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

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

Up to $1,000,000									1.001 Win RCE Zero Click (Win)
Up to $500,000						3.001 Chrome RCE+LPE (Win)	2.001 Apache RCE (Linux)	2.002 MS IIS RCE (Win)	
Up to $250,000					5.001 MS Outlook RCE (Win)	4.001 MS Exchange RCE (Win)	2.003 OpenSSL RCE (Linux)	2.004 PHP RCE (Linux)	
Up to $200,000	6.001 VMware ESXi VME	5.002 Thunderbird RCE (Win/Linux)		4.002 Sendmail RCE (Linux)	4.003 Postfix RCE (Linux)	4.004 Dovecot RCE (Linux)	4.005 Exim RCE (Linux)	2.005 nginx RCE (Linux)	
Up to $100,000		3.002 Safari RCE+LPE (Mac)	3.003 Edge RCE+LPE (Win)	3.004 Firefox RCE+LPE (Win)	5.003 Word/Excel RCE (Win)	7.001 WordPress RCE (Linux)	7.002 cPanel/WHM RCE (Linux)	7.003 Plesk RCE (Linux)	7.004 Webmin RCE (Linux)
Up to $80,000	6.002 VMware WS VME (Win/Linux)				5.004 Adobe PDF RCE+SBX (Win)	5.005 WinRAR RCE (Win)	5.006 7-Zip RCE (Win)	6.003 Windows LPE/SBX (Win)	
Up to $50,000	6.004 USB LPE (Win/Mac)	8.001 Antivirus RCE (Win)		5.007 WinZip RCE (Win)	5.008 tar RCE (Linux)	6.005 macOS LPE/SBX (Mac)	6.006 Linux LPE (Linux)	6.007 BSD LPE (BSD)	
Up to $10,000	9.001 Routers RCE	8.002 Antivirus LPE (Win)	7.005 phpBB RCE (Linux)	7.006 vBulletin RCE (Linux)	7.007 MyBB RCE (Linux)	7.008 Joomla RCE (Linux)	7.009 Drupal RCE (Linux)	7.010 Roundcube RCE (Linux)	7.011 Horde RCE (Linux)

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

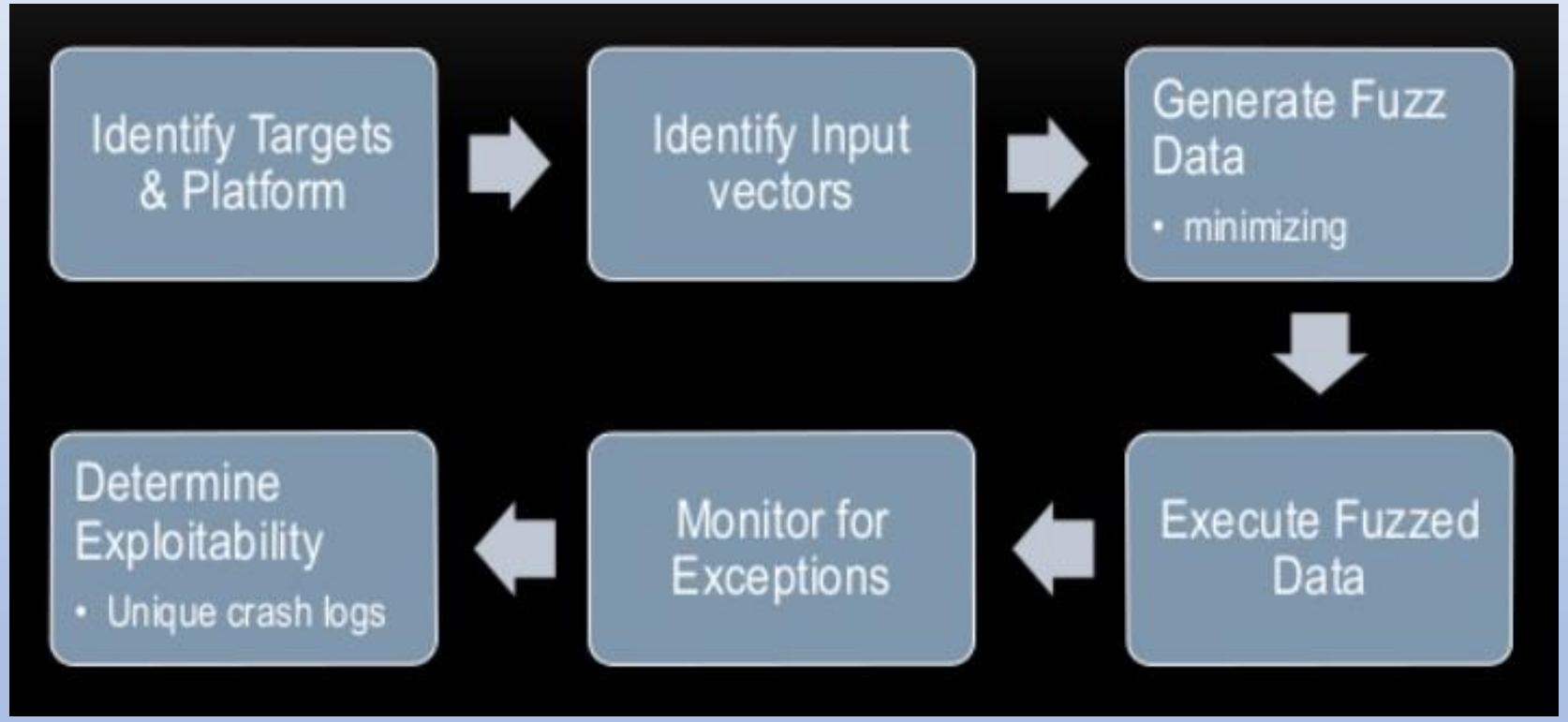
- 模糊测试 （fuzz testing, fuzzing）是一种软件测试技术。其核心思想是将自动或半自动生成的随机数据输入到一个程序中，并监视程序异常，如崩溃，断言（assertion）失败，以发现可能的程序错误，比如内存泄漏。

- 主要测试方式：变异测试（mutation-based）以及生成测试（generation-based）

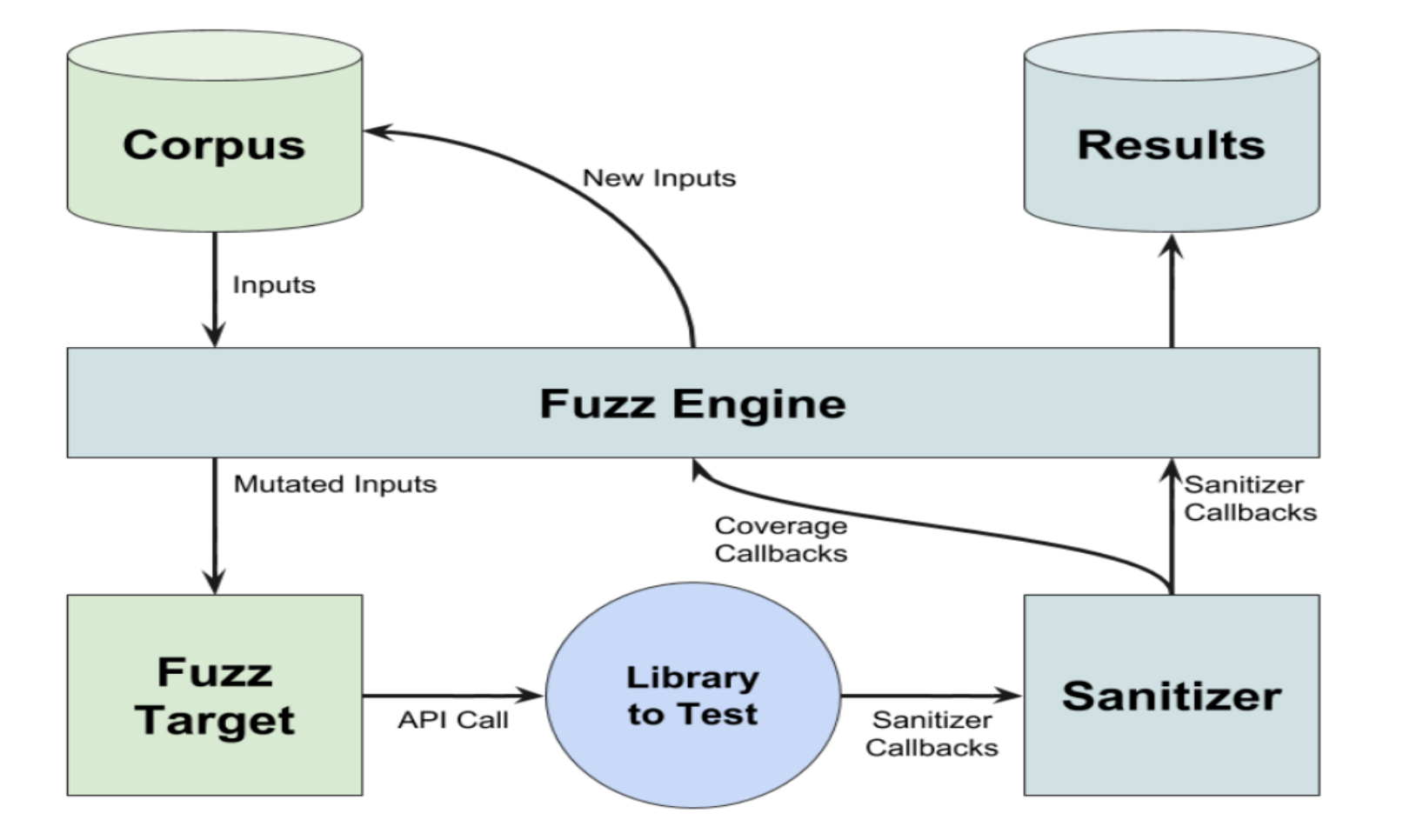
- 主要测试目标：文件格式与网络协议等

- 简单分类：开源/闭源模糊测试

生成测试


Identify Targets & Platform
Identify Input vectors
Generate Fuzz Data
• minimizing
Execute Fuzzed Data
Monitor for Exceptions
Determine Exploitability
• Unique crash logs

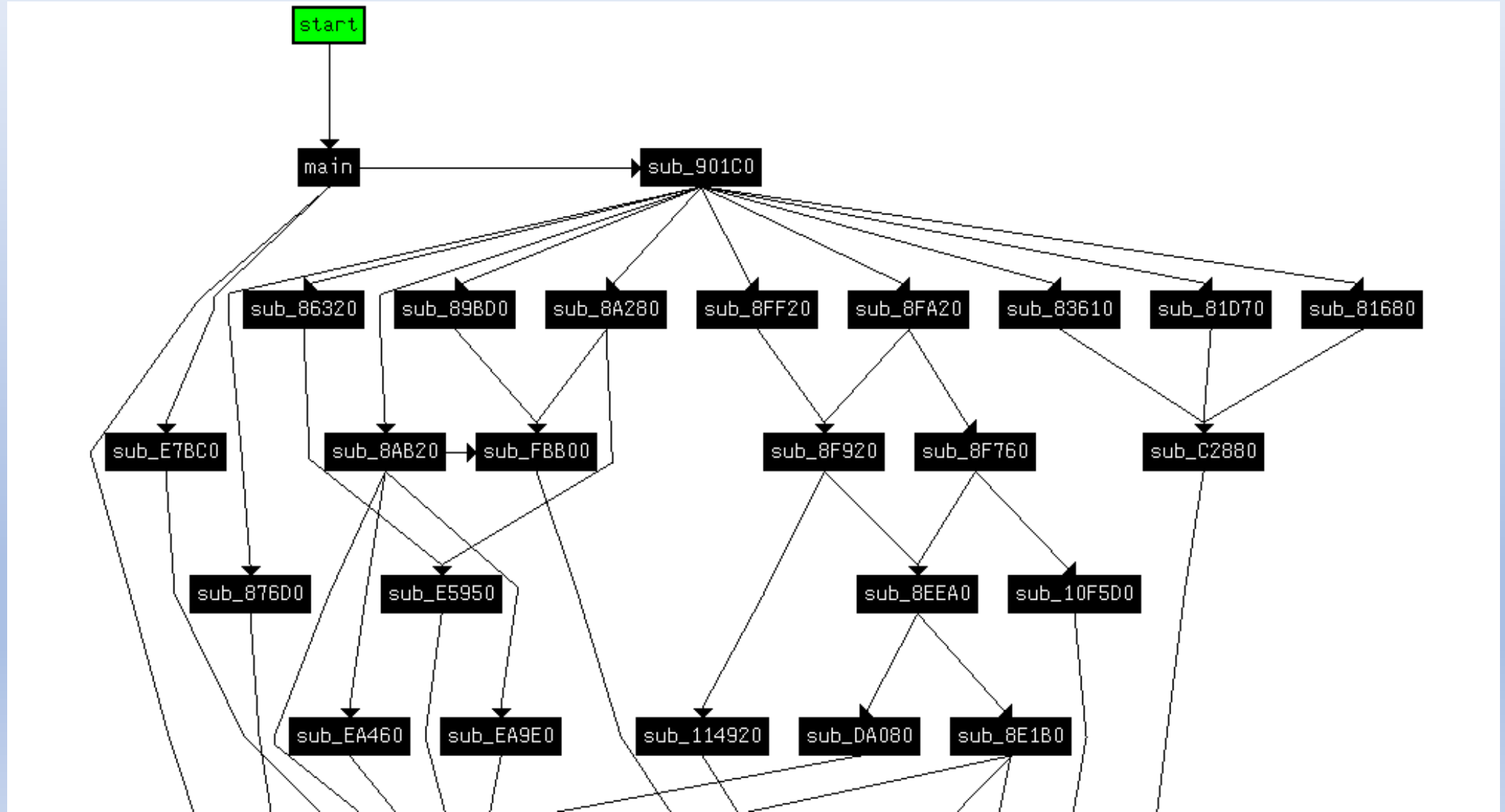
变异测试


Corpus
New Inputs
Results
Inputs
Fuzz Engine
Mutated Inputs
Coverage
Callbacks
Sanitizer
Callbacks
Fuzz
Target
API Call
Library
to Test
Sanitizer
Callbacks
Sanitizer

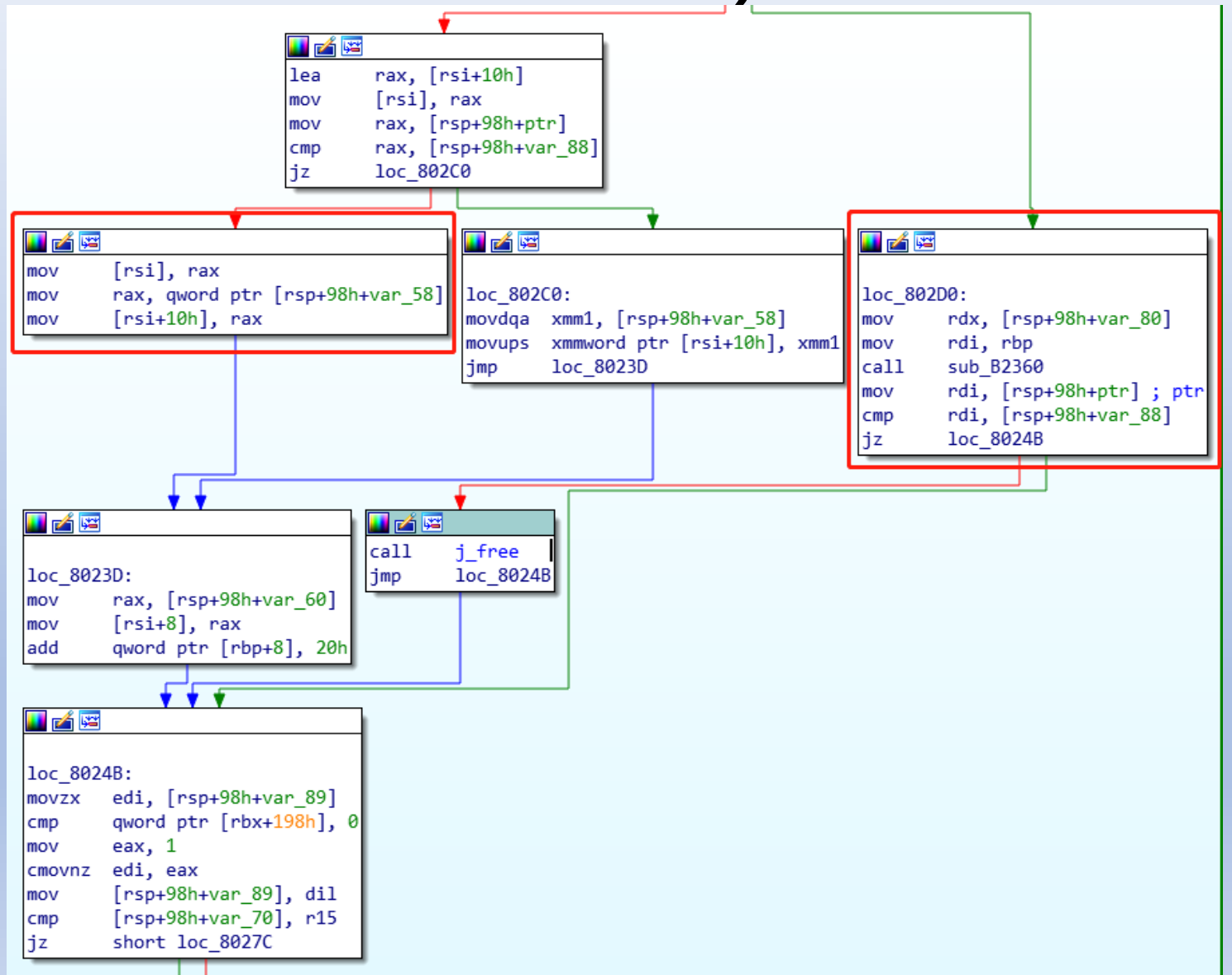
Code Coverage(代码覆盖)

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

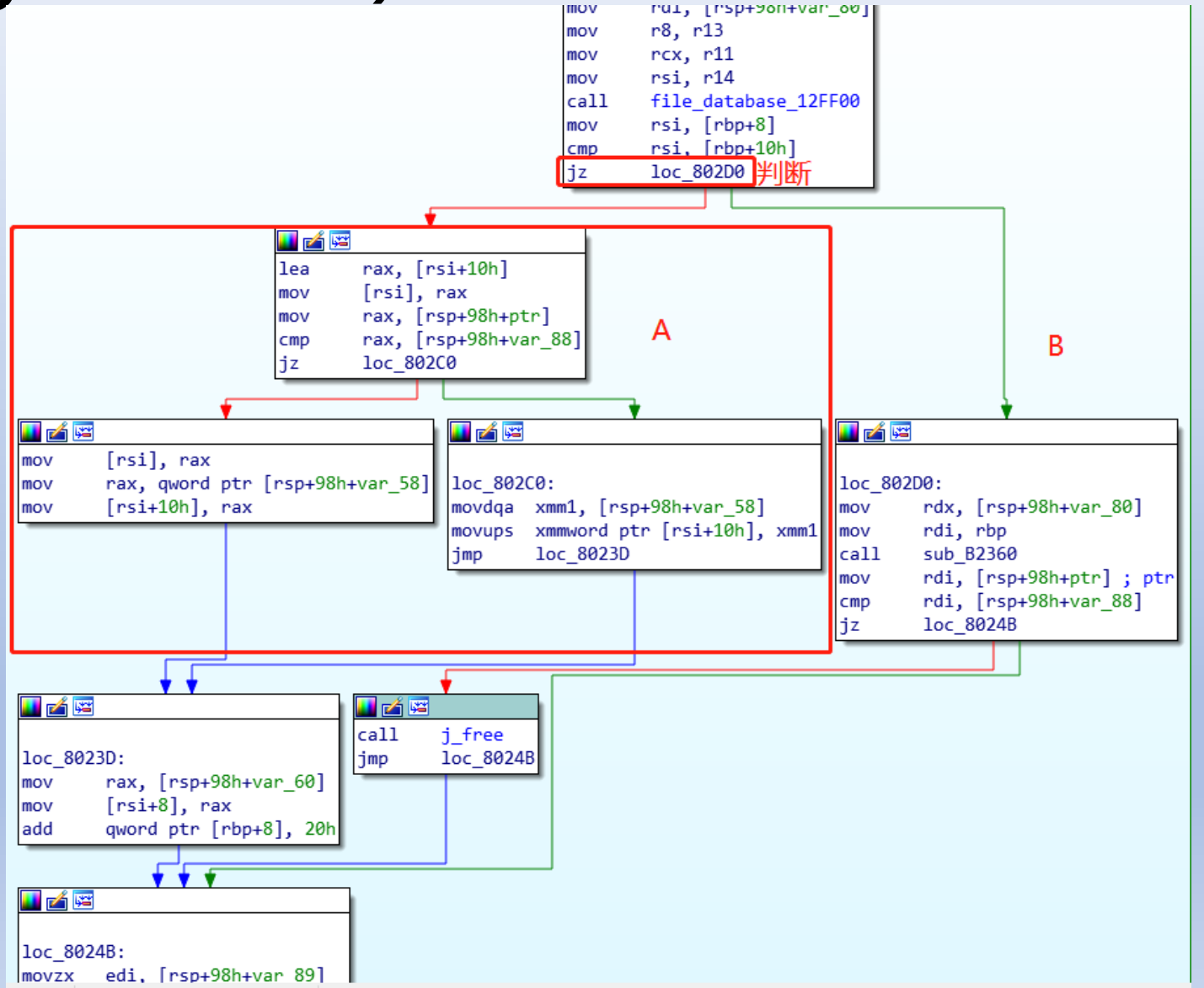
函数（Fuction-Level）



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)

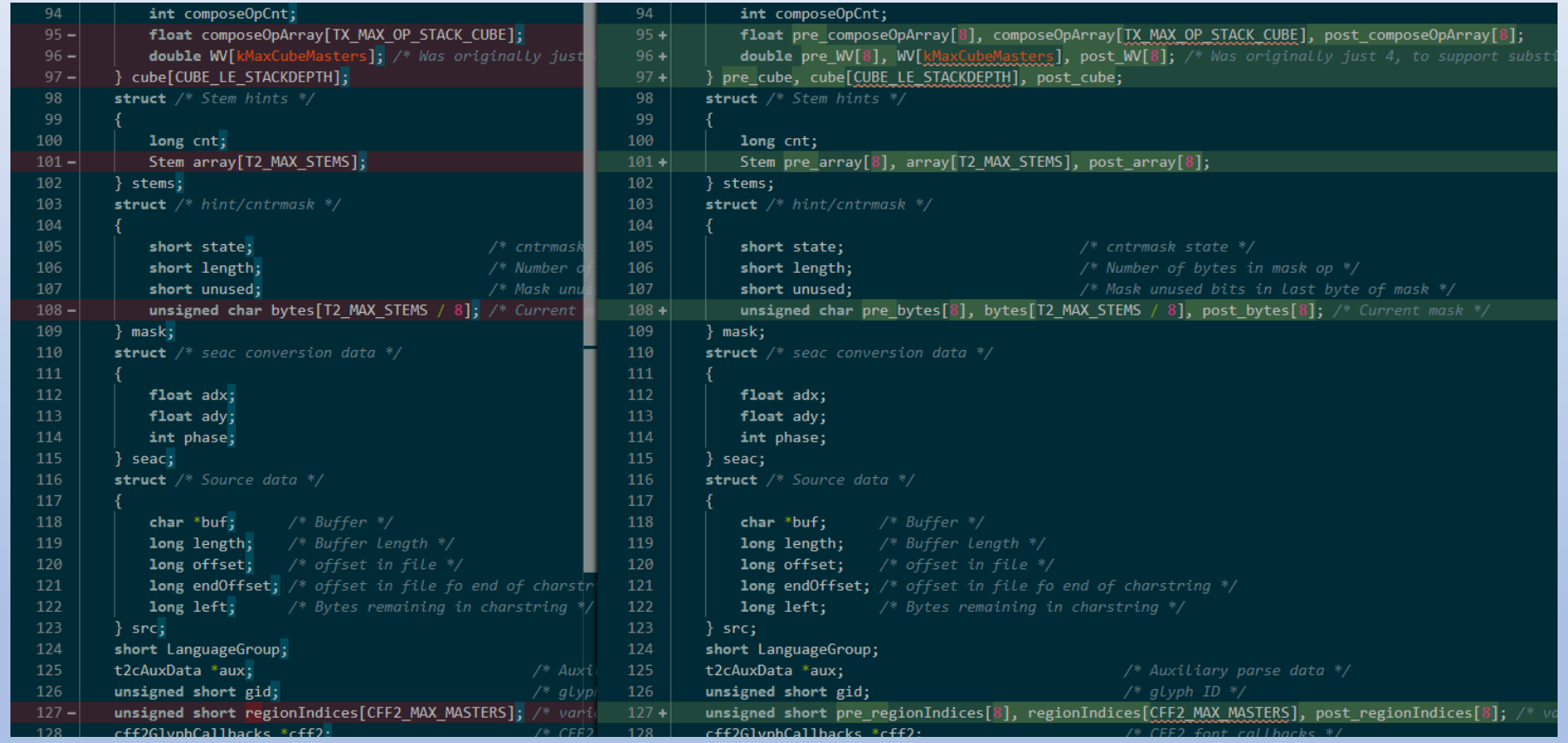
- 基于源码编译指令插桩(afl/honggfuzz/libfuzzer等等)

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 o
107       short unused;                         /* Mask unu
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;                              /* Auxi
126   unsigned short gid;                           /* glyp
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 substi
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(winafl)
- TinyInst(Jackalope)
- Frida
- ...

模糊测试实战案例

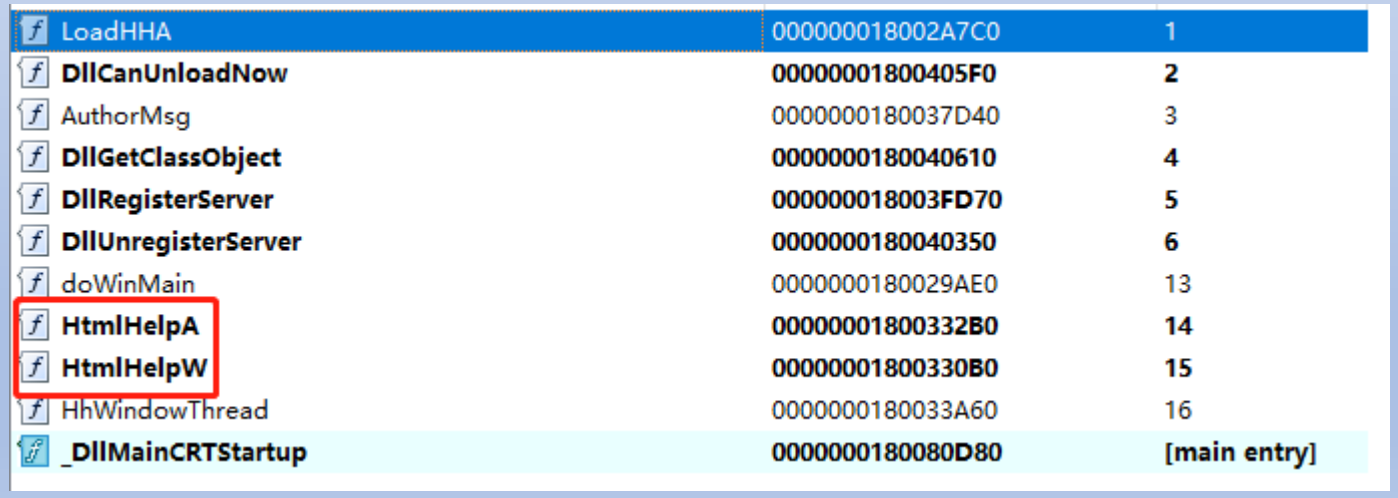
Windows 10 chm文件格式漏洞挖掘

- **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	0000000180002A7C0	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
);
```

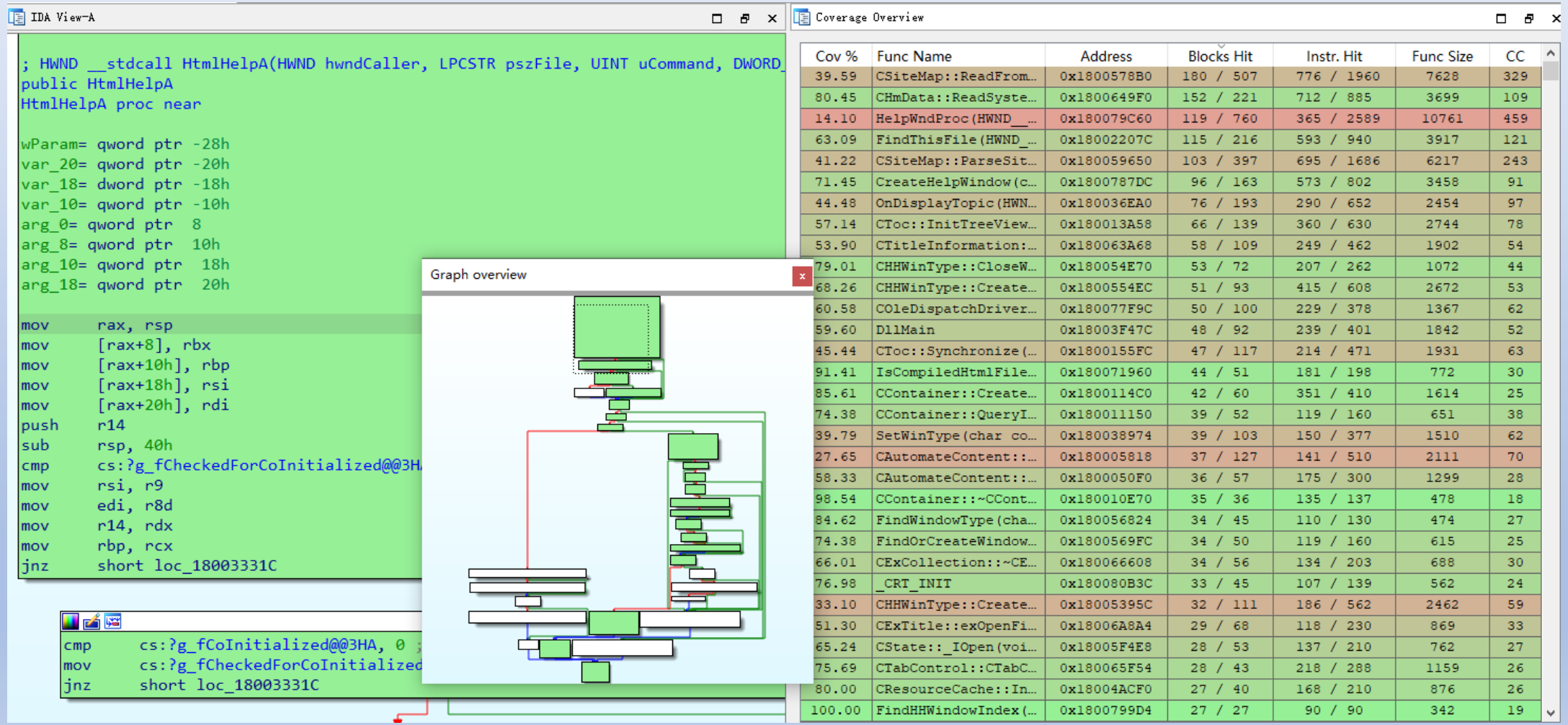
```C++
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <Windows.h>
#include <HtmlHelp.h>

typedef int(*PFN_HtmlHelpA)(HWND hwndCaller, LPCSTR pszFile, UINT uCommand, DWORD_PTR dwData);


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

    hLibHH = LoadLibraryA("C:\\Windows\\System32\\hhctrl.ocx");
    if (hLibHH == NULL)
        return 2;

    pfnHtmlHelpA = (PFN_HtmlHelpA)GetProcAddress(hLibHH, "HtmlHelpA");
    if (pfnHtmlHelpA != NULL)
    {
        pfnHtmlHelpA(NULL, input_file, HH_DISPLAY_TOPIC, NULL);
        pfnHtmlHelpA(NULL, NULL, HH_CLOSE_ALL, NULL);
    }
    return 0;
}


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

Fuzz HtmlHelpA

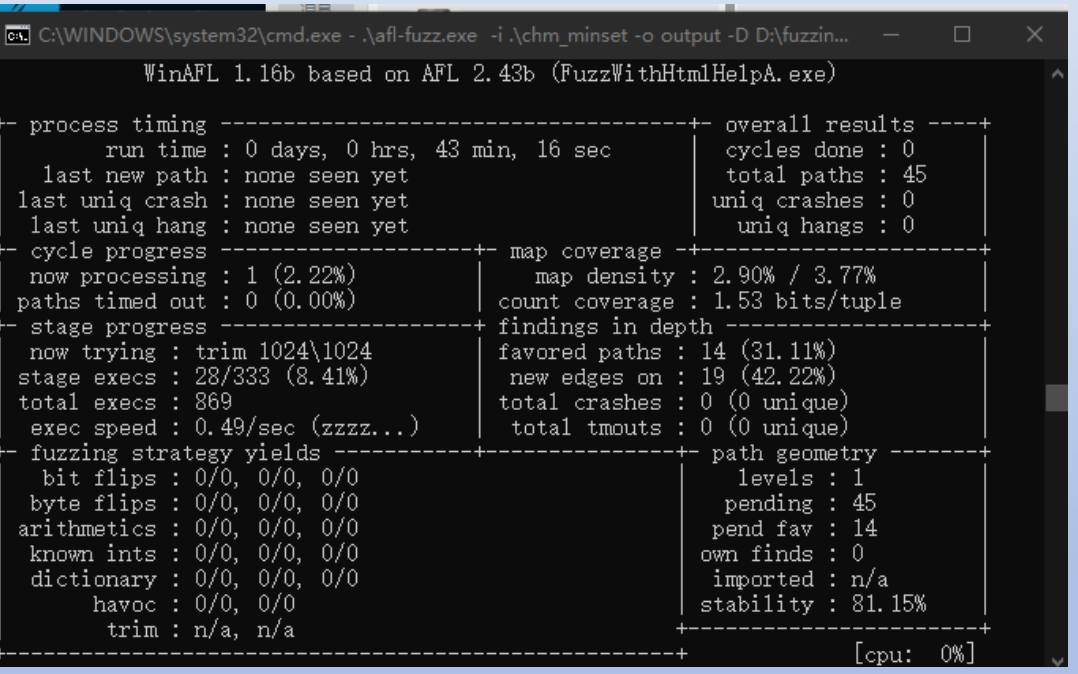
- 获取代码覆盖率
  - drrun.exe -t drcov -- .\FuzzWithHtmlHelpA.exe .\chm_corpus\0cb03cb986acfc2fc0140ec8d41e3515671bf76dfa2fee029b07d6444c957756.chm

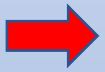
- 精简输入数据集corpus
  - C:\python27\python.exe .\winafl-cmin.py -D D:\fuzzing\DynamoRIO8\bin64 -t 20000 -i chm_corpus -o .\chm_minset -covtype edge -coverage_module hhctrl.ocx -target_module FuzzWithHtmlHelpA.exe -target_method fuzz -nargs 2 -v -- FuzzWithHtmlHelpA.exe @@

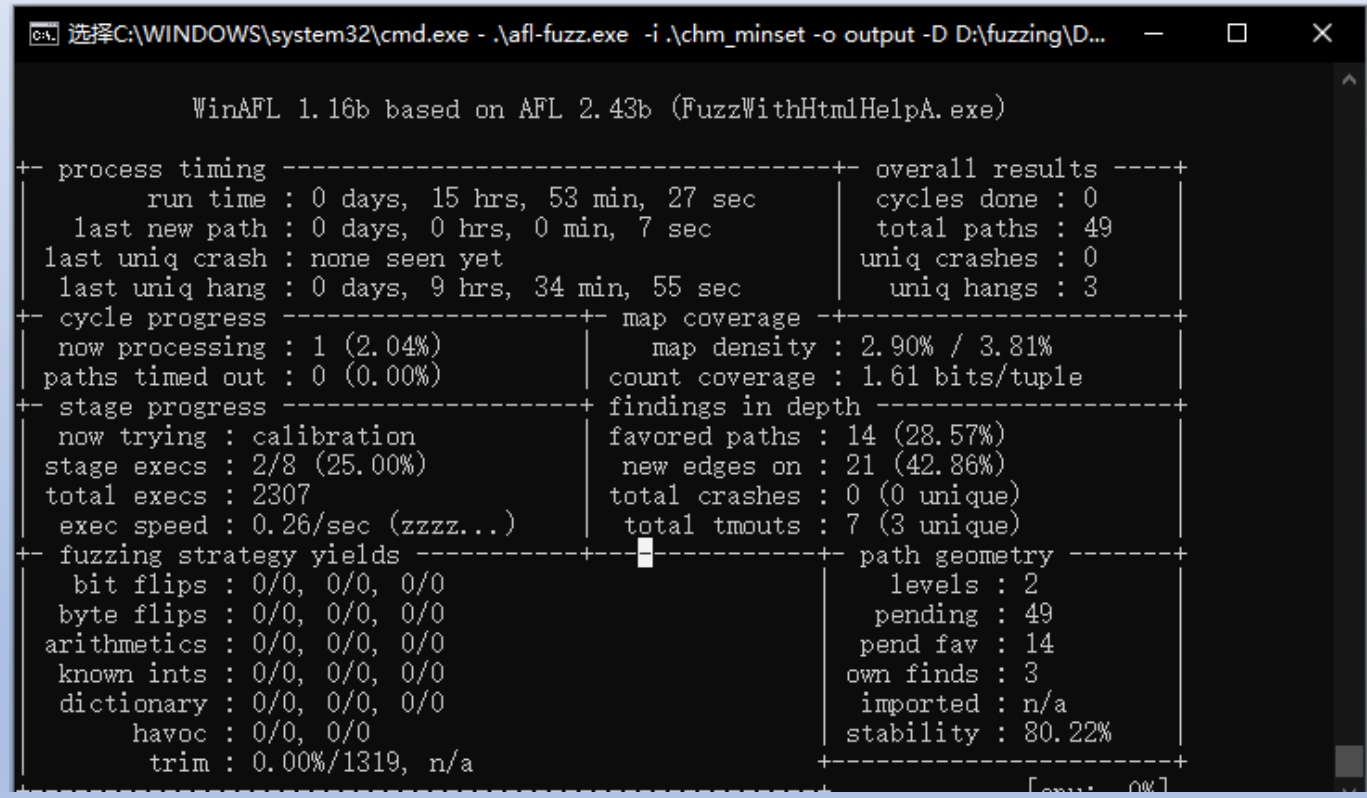
Fuzz HtmlHelpA



Fuzz HtmlHelpA







优化前的思考

- 速度过慢 zzzz

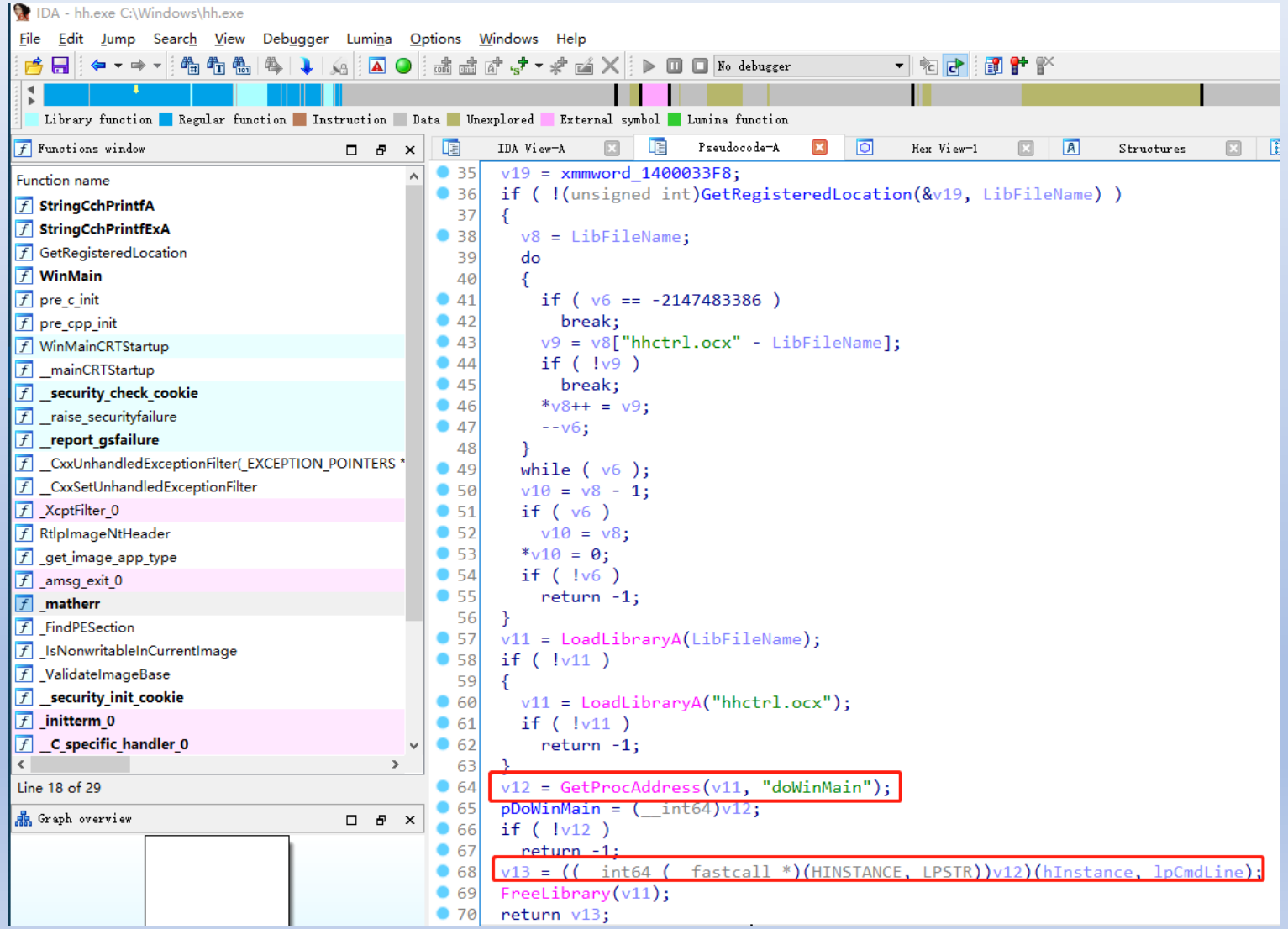
- 有没有更好的办法?

- 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



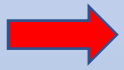
使用doWinMain优化

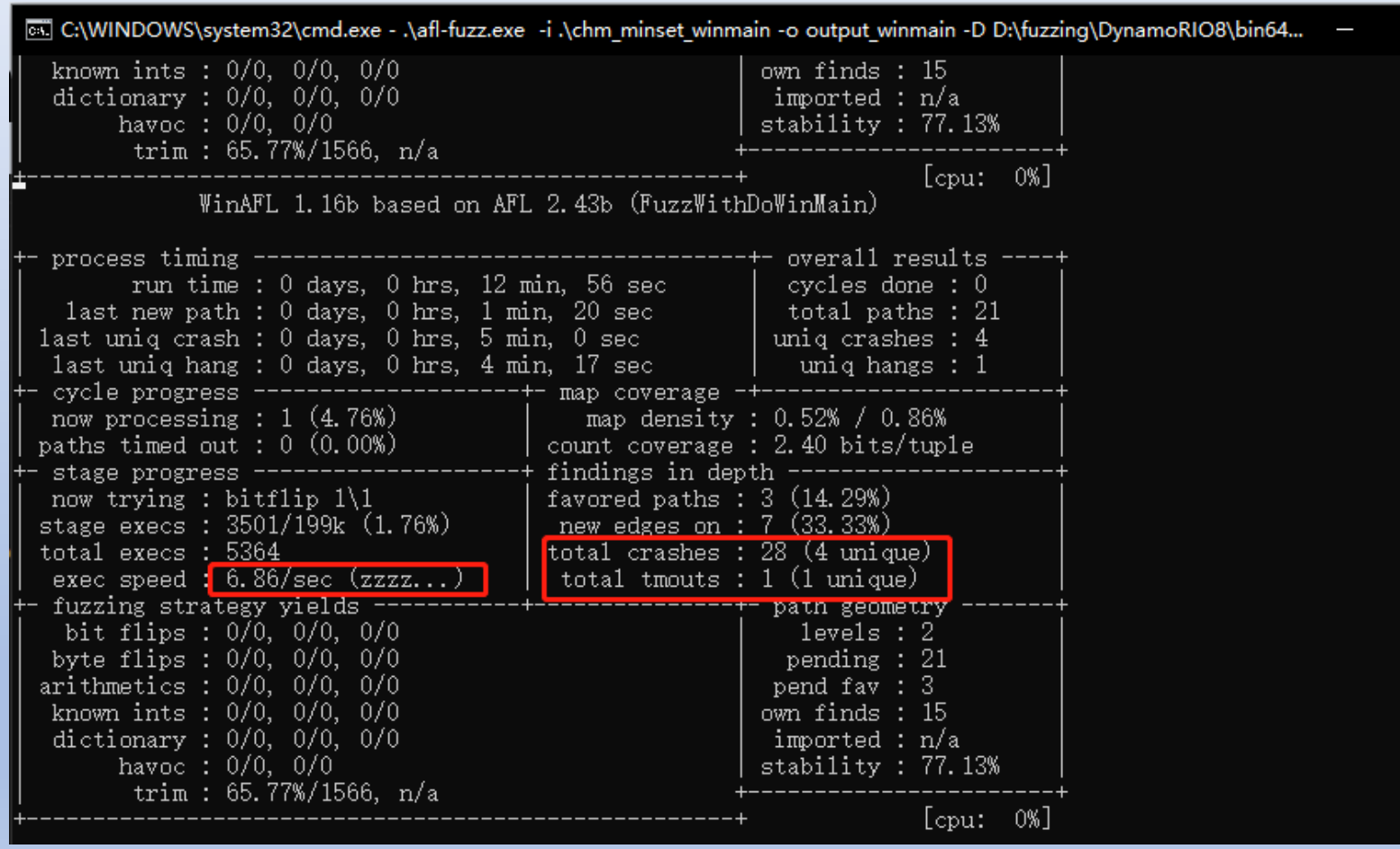
```
__int64 __fastcall doWinMain(HINSTANCE hModule, char *a2)
{
  unsigned int v4; // ebx
  int v6[6]; // [rsp+20h] [rbp-18h] BYREF

  v6[0] = 0;
  v4 = -1;
  if ( InitializeSession((unsigned __int64)v6) )
  {
    WinSqmIncrementDWORD(0i64, 2400i64, 1i64);
    v4 = doInternalWinMain(hModule, a2);
    if ( g_fCoInitialized )
    {
      OleUninitialize();
      g_fCoInitialized = 0;
    }
  }
  return v4;
}
```

```
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: 00007ff7`27420000 00007ff7`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\msvcp_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,rsp
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  ae3fd8fcda86803.
000001a7`22913bf2  006d6863 abababab abababab abababab  chm.............
000001a7`22913c02  abababab feeefeee feeefeee 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\\hhctrl.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;
}
```





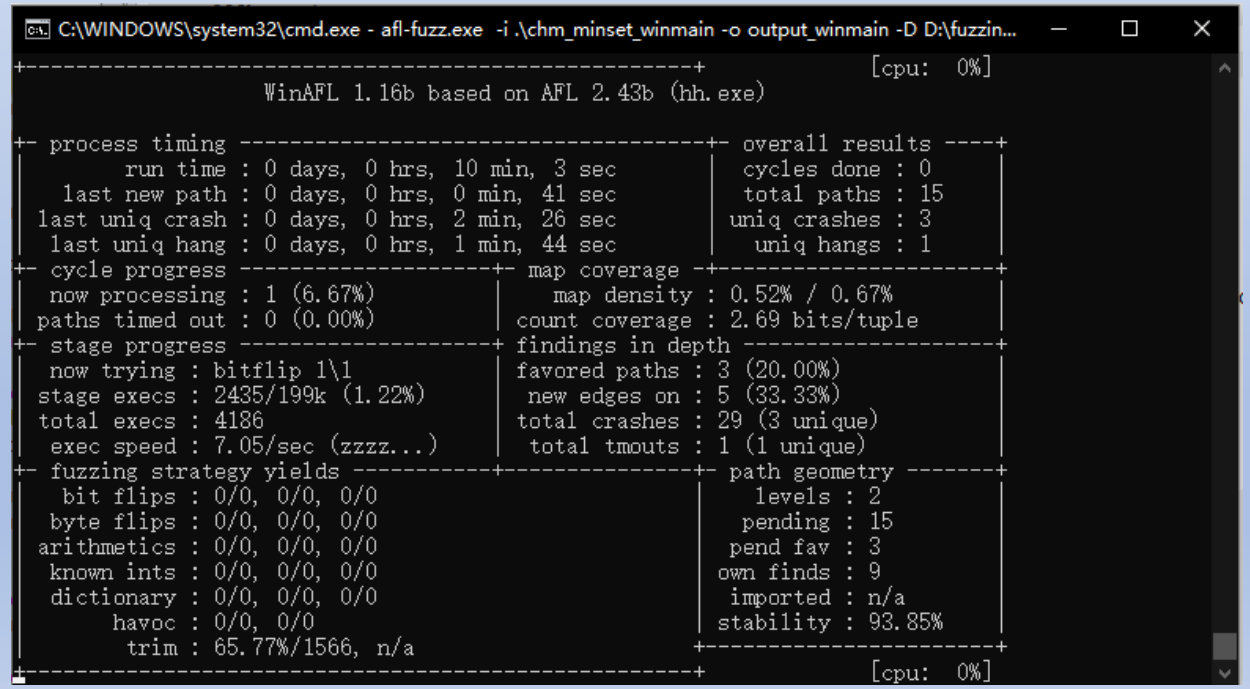
进一步优化

```
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);
            }
        }
```

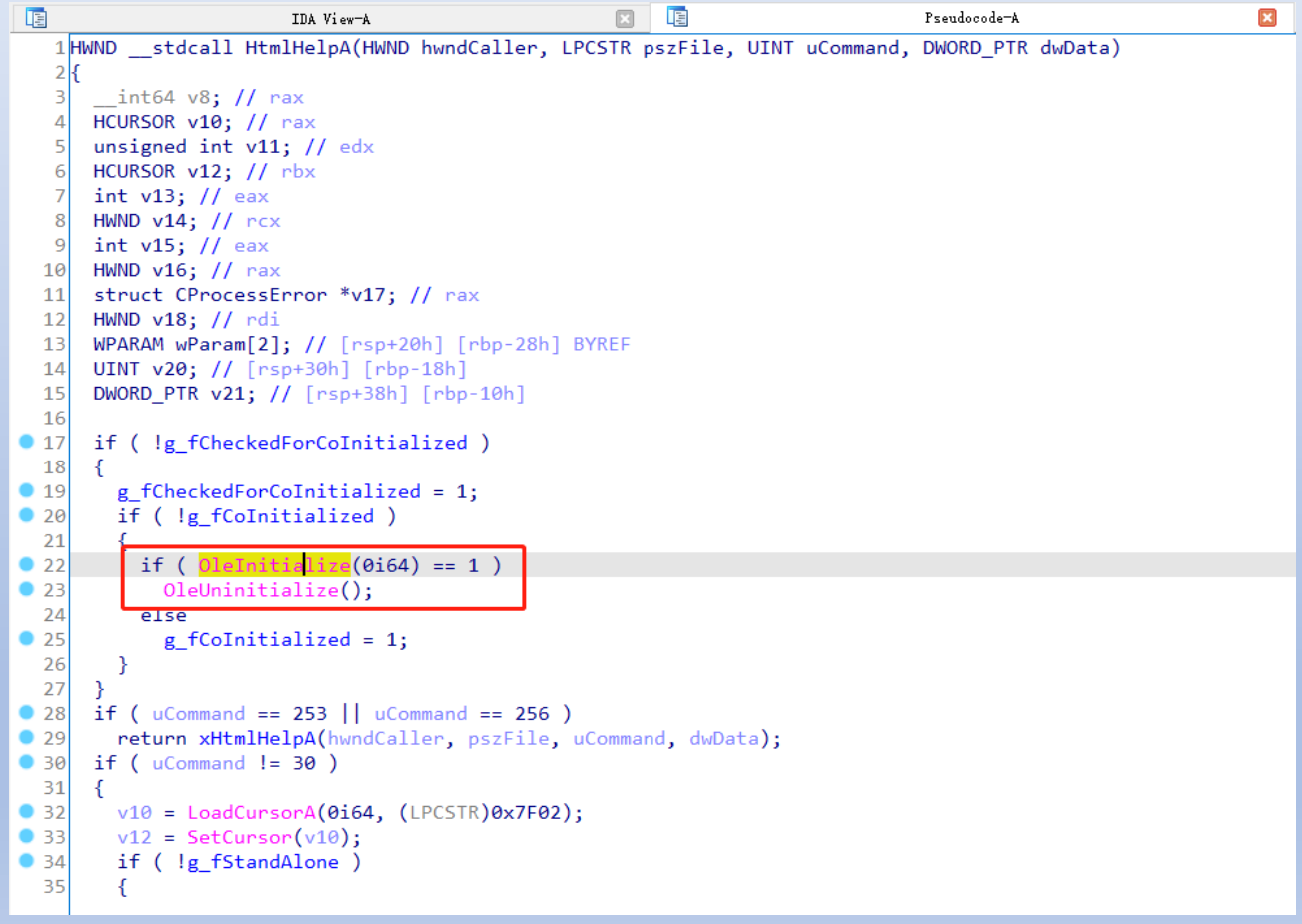
```
99  On Windows, the <tt>_NT_SYMBOL_PATH</tt> environment variable is honored by
100 \p drsyms as a local cache of \p pdb files.  However, \p drsyms does not
101 support symbol store paths (those that contain \p srv) when used inside of
102 a DynamoRIO client.  Such paths should work fine when used in standalone
103 applications, provided that both \p symsrv.dll and \p dbghelp.dll are
104 locatable by the Windows loader.
105
106 \section sec_drsyms_exports Exported Functions
107
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
113 - dr_get_proc_address()
114 - dr_get_application_name()
115 - dr_register_module_load_event()
116 - dr_lookup_module()
117 - dr_lookup_module_by_name()
118 - dr_module_iterator_start()
```

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 @@



再进一步优化?


IDA View-A
Pseudocode-A
HWND __stdcall HtmlHelpA(HWND hwndCaller, LPCSTR pszFile, UINT uCommand, DWORD_PTR dwData)
{
__int64 v8; // rax
HCURSOR v10; // rax
unsigned int v11; // edx
HCURSOR v12; // rbx
int v13; // eax
HWND v14; // rcx
int v15; // eax
HWND v16; // rax
struct CProcessError *v17; // rax
HWND v18; // rdi
WPARAM wParam[2]; // [rsp+20h] [rbp-28h] BYREF
UINT v20; // [rsp+30h] [rbp-18h]
DWORD_PTR v21; // [rsp+38h] [rbp-10h]
if (!g_fCheckedForCoInitialized)
{
g_fCheckedForCoInitialized = 1;
if (!g_fCoInitialized)
{
if (OleInitialize(0i64) == 1)
OleUninitialize();
else
g_fCoInitialized = 1;
}
}
if (uCommand == 253 || uCommand == 256)
return xHtmlHelpA(hwndCaller, pszFile, uCommand, dwData);
if (uCommand != 30)
{
v10 = LoadCursorA(0i64, (LPCSTR)0x7F02);
v12 = SetCursor(v10);
if (!g_fStandAlone)
{

结果处理

uniq_crash_type_2020-01-17_domain.txt

D: > Dropbox > fuzzing > scripts > uniq_crash_type_2020-01-17_domain.txt

```
itss!DllGetClassObject+0x195a7 -> itss!DllGetClassObject+0xc948:
    [+] ..\TestHH\output\test_hh\crashes\id_000000_00
    [+] ..\TestHH\output\test_hh\crashes\id_000002_00
    [+] ..\TestHH\output\test_hh2\crashes_20200117143458\id_000000_00
itss!DllGetClassObject+0x195a7 -> itss!DllGetClassObject+0x4fe2:
    [+] ..\TestHH\output\test_hh\crashes\id_000001_00
    [+] ..\TestHH\output\test_hh\crashes\id_000003_00
    [+] ..\TestHH\output\test_hh\crashes\id_000009_00
    [+] ..\TestHH\output\test_hh2\crashes_20200117143458\id_000001_00
    [+] ..\TestHH\output\test_hh2\crashes_20200117143458\id_000002_00
    [+] ..\TestHH\output\test_hh2\crashes_20200117143458\id_000003_00
    [+] ..\TestHH\output\test_hh2\crashes_20200117143458\id_000004_00
    [+] ..\TestHH\output\test_hh2\crashes_20200117143458\id_000008_00
    [+] ..\TestHH\output\test_hh2\crashes_20200117143458\id_000009_00
itss!DllGetClassObject+0x18508 -> itss!DllGetClassObject+0x80e4:
    [+] ..\TestHH\output\test_hh\crashes\id_000004_00
    [+] ..\TestHH\output\test_hh\crashes\id_000005_00
    [+] ..\TestHH\output\test_hh\crashes\id_000006_00
    [+] ..\TestHH\output\test_hh\crashes\id_000008_00
    [+] ..\TestHH\output\test_hh\crashes\id_000011_00
    [+] ..\TestHH\output\test_hh\crashes\id_000013_00
    [+] ..\TestHH\output\test_hh\crashes\id_000015_00
    [+] ..\TestHH\output\test_hh\crashes\id_000016_00
    [+] ..\TestHH\output\test_hh2\crashes_20200117143458\id_000006_00
    [+] ..\TestHH\output\test_hh2\crashes_20200117143458\id_000012_00
    [+] ..\TestHH\output\test_hh2\crashes_20200117143458\id_000014_00
    [+] ..\TestHH\output\test_hh2\crashes_20200117143458\id_000016_00
    [+] ..\TestHH\output\test_hh2\crashes_20200117143458\id_000017_00
    [+] ..\TestHH\output\test_hh2\crashes_20200117143458\id_000018_00
    [+] ..\TestHH\output\test_hh2\crashes_20200117143458\id_000019_00
> itss!DllGetClassObject+0x108df -> itss!DllGetClassObject+0x6de5: ···
ntdll!RtlpBreakPointHeap+0x16 -> ntdll!RtlpCheckBusyBlockTail+0x20d:
    [+] ..\TestHH\output\test_hh\crashes\id_000012_00
    [+] ..\TestHH\output\test_hh\hangs\id_000001
> ntdll!NtTerminateProcess+0x14 -> ntdll!RtlExitUserProcess+0xb8: ···
itss!DllGetClassObject+0x3706 -> itss!DllGetClassObject+0x2d72:
    [+] ..\TestHH\output\test_hh2\crashes_20200117143458\id_000013_00
    [+] ..\TestHH\output\test_hh2\hangs_20200117143458\id_000001
```

```
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 00007ff7`29521226 hhctrl!doWinMain+0x5e
08 000000ea`e0affbe0 00007ff7`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
- winafl源码：
  https://github.com/googleprojectzero/winafl
- 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?