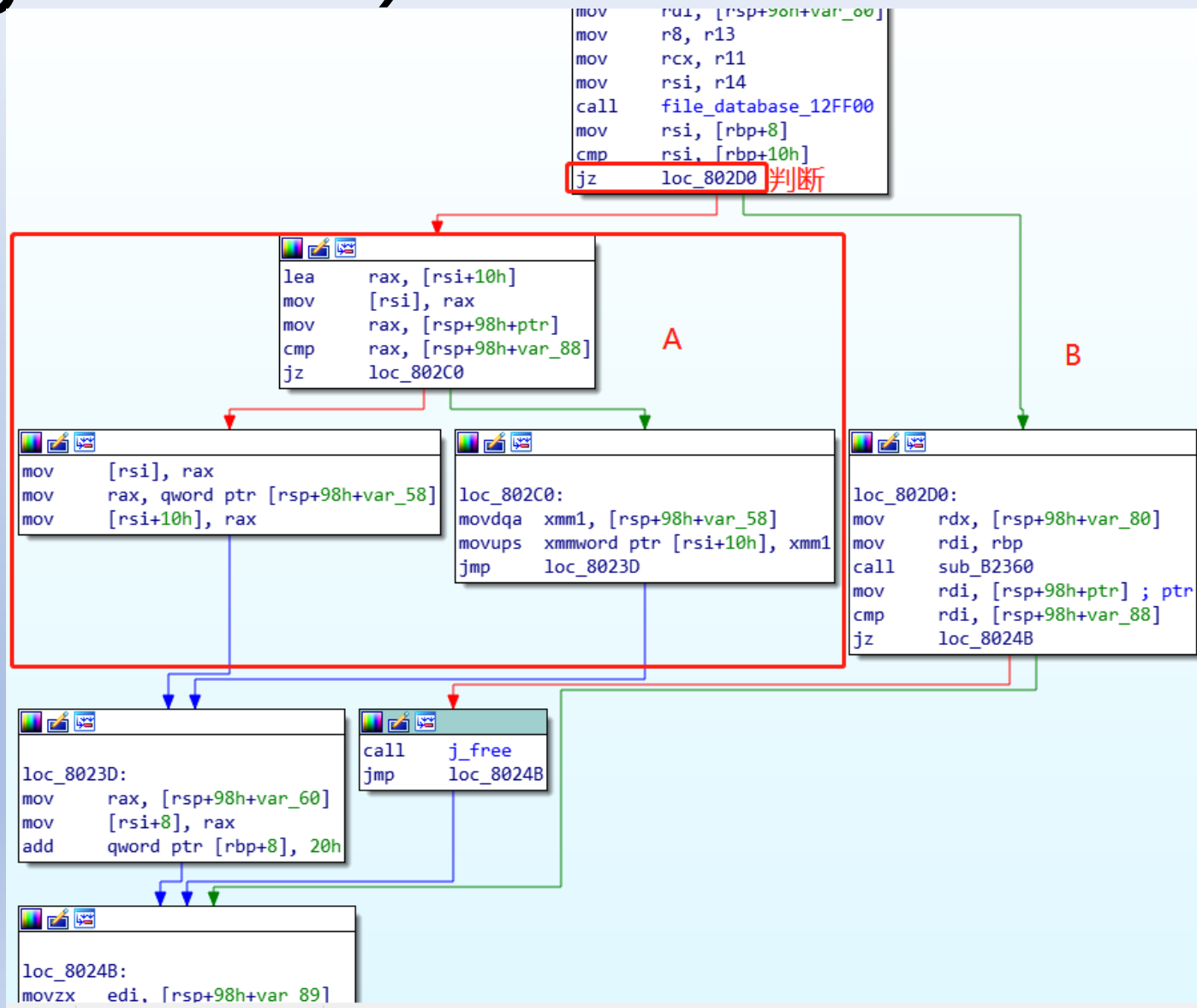
函数 (Function-Level)



基本块 (BasicBlock-Level)



边界 (Edge-Level)



开源模糊测试

- 基于源码手动插桩(理解原理, 具有实验意义)
- 基于源码编译指令插桩(afl/honggfuzz/libfuzzer等等)

基于源码手动插桩

```
// Macros provided for convenience.
#if __has_feature(address_sanitizer) || defined(__SANITIZE_ADDRESS__)
// Marks a memory region as unaddressable.
//
// \note Macro provided for convenience; defined as a no-op if ASan is not
// enabled.
//
// \param addr Start of memory region.
// \param size Size of memory region.
#define ASAN_POISON_MEMORY_REGION(addr, size) \
    __asan_poison_memory_region((addr), (size))

// Marks a memory region as addressable.
//
// \note Macro provided for convenience; defined as a no-op if ASan is not
// enabled.
//
// \param addr Start of memory region.
// \param size Size of memory region.
#define ASAN_UNPOISON_MEMORY_REGION(addr, size) \
    __asan_unpoison_memory_region((addr), (size))
#else
#define ASAN_POISON_MEMORY_REGION(addr, size) \
    ((void)(addr), (void)(size))
#define ASAN_UNPOISON_MEMORY_REGION(addr, size) \
    ((void)(addr), (void)(size))
#endif
```

```
/* Execute "blend" op. Return 0 on success else error code. */
static int do_blend_cube(t2cCtx h, int nBlends) {
    int i;
    __asan_poison_memory_region(h->cube-1, sizeof(struct_t2cCtx)); // 设置redzone
    int nElements = nBlends * h->cube[h->cubeStackDepth].nMasters;
    int iBase = h->stack.cnt - nElements;
    int k = iBase + nBlends;

    if (h->cube[h->cubeStackDepth].nMasters <= 1)
        return t2cErrInvalidWV;
    CHKUFLOW(h, nElements);

    if (h->flags & FLATTEN_CUBE) {
        for (i = 0; i < nBlends; i++) {
            int j;
            double x = INDEX(iBase + i);
            for (j = 1; j < h->cube[h->cubeStackDepth].nMasters; j++)
                x += INDEX(k++) * h->cube[h->cubeStackDepth].WV[j];
            INDEX(iBase + i) = (float)x;
        }
    } else {
        float blendVals[kMaxCubeMasters * kMaxBlendOps];
        for (i = 0; i < nElements; i++) {
            blendVals[i] = INDEX(iBase + i);
        }
        callback_blend_cube(h, nBlends, nElements, blendVals);
    }

    h->stack.cnt = iBase + nBlends;

    __asan_unpoison_memory_region(h->cube-1, sizeof(struct_t2cCtx)); // 解除redzone

    return 0;
}
```

基于源码手动插桩

```
94     int composeOpCnt;
95 -    float composeOpArray[TX_MAX_OP_STACK_CUBE];
96 -    double WV[kMaxCubeMasters]; /* Was originally just
97 - } cube[CUBE_LE_STACKDEPTH];
98     struct /* Stem hints */
99     {
100         long cnt;
101 -    Stem array[T2_MAX_STEMS];
102     } stems;
103     struct /* hint/cntrmask */
104     {
105         short state; /* cntrmask
106         short length; /* Number of
107         short unused; /* Mask unus
108 -    unsigned char bytes[T2_MAX_STEMS / 8]; /* Current
109     } mask;
110     struct /* seac conversion data */
111     {
112         float adx;
113         float ady;
114         int phase;
115     } seac;
116     struct /* Source data */
117     {
118         char *buf; /* Buffer */
119         long length; /* Buffer length */
120         long offset; /* offset in file */
121         long endOffset; /* offset in file fo end of charstr
122         long left; /* Bytes remaining in charstring */
123     } src;
124     short LanguageGroup;
125     t2cAuxData *aux; /* Aux
126     unsigned short gid; /* glyph
127 -    unsigned short regionIndices[CFF2_MAX_MASTERS]; /* vari
128     cff2GlyphCallbacks *cff2; /* CFF2
```

```
94     int composeOpCnt;
95 +    float pre_composeOpArray[8], composeOpArray[TX_MAX_OP_STACK_CUBE], post_composeOpArray[8];
96 +    double pre_WV[8], WV[kMaxCubeMasters], post_WV[8]; /* Was originally just 4, to support subst
97 + } pre_cube, cube[CUBE_LE_STACKDEPTH], post_cube;
98     struct /* Stem hints */
99     {
100         long cnt;
101 +    Stem pre_array[8], array[T2_MAX_STEMS], post_array[8];
102     } stems;
103     struct /* hint/cntrmask */
104     {
105         short state; /* cntrmask state */
106         short length; /* Number of bytes in mask op */
107         short unused; /* Mask unused bits in last byte of mask */
108 +    unsigned char pre_bytes[8], bytes[T2_MAX_STEMS / 8], post_bytes[8]; /* Current mask */
109     } mask;
110     struct /* seac conversion data */
111     {
112         float adx;
113         float ady;
114         int phase;
115     } seac;
116     struct /* Source data */
117     {
118         char *buf; /* Buffer */
119         long length; /* Buffer length */
120         long offset; /* offset in file */
121         long endOffset; /* offset in file fo end of charstring */
122         long left; /* Bytes remaining in charstring */
123     } src;
124     short LanguageGroup;
125     t2cAuxData *aux; /* Auxiliary parse data */
126     unsigned short gid; /* glyph ID */
127 +    unsigned short pre_regionIndices[8], regionIndices[CFF2_MAX_MASTERS], post_regionIndices[8]; /* va
128     cff2GlyphCallbacks *cff2; /* CFF2 font callbacks */
```

基于源码手动插桩

```
static void PoisonArrays(t2cCtx h) {
    int i;

    ASAN_POISON_MEMORY_REGION(&h->stack.pre_array, sizeof(h->stack.pre_array));
    ASAN_POISON_MEMORY_REGION(&h->stack.pre_blendArray, sizeof(h->stack.pre_blendArray));
    ASAN_POISON_MEMORY_REGION(&h->stack.pre_blendArgs, sizeof(h->stack.pre_blendArgs));
    ASAN_POISON_MEMORY_REGION(&h->pre_BCA, sizeof(h->pre_BCA));
    ASAN_POISON_MEMORY_REGION(&h->pre_cube, sizeof(h->pre_cube));
    ASAN_POISON_MEMORY_REGION(&h->stems.pre_array, sizeof(h->stems.pre_array));
    ASAN_POISON_MEMORY_REGION(&h->mask.pre_bytes, sizeof(h->mask.pre_bytes));
    ASAN_POISON_MEMORY_REGION(&h->pre_regionIndices, sizeof(h->pre_regionIndices));

    ASAN_POISON_MEMORY_REGION(&h->stack.post_array, sizeof(h->stack.post_array));
    ASAN_POISON_MEMORY_REGION(&h->stack.post_blendArray, sizeof(h->stack.post_blendArray));
    ASAN_POISON_MEMORY_REGION(&h->stack.post_blendArgs, sizeof(h->stack.post_blendArgs));
    ASAN_POISON_MEMORY_REGION(&h->post_BCA, sizeof(h->post_BCA));
    ASAN_POISON_MEMORY_REGION(&h->post_cube, sizeof(h->post_cube));
    ASAN_POISON_MEMORY_REGION(&h->stems.post_array, sizeof(h->stems.post_array));
    ASAN_POISON_MEMORY_REGION(&h->mask.post_bytes, sizeof(h->mask.post_bytes));
    ASAN_POISON_MEMORY_REGION(&h->post_regionIndices, sizeof(h->post_regionIndices));

    for (i = 0; i < CUBE_LE_STACKDEPTH; i++) {
        ASAN_POISON_MEMORY_REGION(&h->cube[i].pre_composeOpArray, sizeof(h->cube[i].pre_composeOpArray));
        ASAN_POISON_MEMORY_REGION(&h->cube[i].pre_WV, sizeof(h->cube[i].pre_WV));

        ASAN_POISON_MEMORY_REGION(&h->cube[i].post_composeOpArray, sizeof(h->cube[i].post_composeOpArray));
        ASAN_POISON_MEMORY_REGION(&h->cube[i].post_WV, sizeof(h->cube[i].post_WV));
    }
}
```

闭源模糊测试(动态插桩)

- Pin
- Dynamorio(winaf1)
- TinyInst(Jackalope)
- Frida
- ...












模糊测试实战案例

Windows 10 chm文件格式漏洞挖掘

- **微软HTML帮助集**，即**编译的HTML帮助文件**（英语：Microsoft Compiled HTML Help, **CHM**），是**微软**继承早先的**WinHelp**发展的一种**文件格式**，用来提供**在线帮助**，是一种应用较广泛的文件格式。因为CHM文件如一本书一样，可以提供内容目录、索引和搜索等功能，所以也常被用来制作**电子书**。
- 漏洞文章：<https://github.com/xinali/articles/issues/53>

Windows 10 chm文件格式漏洞挖掘

- 二进制exe/dll:
 - hh.exe
 - hhctrl.ocx 相当于hhctrl.dll

 LoadHHA	000000018002A7C0	1
 DllCanUnloadNow	00000001800405F0	2
 AuthorMsg	0000000180037D40	3
 DllGetClassObject	0000000180040610	4
 DllRegisterServer	000000018003FD70	5
 DllUnregisterServer	0000000180040350	6
 doWinMain	0000000180029AE0	13
 HtmlHelpA	00000001800332B0	14
 HtmlHelpW	00000001800330B0	15
 HhWindowThread	0000000180033A60	16
 _DllMainCRTStartup	0000000180080D80	[main entry]

Fuzz HtmlHelpA

C++

```
HWND HtmlHelpA(  
    HWND    hwndCaller,  
    LPCSTR  pszFile,  
    UINT    uCommand,  
    DWORD_PTR dwData  
);
```



```
4 #define _CRT_SECURE_NO_WARNINGS  
5  
6 #include <iostream>  
7 #include <Windows.h>  
8 #include <HtmlHelp.h>  
9  
10 typedef int(*PFN_HtmlHelpA)(HWND hwndCaller, LPCSTR pszFile, UINT uCommand, DWORD_PTR dwData);  
11  
12  
13 extern "C"  
14 __declspec(noinline, dllexport)  
15 int __cdecl fuzz(char* input_file)  
16 {  
17     HMODULE hLibHH = NULL;  
18     PFN_HtmlHelpA pfnHtmlHelpA = NULL;  
19  
20     hLibHH = LoadLibraryA("C:\\Windows\\System32\\hhctrl.ocx");  
21     if (hLibHH == NULL)  
22         return 2;  
23  
24     pfnHtmlHelpA = (PFN_HtmlHelpA)GetProcAddress(hLibHH, "HtmlHelpA");  
25     if (pfnHtmlHelpA != NULL)  
26     {  
27         pfnHtmlHelpA(NULL, input_file, HH_DISPLAY_TOPIC, NULL);  
28         pfnHtmlHelpA(NULL, NULL, HH_CLOSE_ALL, NULL);  
29     }  
30     return 0;  
31 }  
32  
33  
34 int main(int argc, char** argv)  
35 {  
36     if (argc != 2) {  
37         printf("Usage: %s input_file\n", argv[0]);  
38         return -1;  
39     }  
40     fuzz(argv[1]);  
41 }
```

Fuzz HtmlHelpA

- 获取代码覆盖率

- `drrun.exe -t drcov --`
 `.\FuzzWithHtmlHelpA.exe .\chm_corpus\0cb03cb986acfc2fc0140ec8`
 `d41e3515671bf76dfa2fee029b07d6444c957756.chm`

- 精简输入数据集corpus

- `C:\python27\python.exe .\winafl-cmin.py -D`
 `D:\fuzzing\DynamoRIO8\bin64 -t 20000 -i chm_corpus -`
 `o .\chm_minset -covtype edge -coverage_module hhctrl.ocx -`
 `target_module FuzzWithHtmlHelpA.exe -target_method fuzz -nargs`
 `2 -v -- FuzzWithHtmlHelpA.exe @@`

Fuzz HtmlHelpA

IDA View-A

```
; HWND __stdcall HtmlHelpA(HWND hwndCaller, LPCSTR pszFile, UINT uCommand, DWORD  
public HtmlHelpA  
HtmlHelpA proc near  
  
wParam= qword ptr -28h  
var_20= qword ptr -20h  
var_18= dword ptr -18h  
var_10= qword ptr -10h  
arg_0= qword ptr 8  
arg_8= qword ptr 10h  
arg_10= qword ptr 18h  
arg_18= qword ptr 20h  
  
mov     rax, rsp  
mov     [rax+8], rbx  
mov     [rax+10h], rbp  
mov     [rax+18h], rsi  
mov     [rax+20h], rdi  
push   r14  
sub     rsp, 40h  
cmp     cs:?g_fCheckedForCoInitialized@@3HA, 0  
mov     rsi, r9  
mov     edi, r8d  
mov     r14, rdx  
mov     rbp, rcx  
jnz     short loc_18003331C  
  
cmp     cs:?g_fCoInitialized@@3HA, 0  
mov     cs:?g_fCheckedForCoInitialized@@3HA, 1  
jnz     short loc_18003331C
```

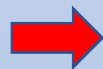
Coverage Overview

Cov %	Func Name	Address	Blocks Hit	Instr. Hit	Func Size	CC
39.59	CSiteMap::ReadFrom...	0x1800578B0	180 / 507	776 / 1960	7628	329
80.45	CHmData::ReadSystem...	0x1800649F0	152 / 221	712 / 885	3699	109
14.10	HelpWndProc (HWND_...	0x180079C60	119 / 760	365 / 2589	10761	459
63.09	FindThisFile (HWND_...	0x18002207C	115 / 216	593 / 940	3917	121
41.22	CSiteMap::ParseSit...	0x180059650	103 / 397	695 / 1686	6217	243
71.45	CreateHelpWindow (c...	0x1800787DC	96 / 163	573 / 802	3458	91
44.48	OnDisplayTopic (HWN...	0x180036EA0	76 / 193	290 / 652	2454	97
57.14	CToc::InitTreeView...	0x180013A58	66 / 139	360 / 630	2744	78
53.90	CTitleInformation:...	0x180063A68	58 / 109	249 / 462	1902	54
79.01	CHHWinType::CloseW...	0x180054E70	53 / 72	207 / 262	1072	44
68.26	CHHWinType::Create...	0x1800554EC	51 / 93	415 / 608	2672	53
60.58	COleDispatchDriver...	0x180077F9C	50 / 100	229 / 378	1367	62
59.60	DllMain	0x18003F47C	48 / 92	239 / 401	1842	52
45.44	CToc::Synchronize (...	0x1800155FC	47 / 117	214 / 471	1931	63
91.41	IsCompiledHtmlFile...	0x180071960	44 / 51	181 / 198	772	30
85.61	CContainer::Create...	0x1800114C0	42 / 60	351 / 410	1614	25
74.38	CContainer::QueryI...	0x180011150	39 / 52	119 / 160	651	38
39.79	SetWinType(char co...	0x180038974	39 / 103	150 / 377	1510	62
27.65	CAutomateContent::...	0x180005818	37 / 127	141 / 510	2111	70
58.33	CAutomateContent::...	0x1800050F0	36 / 57	175 / 300	1299	28
98.54	CContainer::~CCont...	0x180010E70	35 / 36	135 / 137	478	18
84.62	FindWindowType (cha...	0x180056824	34 / 45	110 / 130	474	27
74.38	FindOrCreateWindow...	0x1800569FC	34 / 50	119 / 160	615	25
66.01	CExCollection::~CE...	0x180066608	34 / 56	134 / 203	688	30
76.98	_CRT_INIT	0x180080B3C	33 / 45	107 / 139	562	24
33.10	CHHWinType::Create...	0x18005395C	32 / 111	186 / 562	2462	59
51.30	CExTitle::exOpenFi...	0x18006A8A4	29 / 68	118 / 230	869	33
65.24	CState::_IOpen(voi...	0x18005F4E8	28 / 53	137 / 210	762	27
75.69	CTabControl::CTabC...	0x180065F54	28 / 43	218 / 288	1159	26
80.00	CResourceCache::In...	0x18004ACF0	27 / 40	168 / 210	876	26
100.00	FindHHWindowIndex (...	0x1800799D4	27 / 27	90 / 90	342	19

Graph overview

Fuzz HtmlHelpA

```
C:\WINDOWS\system32\cmd.exe - .\ afl-fuzz.exe -i .\chm_minset -o output -D D:\fuzzin...
WinAFL 1.16b based on AFL 2.43b (FuzzWithHtmlHelpA.exe)
-----+-----+-----+
process timing -----+ overall results -----+
  run time : 0 days, 0 hrs, 43 min, 16 sec | cycles done : 0 |
  last new path : none seen yet | total paths : 45 |
  last uniq crash : none seen yet | uniq crashes : 0 |
  last uniq hang : none seen yet | uniq hangs : 0 |
-----+-----+-----+
cycle progress -----+ map coverage -----+
now processing : 1 (2.22%) | map density : 2.90% / 3.77% |
paths timed out : 0 (0.00%) | count coverage : 1.53 bits/tuple |
-----+-----+-----+
stage progress -----+ findings in depth -----+
now trying : trim 1024\1024 | favored paths : 14 (31.11%) |
stage execs : 28/333 (8.41%) | new edges on : 19 (42.22%) |
total execs : 869 | total crashes : 0 (0 unique) |
exec speed : 0.49/sec (zzzz...) | total tmouts : 0 (0 unique) |
-----+-----+-----+
fuzzing strategy yields -----+ path geometry -----+
bit flips : 0/0, 0/0, 0/0 | levels : 1 |
byte flips : 0/0, 0/0, 0/0 | pending : 45 |
arithmetics : 0/0, 0/0, 0/0 | pend fav : 14 |
known ints : 0/0, 0/0, 0/0 | own finds : 0 |
dictionary : 0/0, 0/0, 0/0 | imported : n/a |
havoc : 0/0, 0/0 | stability : 81.15% |
trim : n/a, n/a |
-----+-----+-----+
[cpu: 0%]
```



```
选择C:\WINDOWS\system32\cmd.exe - .\ afl-fuzz.exe -i .\chm_minset -o output -D D:\fuzzin...
WinAFL 1.16b based on AFL 2.43b (FuzzWithHtmlHelpA.exe)
-----+-----+-----+
process timing -----+ overall results -----+
  run time : 0 days, 15 hrs, 53 min, 27 sec | cycles done : 0 |
  last new path : 0 days, 0 hrs, 0 min, 7 sec | total paths : 49 |
  last uniq crash : none seen yet | uniq crashes : 0 |
  last uniq hang : 0 days, 9 hrs, 34 min, 55 sec | uniq hangs : 3 |
-----+-----+-----+
cycle progress -----+ map coverage -----+
now processing : 1 (2.04%) | map density : 2.90% / 3.81% |
paths timed out : 0 (0.00%) | count coverage : 1.61 bits/tuple |
-----+-----+-----+
stage progress -----+ findings in depth -----+
now trying : calibration | favored paths : 14 (28.57%) |
stage execs : 2/8 (25.00%) | new edges on : 21 (42.86%) |
total execs : 2307 | total crashes : 0 (0 unique) |
exec speed : 0.26/sec (zzzz...) | total tmouts : 7 (3 unique) |
-----+-----+-----+
fuzzing strategy yields -----+ path geometry -----+
bit flips : 0/0, 0/0, 0/0 | levels : 2 |
byte flips : 0/0, 0/0, 0/0 | pending : 49 |
arithmetics : 0/0, 0/0, 0/0 | pend fav : 14 |
known ints : 0/0, 0/0, 0/0 | own finds : 3 |
dictionary : 0/0, 0/0, 0/0 | imported : n/a |
havoc : 0/0, 0/0 | stability : 80.22% |
trim : 0.00%/1319, n/a |
-----+-----+-----+
[cpu: 0%]
```

优化前的思考

- 速度过慢 zzzz



- 有没有更好的办法?



- hhctrl.ocx有没有更合适的导出函数?



- hh.exe还没有使用, 能否直接利用?



- 还有其他方法吗?

使用doWinMain优化

```
IDA - hh.exe C:\Windows\hh.exe
File Edit Jump Search View Debugger Lumina Options Windows Help
Library function Regular function Instruction Data Unexplored External symbol Lumina function
Functions window
Function name
StringCchPrintfA
StringCchPrintfExA
GetRegisteredLocation
WinMain
pre_c_init
pre_cpp_init
WinMainCRTStartup
_mainCRTStartup
_security_check_cookie
_raise_securityfailure
_report_gsfailure
_CxxUnhandledExceptionFilter_EXCEPTION_POINTERS
_CxxSetUnhandledExceptionFilter
_XcptFilter_0
RtlpImageNtHeader
_get_image_app_type
_amsmsg_exit_0
_matherr
_FindPESection
_IsNonwritableInCurrentImage
_ValidateImageBase
_security_init_cookie
_initterm_0
_C_specific_handler_0
Line 18 of 29
Graph overview
35 v19 = xmmword_1400033F8;
36 if ( !(unsigned int)GetRegisteredLocation(&v19, LibFileName) )
37 {
38     v8 = LibFileName;
39     do
40     {
41         if ( v6 == -2147483386 )
42             break;
43         v9 = v8["hhctrl.ocx" - LibFileName];
44         if ( !v9 )
45             break;
46         *v8++ = v9;
47         --v6;
48     }
49     while ( v6 );
50     v10 = v8 - 1;
51     if ( v6 )
52         v10 = v8;
53     *v10 = 0;
54     if ( !v6 )
55         return -1;
56 }
57 v11 = LoadLibraryA(LibFileName);
58 if ( !v11 )
59 {
60     v11 = LoadLibraryA("hhctrl.ocx");
61     if ( !v11 )
62         return -1;
63 }
64 v12 = GetProcAddress(v11, "doWinMain");
65 pDownMain = (__int64)v12;
66 if ( !v12 )
67     return -1;
68 v13 = ((int64 (fastcall *) (HINSTANCE, LPSTR))v12)(hInstance, lpCmdLine);
69 FreeLibrary(v11);
70 return v13;
```

使用doWinMain优化

```
1 __int64 __fastcall doWinMain(HINSTANCE hModule, char *a2)
2 {
3     unsigned int v4; // ebx
4     int v6[6]; // [rsp+20h] [rbp-18h] BYREF
5
6     v6[0] = 0;
7     v4 = -1;
8     if ( InitializeSession((unsigned __int64)v6) )
9     {
10        WinSqmIncrementDWORD(0i64, 2400i64, 1i64);
11        v4 = doInternalWinMain(hModule, a2);
12        if ( g_fCoInitialized )
13        {
14            OleUninitialize();
15            g_fCoInitialized = 0;
16        }
17    }
18    return v4;
19 }
```



```
Command
Response Time (ms) Location
Deferred srv*c:\symbols*https://msdl.microsoft.com/download/symbols
Symbol search path is: srv*c:\symbols*https://msdl.microsoft.com/download/symbols
Executable search path is:
ModLoad: 00007fff`27420000 00007fff`27429000 hh.exe
ModLoad: 00007fff`57f90000 00007fff`58185000 ntdll.dll
ModLoad: 00007fff`562f0000 00007fff`563ad000 C:\WINDOWS\System32\KERNEL32.DLL
ModLoad: 00007fff`55b00000 00007fff`55dc9000 C:\WINDOWS\System32\KERNELBASE.dll
ModLoad: 00007fff`566c0000 00007fff`5676c000 C:\WINDOWS\System32\ADVAPI32.dll
ModLoad: 00007fff`57200000 00007fff`5729e000 C:\WINDOWS\System32\msvcrt.dll
ModLoad: 00007fff`561e0000 00007fff`5627c000 C:\WINDOWS\System32\sechost.dll
ModLoad: 00007fff`575c0000 00007fff`576eb000 C:\WINDOWS\System32\RPCRT4.dll
(31d8.42a8): Break instruction exception - code 80000003 (first chance)
ntdll!LdrpDoDebuggerBreak+0x30:
00007fff`58060570 cc int 3
0:000> sxe ld hhctrl.ocx
0:000> g
ModLoad: 00007fff`24ba0000 00007fff`24c59000 C:\Windows\System32\hhctrl.ocx
ntdll!NtMapViewOfSection+0x14:
00007fff`5802d114 c3 ret
0:000> bu hhctrl!doWinMain
0:000> g
ModLoad: 00007fff`57410000 00007fff`575b0000 C:\WINDOWS\System32\USER32.dll
ModLoad: 00007fff`55830000 00007fff`55852000 C:\WINDOWS\System32\win32u.dll
ModLoad: 00007fff`57f20000 00007fff`57f4a000 C:\WINDOWS\System32\GDI32.dll
ModLoad: 00007fff`556d0000 00007fff`557db000 C:\WINDOWS\System32\gdi32full.dll
ModLoad: 00007fff`55f50000 00007fff`55fed000 C:\WINDOWS\System32\msvc_p_win.dll
ModLoad: 00007fff`55dd0000 00007fff`55ed0000 C:\WINDOWS\System32\ucrtbase.dll
ModLoad: 00007fff`576f0000 00007fff`57e32000 C:\WINDOWS\System32\SHELL32.dll
ModLoad: 00007fff`56c90000 00007fff`56dba000 C:\WINDOWS\System32\ole32.dll
ModLoad: 00007fff`29cc0000 00007fff`29d70000 C:\WINDOWS\WinSxS\amd64_microsoft.windows.common-controls_65
ModLoad: 00007fff`56ea0000 00007fff`571f5000 C:\WINDOWS\System32\combase.dll
ModLoad: 00007fff`56110000 00007fff`561dd000 C:\WINDOWS\System32\OLEAUT32.dll
ModLoad: 00007fff`57300000 00007fff`57355000 C:\WINDOWS\System32\SHLWAPI.dll
ModLoad: 00007fff`57ef0000 00007fff`57f20000 C:\WINDOWS\System32\IMM32.DLL
Breakpoint 0 hit
hhctrl!doWinMain:
00007fff`24bc9ae0 488bc4 mov rax,rsq
0:000> dc rdx
000001a7`22913b92 665c3a44 697a7a75 465c676e 547a7a75 D:\fuzzing\FuzzT
000001a7`22913ba2 5c747365 5f6d6863 70726f63 305c7375 est\chm_corpus\0
000001a7`22913bb2 34316462 62386362 34663830 32636261 bd14bc8b08f4abc2
000001a7`22913bc2 61613136 65363934 35363231 62316430 61aa496e12650d1b
000001a7`22913bd2 38643661 65353531 38623838 64346533 a6d8155e88b83e4d
000001a7`22913be2 66336561 63663864 36386164 2e333038 ae3fd8fcd8a86803.
000001a7`22913bf2 006d6863 abababab abababab abababab chm.....
000001a7`22913c02 abababab feefeeee feefeeee 0000feee .....
```


使用doWinMain优化

```
typedef int(*PFN_DoWinMain)(HMODULE, const char*);

extern "C"
__declspec(noinline, dllexport)
// int __cdecl fuzz(char* input_file, PFN_HtmlHelpA pfnHtmlHelpA)
int __cdecl fuzz(char* input_file)
{
    HMODULE hLibHH = NULL;
    PFN_DoWinMain pfnDoWinMain = NULL;

    hLibHH = LoadLibraryA("C:\\Windows\\system32\\hhctrl1.ocx");
    if (hLibHH == NULL)
        return 2;

    pfnDoWinMain = (PFN_DoWinMain)GetProcAddress(hLibHH, "doWinMain");
    std::string chCommandLine = "-decompile test_chm ";
    chCommandLine += std::string(input_file);

    if (pfnDoWinMain != NULL)
        pfnDoWinMain(hLibHH, chCommandLine.c_str());
}

int main(int argc, char** argv)
{
    if (argc != 2) {
        printf("Usage: %s input_file\n", argv[0]);
        return 1;
    }
    fuzz(argv[1]);
    return 0;
}
```



```
C:\WINDOWS\system32\cmd.exe - .\afl-fuzz.exe -i .\chm_minset_winmain -o output_winmain -D D:\fuzzing\DynamoRIO8\bin64...
known ints : 0/0, 0/0, 0/0      own finds : 15
dictionary : 0/0, 0/0, 0/0     imported  : n/a
havoc      : 0/0, 0/0          stability  : 77.13%
trim       : 65.77%/1566, n/a
-----+-----+-----+
[cpu: 0%]
WinAFL 1.16b based on AFL 2.43b (FuzzWithDoWinMain)
+-----+-----+-----+
| process timing | overall results |
+-----+-----+-----+
| run time      : 0 days, 0 hrs, 12 min, 56 sec | cycles done : 0 |
| last new path : 0 days, 0 hrs, 1 min, 20 sec | total paths : 21 |
| last uniq crash : 0 days, 0 hrs, 5 min, 0 sec | uniq crashes : 4 |
| last uniq hang  : 0 days, 0 hrs, 4 min, 17 sec | uniq hangs  : 1 |
+-----+-----+-----+
| cycle progress | map coverage |
+-----+-----+-----+
| now processing : 1 (4.76%) | map density : 0.52% / 0.86% |
| paths timed out : 0 (0.00%) | count coverage : 2.40 bits/tuple |
+-----+-----+-----+
| stage progress | findings in depth |
+-----+-----+-----+
| now trying : bitflip 1\1 | favored paths : 3 (14.29%) |
| stage execs : 3501/199k (1.76%) | new edges on : 7 (33.33%) |
| total execs : 5364 | total crashes : 28 (4 unique) |
| exec speed : 6.86/sec (zzzz...) | total tmouts : 1 (1 unique) |
+-----+-----+-----+
| fuzzing strategy yields | path geometry |
+-----+-----+-----+
| bit flips : 0/0, 0/0, 0/0 | levels : 2 |
| byte flips : 0/0, 0/0, 0/0 | pending : 21 |
| arithmetics : 0/0, 0/0, 0/0 | pend fav : 3 |
| known ints : 0/0, 0/0, 0/0 | own finds : 15 |
| dictionary : 0/0, 0/0, 0/0 | imported  : n/a |
| havoc      : 0/0, 0/0 | stability  : 77.13% |
| trim       : 65.77%/1566, n/a |
+-----+-----+-----+
[cpu: 0%]
```

进一步优化

```
static void
event_module_load(void *drcontext, const module_data_t *info, bool loaded)
{
    const char *module_name = info->names.exe_name;
    app_pc to_wrap = 0;

    if (module_name == NULL) {
        // In case exe_name is not defined, we will fall back on the preferred name.
        module_name = dr_module_preferred_name(info);
    }

    if (options.debug_mode)
        dr_fprintf(winafl_data.log, "Module loaded, %s\n", module_name);

    if (options.fuzz_module[0]) {
        if (_stricmp(module_name, options.fuzz_module) == 0) {
            if (options.fuzz_offset) {
                to_wrap = info->start + options.fuzz_offset;
            } else {
                //first try exported symbols
                to_wrap = (app_pc)dr_get_proc_address(info->handle, options.fuzz_method);
                if (!to_wrap) {
                    //if that fails, try with the symbol access library
                    #ifdef USE_DRSYMS
                        drsym_init(0);
                        drsym_lookup_symbol(info->full_path, options.fuzz_method, (size_t *)&to_wrap, 0);
                        drsym_exit();
                    #endif
                    DR_ASSERT_MSG(to_wrap, "Can't find specified method in fuzz_module");
                    to_wrap += (size_t)info->start;
                }
            }
        }
        if (options.persistence_mode == native_mode)
        {
            drwrap_wrap_ex(to_wrap, pre_fuzz_handler, post_fuzz_handler, NULL, options.callconv);
        }
        if (options.persistence_mode == in_app)
        {
            drwrap_wrap_ex(to_wrap, pre_loop_start_handler, NULL, NULL, options.callconv);
        }
    }
}
```

99 On Windows, the `<tt>_NT_SYMBOL_PATH</tt>` environment variable is honored by
100 `\p drsyms` as a local cache of `\p pdb files`. However, `\p drsyms` does not
101 support symbol store paths (those that contain `\p srv`) when used inside of
102 a DynamoRIO client. Such paths should work fine when used in standalone
103 applications, provided that both `\p symsrv.dll` and `\p dbghelp.dll` are
104 locatable by the Windows loader.

106 `\section sec_drsyms_exports Exported Functions`

108 For clients interested only in locating specific functions exported from a
109 library, it is not necessary to use `\p drsyms` as the core DynamoRIO API
110 provides functions for iterating modules and looking up module exports.
111 The following core DynamoRIO API functions are relevant:

- 112 - `dr_get_proc_address()`
- 113 - `dr_get_application_name()`
- 114 - `dr_register_module_load_event()`
- 115 - `dr_lookup_module()`
- 116 - `dr_lookup_module_by_name()`
- 117 - `dr_module_iterator_start()`
- 118

进一步优化

- `afl-fuzz.exe -i .\chm minset winmain -o output_winmain -D D:\fuzzing\DynamoRIO8\bin64 -t 20000 -- -coverage_module hhctrl.ocx -target_module hhctrl.ocx -target_method doWinMain -nargs 2 -- hh.exe -decompile D:\fuzzing\FuzzTest\test_chm @@`

```
C:\WINDOWS\system32\cmd.exe - afl-fuzz.exe -i .\chm_minset_winmain -o output_winmain -D D:\fuzzin... [cpu: 0%]
-----+
WinAFL 1.16b based on AFL 2.43b (hh.exe)
-----+
+-- process timing -----+-- overall results -----+
|   run time : 0 days, 0 hrs, 10 min, 3 sec   |   cycles done : 0   |
|   last new path : 0 days, 0 hrs, 0 min, 41 sec |   total paths : 15 |
|   last uniq crash : 0 days, 0 hrs, 2 min, 26 sec |   uniq crashes : 3 |
|   last uniq hang : 0 days, 0 hrs, 1 min, 44 sec |   uniq hangs : 1   |
+-----+-----+-----+
+-- cycle progress -----+-- map coverage -----+
| now processing : 1 (6.67%) |   map density : 0.52% / 0.67% |
| paths timed out : 0 (0.00%) |   count coverage : 2.69 bits/tuple |
+-----+-----+-----+
+-- stage progress -----+-- findings in depth -----+
| now trying : bitflip 1\1 |   favored paths : 3 (20.00%) |
| stage execs : 2435/199k (1.22%) |   new edges on : 5 (33.33%) |
| total execs : 4186 |   total crashes : 29 (3 unique) |
| exec speed : 7.05/sec (zzzz...) |   total tmouts : 1 (1 unique) |
+-----+-----+-----+
+-- fuzzing strategy yields -----+-- path geometry -----+
| bit flips : 0/0, 0/0, 0/0 |   levels : 2 |
| byte flips : 0/0, 0/0, 0/0 |   pending : 15 |
| arithmetics : 0/0, 0/0, 0/0 |   pend fav : 3 |
| known ints : 0/0, 0/0, 0/0 |   own finds : 9 |
| dictionary : 0/0, 0/0, 0/0 |   imported : n/a |
|   havoc : 0/0, 0/0 |   stability : 93.85% |
|   trim : 65.77%/1566, n/a |   |
+-----+-----+-----+
[cpu: 0%]
```

再进一步优化?

```
IDA View-A Pseudocode-A
1 Hwnd __stdcall HtmlHelpA(Hwnd hwndCaller, LPCSTR pszFile, UINT uCommand, DWORD_PTR dwData)
2 {
3     __int64 v8; // rax
4     HCURSOR v10; // rax
5     unsigned int v11; // edx
6     HCURSOR v12; // rbx
7     int v13; // eax
8     Hwnd v14; // rcx
9     int v15; // eax
10    Hwnd v16; // rax
11    struct CProcessError *v17; // rax
12    Hwnd v18; // rdi
13    WPARAM wParam[2]; // [rsp+20h] [rbp-28h] BYREF
14    UINT v20; // [rsp+30h] [rbp-18h]
15    DWORD_PTR v21; // [rsp+38h] [rbp-10h]
16
17    if ( !g_fCheckedForCoInitialized )
18    {
19        g_fCheckedForCoInitialized = 1;
20        if ( !g_fCoInitialized )
21        {
22            if ( OleInitialize(0i64) == 1 )
23                OleUninitialize();
24            else
25                g_fCoInitialized = 1;
26        }
27    }
28    if ( uCommand == 253 || uCommand == 256 )
29        return xHtmlHelpA(hwndCaller, pszFile, uCommand, dwData);
30    if ( uCommand != 30 )
31    {
32        v10 = LoadCursorA(0i64, (LPCSTR)0x7F02);
33        v12 = SetCursor(v10);
34        if ( !g_fStandAlone )
35        {
```

结果处理

```
uniq_crash_type_2020-01-17_domain.txt x
D: > Dropbox > fuzzing > scripts > uniq_crash_type_2020-01-17_domain.txt
1  itss!DllGetClassObject+0x195a7 -> itss!DllGetClassObject+0xc948:
2  [+].\TestHH\output\test_hh\crashes\id_000000_00
3  [+].\TestHH\output\test_hh\crashes\id_000002_00
4  [+].\TestHH\output\test_hh2\crashes_20200117143458\id_000000_00
5  itss!DllGetClassObject+0x195a7 -> itss!DllGetClassObject+0x4fe2:
6  [+].\TestHH\output\test_hh\crashes\id_000001_00
7  [+].\TestHH\output\test_hh\crashes\id_000003_00
8  [+].\TestHH\output\test_hh\crashes\id_000009_00
9  [+].\TestHH\output\test_hh2\crashes_20200117143458\id_000001_00
10 [+].\TestHH\output\test_hh2\crashes_20200117143458\id_000002_00
11 [+].\TestHH\output\test_hh2\crashes_20200117143458\id_000003_00
12 [+].\TestHH\output\test_hh2\crashes_20200117143458\id_000004_00
13 [+].\TestHH\output\test_hh2\crashes_20200117143458\id_000008_00
14 [+].\TestHH\output\test_hh2\crashes_20200117143458\id_000009_00
15 itss!DllGetClassObject+0x18508 -> itss!DllGetClassObject+0x80e4:
16 [+].\TestHH\output\test_hh\crashes\id_000004_00
17 [+].\TestHH\output\test_hh\crashes\id_000005_00
18 [+].\TestHH\output\test_hh\crashes\id_000006_00
19 [+].\TestHH\output\test_hh\crashes\id_000008_00
20 [+].\TestHH\output\test_hh\crashes\id_000011_00
21 [+].\TestHH\output\test_hh\crashes\id_000013_00
22 [+].\TestHH\output\test_hh\crashes\id_000015_00
23 [+].\TestHH\output\test_hh\crashes\id_000016_00
24 [+].\TestHH\output\test_hh2\crashes_20200117143458\id_000006_00
25 [+].\TestHH\output\test_hh2\crashes_20200117143458\id_000012_00
26 [+].\TestHH\output\test_hh2\crashes_20200117143458\id_000014_00
27 [+].\TestHH\output\test_hh2\crashes_20200117143458\id_000016_00
28 [+].\TestHH\output\test_hh2\crashes_20200117143458\id_000017_00
29 [+].\TestHH\output\test_hh2\crashes_20200117143458\id_000018_00
30 [+].\TestHH\output\test_hh2\crashes_20200117143458\id_000019_00
31 > itss!DllGetClassObject+0x108df -> itss!DllGetClassObject+0x6de5: ...
40 ntdll!RtlpBreakPointHeap+0x16 -> ntdll!RtlpCheckBusyBlockTail+0x20d:
41 [+].\TestHH\output\test_hh\crashes\id_000012_00
42 [+].\TestHH\output\test_hh\hangs\id_000001
43 > ntdll!NtTerminateProcess+0x14 -> ntdll!RtlExitUserProcess+0xb8: ...
52 itss!DllGetClassObject+0x3706 -> itss!DllGetClassObject+0x2d72:
53 [+].\TestHH\output\test_hh2\crashes_20200117143458\id_000013_00
54 [+].\TestHH\output\test_hh2\hangs_20200117143458\id_000001
55
```

```
ModLoad: 00007fff`90720000 00007fff`907c2000 C:\WINDOWS\System32\clbcatq.dll
ModLoad: 00007fff`71a90000 00007fff`71abe000 C:\Windows\System32\itss.dll
ModLoad: 00007fff`844c0000 00007fff`84696000 C:\Windows\System32\urlmon.dll
ModLoad: 00007fff`7d830000 00007fff`7dd06000 C:\Windows\System32\WININET.dll
ModLoad: 00007fff`84210000 00007fff`844b6000 C:\Windows\System32\iertutil.dll
ModLoad: 00007fff`8df50000 00007fff`8df5c000 C:\Windows\System32\CRYPTBASE.DLL
(1bdc.34b0): Access violation - code c0000005 (first chance)
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
itss!DllGetClassObject+0x3706:
00007fff`71a9b346 488b4020 mov rax,qword ptr [rax+20h] ds:6e65746e`6f432f84=????????????????
0:000> k
# Child-SP RetAddr Call Site
00 000000ea`e0aff450 00007fff`71a9a9b2 itss!DllGetClassObject+0x3706
01 000000ea`e0aff480 00007fff`71a9b214 itss!DllGetClassObject+0x2d72
02 000000ea`e0aff4b0 00007fff`71a9b10e itss!DllGetClassObject+0x35d4
03 000000ea`e0aff4f0 00007fff`6de74963 itss!DllGetClassObject+0x34ce
04 000000ea`e0aff530 00007fff`6de75403 hhctrl!Ordinal10+0x24963
05 000000ea`e0aff7c0 00007fff`6de77811 hhctrl!Ordinal10+0x25403
06 000000ea`e0aff800 00007fff`6de774de hhctrl!doWinMain+0x391
07 000000ea`e0affbb0 00007fff`29521226 hhctrl!doWinMain+0x5e
08 000000ea`e0affbe0 00007fff`29521868 TestDoWinMain+0x1226
09 000000ea`e0affc80 00007fff`8fde7bd4 TestDoWinMain+0x1868
0a 000000ea`e0affcc0 00007fff`9170ced1 KERNEL32!BaseThreadInitThunk+0x14
0b 000000ea`e0affcf0 00000000`00000000 ntdll!RtlUserThreadStart+0x21
0:000> !load msec_x64.dll
0:000> !exploitable
!exploitable 1.6.0.0
Exploitability Classification: UNKNOWN
Recommended Bug Title: Read Access Violation starting at itss!DllGetClassObject+0x0000000000003706 (Hash=0xb07b0e28.)
```

结论

Hello Simon,

Thank you for contacting the Microsoft Security Response Center (MSRC). We appreciate the time taken to submit this assessment.

Upon investigation, we have determined that this submission does not meet the bar for security servicing.

CHM files are essentially equivalent to EXE files. If an attacker can coerce a user into running a CHM (clicking through the warning), then the attacker can run arbitrary code on the user's system. This is why it does not matter from a security perspective if there are vulnerabilities in CHM parsing itself, because the CHM format already gives an attacker the ability to run arbitrary code.

Here is a link from MITRE for more information: <https://attack.mitre.org/techniques/T1223/>

As such, this email thread has been closed and will no longer be monitored.

If you believe this determination to be in error, submit a new report at .

Please include:

- Relevant information previously provided in your initial report
- Detailed steps required to consistently reproduce the issue
- Short explanation on how an attacker could use the information to exploit another user remotely
- Proof-of-concept (POC), such as a video recording, crash reports, screenshots, or relevant code samples

Regards,
MSRC

附录:

- chm漏洞文章:
<https://github.com/xinali/articles/issues/53>
- winaf1源码:
<https://github.com/googleprojectzero/winaf1>
- 代码覆盖率查看工具:
<https://github.com/gaasedelen/lighthouse>
- Dynamorio源码:
<https://github.com/DynamoRIO/dynamorio>

谢谢

Any Questions?