HexRaysCodeXplorer: make object-oriented RE easier

Aleksandr MatrosovEugene Rodionov

@matrosov @vxradius

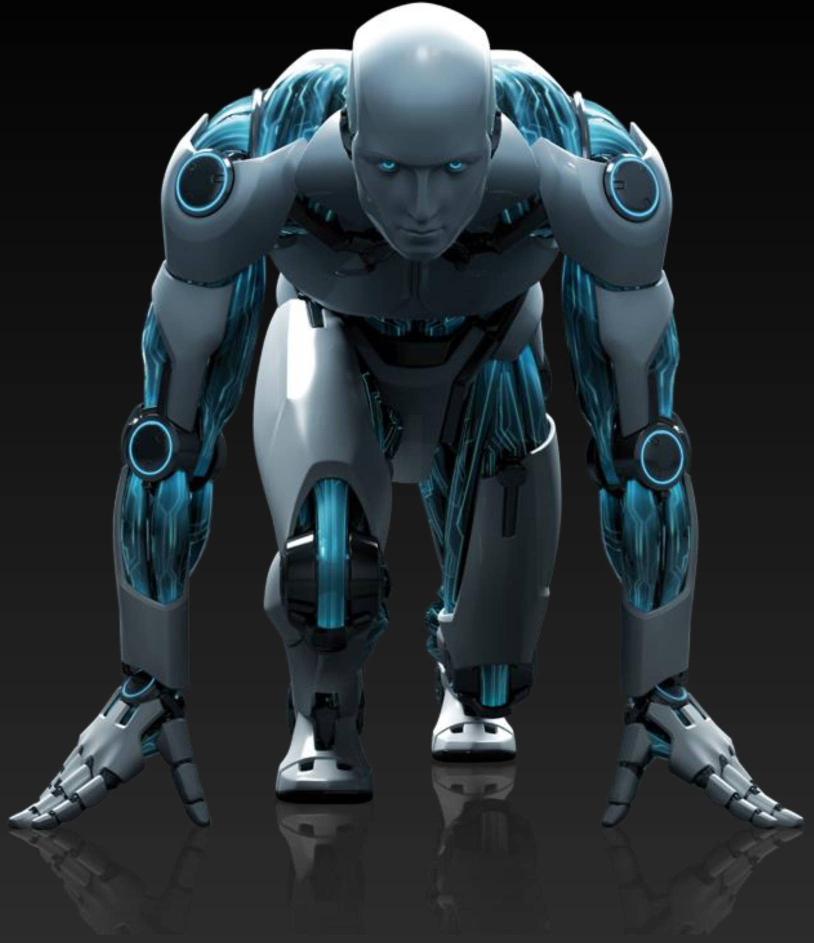
C++ Code Reconstruction Problems

- **Object identification**
  - ✓ Type reconstruction

- **Class layout reconstruction**
  - ✓ Identify constructors/destructors
  - ✓ Identify class members
  - ✓ Local/global type reconstruction
  - ✓ Associate object with exact method calls

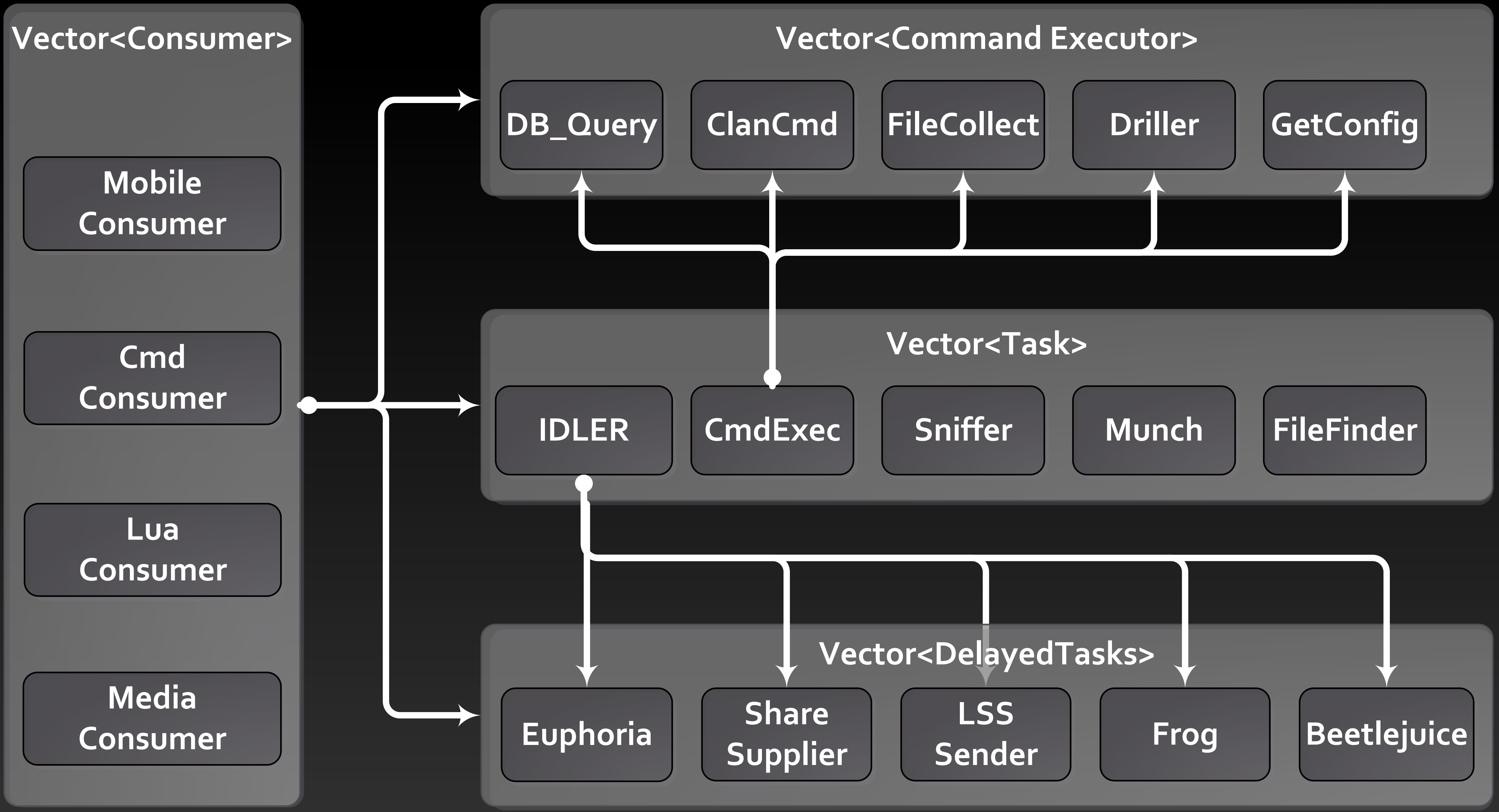
- **RTTI reconstruction**
  - ✓ vftable reconstruction
  - ✓ Associate vftable object with exact object
  - ✓ Class hierarchy reconstruction

Depend on compiler design

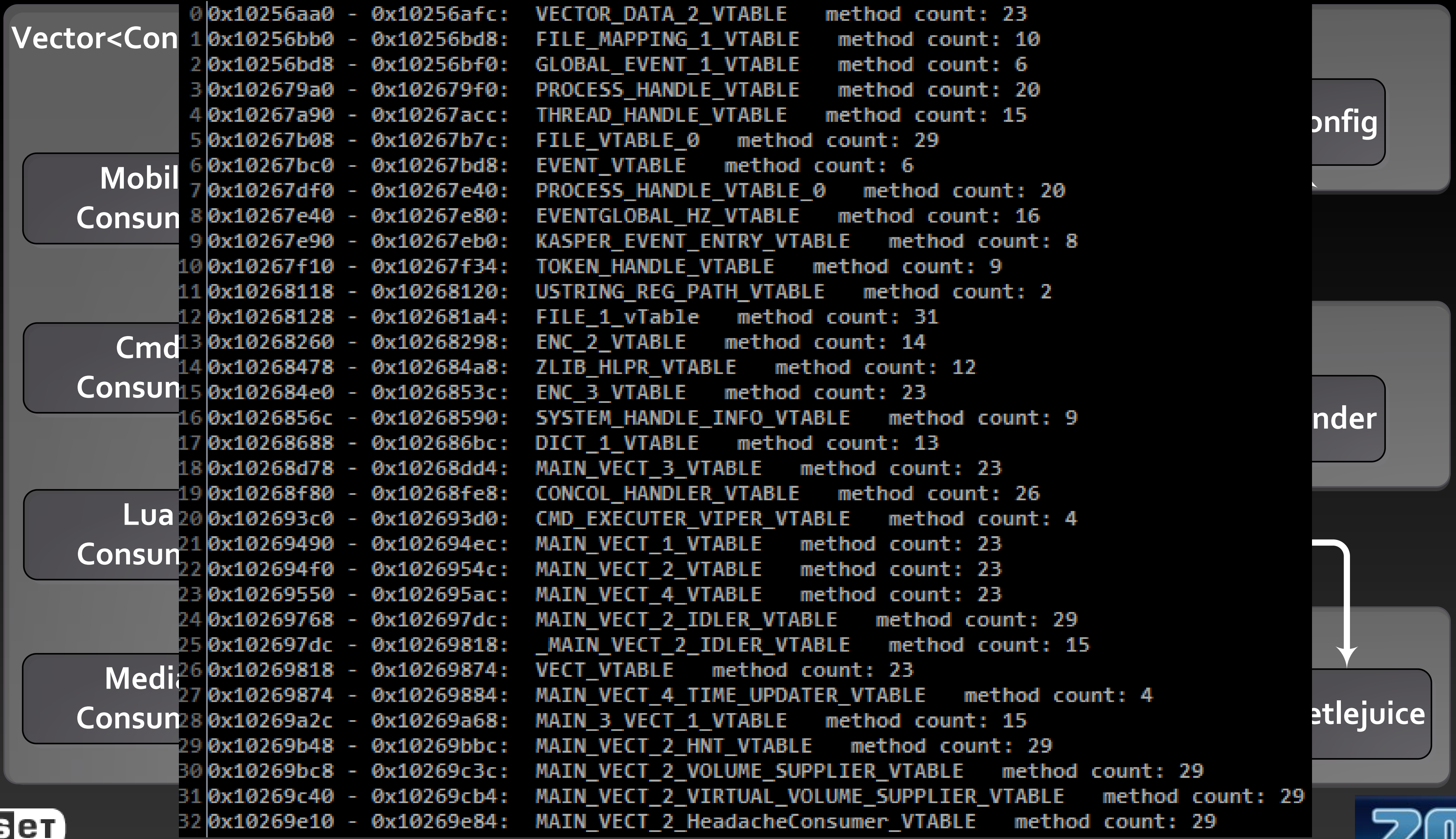


C++ Code Reconstruction: the truth is out there

An overview of the Flamer Framework



An overview of the Flamer Framework



Vector<Con

Mobil
Consun

Cmd
Consun

Lua
Consun

Medi
Consun

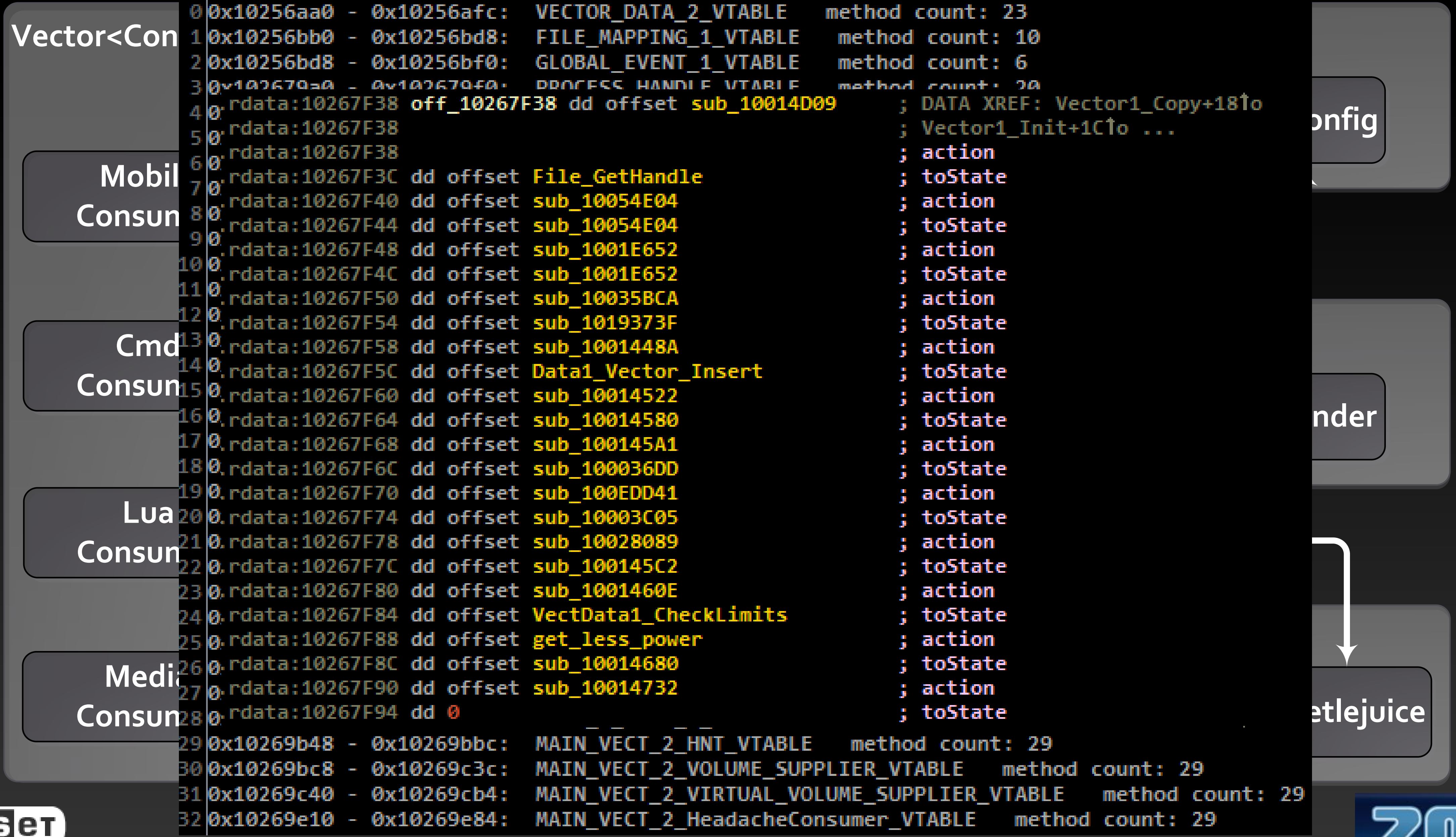
onfig

nder

etlejuice

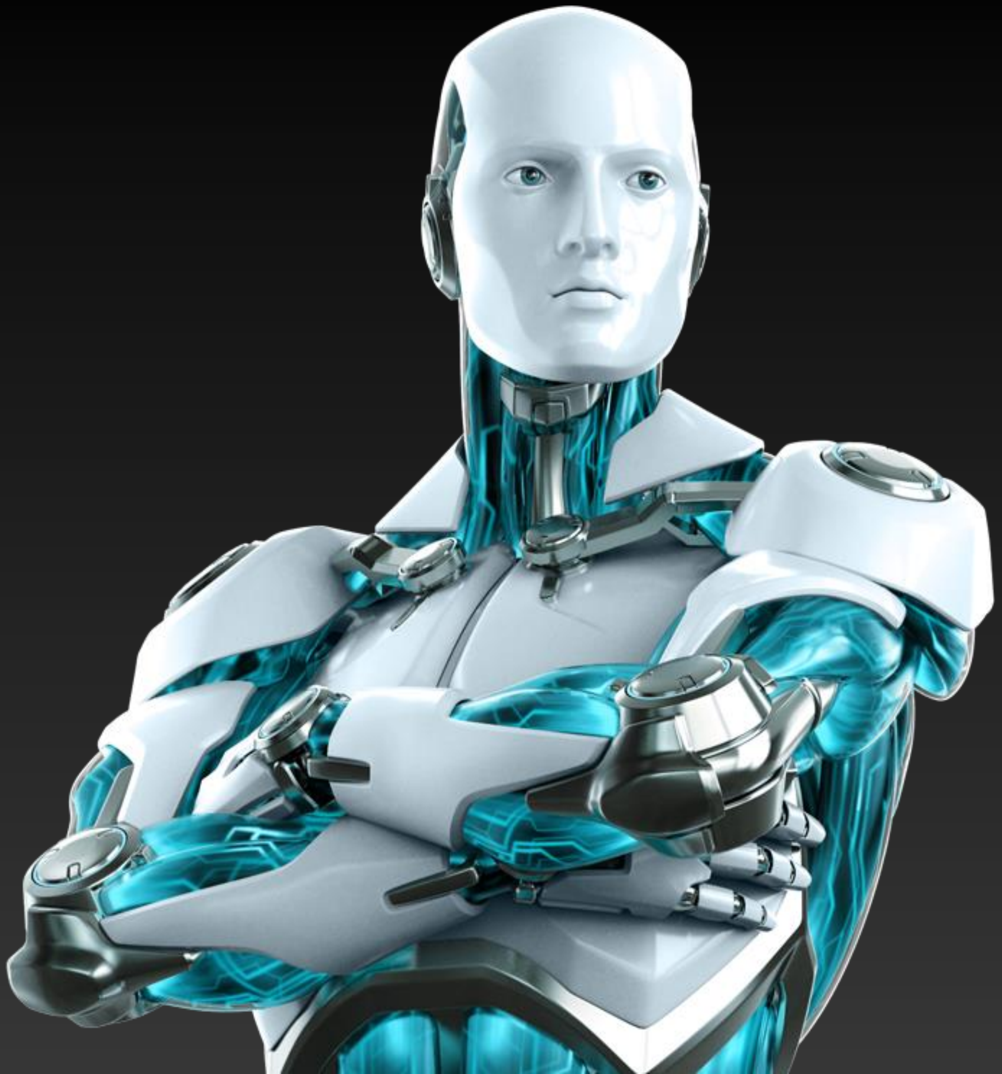
```
 0 0x10256aa0 - 0x10256afc:  VECTOR_DATA_2_VTABLE   method count: 23
 1 0x10256bb0 - 0x10256bd8:  FILE_MAPPING_1_VTABLE   method count: 10
 2 0x10256bd8 - 0x10256bf0:  GLOBAL_EVENT_1_VTABLE   method count: 6
 3 0x102679a0 - 0x102679f0:  PROCESS_HANDLE_VTABLE   method count: 20
 4 0x10267a90 - 0x10267acc:  THREAD_HANDLE_VTABLE   method count: 15
 5 0x10267b08 - 0x10267b7c:  FILE_VTABLE_0   method count: 29
 6 0x10267bc0 - 0x10267bd8:  EVENT_VTABLE   method count: 6
 7 0x10267df0 - 0x10267e40:  PROCESS_HANDLE_VTABLE_0   method count: 20
 8 0x10267e40 - 0x10267e80:  EVENTGLOBAL_HZ_VTABLE   method count: 16
 9 0x10267e90 - 0x10267eb0:  KASPER_EVENT_ENTRY_VTABLE   method count: 8
10 0x10267f10 - 0x10267f34:  TOKEN_HANDLE_VTABLE   method count: 9
11 0x10268118 - 0x10268120:  USTRING_REG_PATH_VTABLE   method count: 2
12 0x10268128 - 0x102681a4:  FILE_1_vTable   method count: 31
13 0x10268260 - 0x10268298:  ENC_2_VTABLE   method count: 14
14 0x10268478 - 0x102684a8:  ZLIB_HLPR_VTABLE   method count: 12
15 0x102684e0 - 0x1026853c:  ENC_3_VTABLE   method count: 23
16 0x1026856c - 0x10268590:  SYSTEM_HANDLE_INFO_VTABLE   method count: 9
17 0x10268688 - 0x102686bc:  DICT_1_VTABLE   method count: 13
18 0x10268d78 - 0x10268dd4:  MAIN_VECT_3_VTABLE   method count: 23
19 0x10268f80 - 0x10268fe8:  CONCOL_HANDLER_VTABLE   method count: 26
20 0x102693c0 - 0x102693d0:  CMD_EXECUTER_VIPER_VTABLE   method count: 4
21 0x10269490 - 0x102694ec:  MAIN_VECT_1_VTABLE   method count: 23
22 0x102694f0 - 0x1026954c:  MAIN_VECT_2_VTABLE   method count: 23
23 0x10269550 - 0x102695ac:  MAIN_VECT_4_VTABLE   method count: 23
24 0x10269768 - 0x102697dc:  MAIN_VECT_2_IDLER_VTABLE   method count: 29
25 0x102697dc - 0x10269818:  _MAIN_VECT_2_IDLER_VTABLE   method count: 15
26 0x10269818 - 0x10269874:  VECT_VTABLE   method count: 23
27 0x10269874 - 0x10269884:  MAIN_VECT_4_TIME_UPDATER_VTABLE   method count: 4
28 0x10269a2c - 0x10269a68:  MAIN_3_VECT_1_VTABLE   method count: 15
29 0x10269b48 - 0x10269bbc:  MAIN_VECT_2_HNT_VTABLE   method count: 29
30 0x10269bc8 - 0x10269c3c:  MAIN_VECT_2_VOLUME_SUPPLIER_VTABLE   method count: 29
31 0x10269c40 - 0x10269cb4:  MAIN_VECT_2_VIRTUAL_VOLUME_SUPPLIER_VTABLE   method count: 29
32 0x10269e10 - 0x10269e84:  MAIN_VECT_2_HeadacheConsumer_VTABLE   method count: 29
```

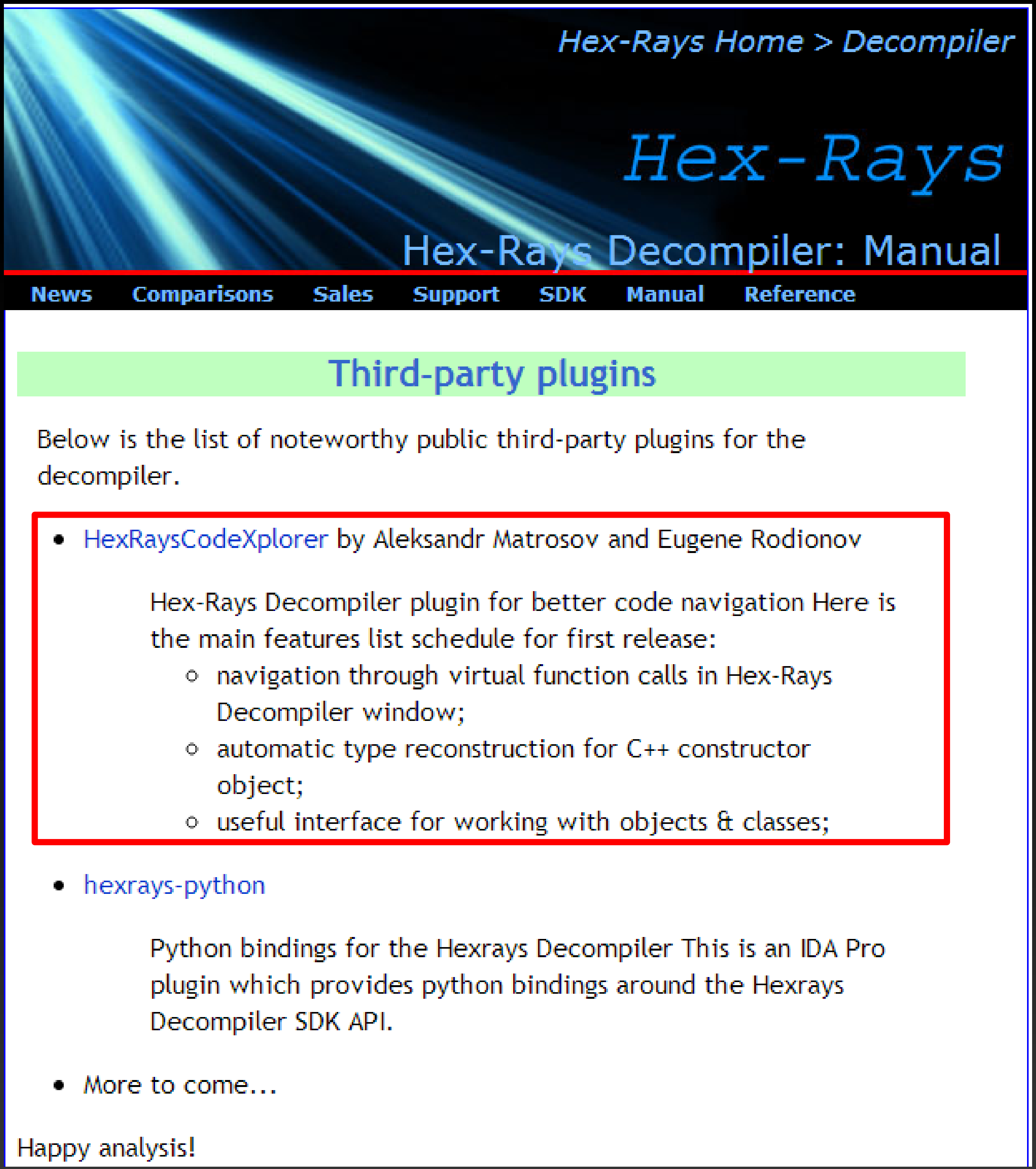
An overview of the Flamer Framework



```
0x10256aa0 - 0x10256afc:  VECTOR_DATA_2_VTABLE   method count: 23
0x10256bb0 - 0x10256bd8:  FILE_MAPPING_1_VTABLE   method count: 10
0x10256bd8 - 0x10256bf0:  GLOBAL_EVENT_1_VTABLE   method count: 6
0x102679a0 - 0x102679f0:  PROCESS_HANDLE_VTABLE   method count: 20
.rdata:10267F38 off_10267F38 dd offset sub_10014D09      ; DATA XREF: Vector1_Copy+18↑o
.rdata:10267F38                                          ; Vector1_Init+1C↑o ...
.rdata:10267F38                                          ; action
.rdata:10267F3C dd offset File_GetHandle                 ; toState
.rdata:10267F40 dd offset sub_10054E04                   ; action
.rdata:10267F44 dd offset sub_10054E04                   ; toState
.rdata:10267F48 dd offset sub_1001E652                   ; action
.rdata:10267F4C dd offset sub_1001E652                   ; toState
.rdata:10267F50 dd offset sub_10035BCA                   ; action
.rdata:10267F54 dd offset sub_1019373F                   ; toState
.rdata:10267F58 dd offset sub_1001448A                   ; action
.rdata:10267F5C dd offset Data1_Vector_Insert            ; toState
.rdata:10267F60 dd offset sub_10014522                   ; action
.rdata:10267F64 dd offset sub_10014580                   ; toState
.rdata:10267F68 dd offset sub_100145A1                   ; action
.rdata:10267F6C dd offset sub_100036DD                   ; toState
.rdata:10267F70 dd offset sub_100EDD41                   ; action
.rdata:10267F74 dd offset sub_10003C05                   ; toState
.rdata:10267F78 dd offset sub_10028089                   ; action
.rdata:10267F7C dd offset sub_100145C2                   ; toState
.rdata:10267F80 dd offset sub_1001460E                   ; action
.rdata:10267F84 dd offset VectData1_CheckLimits          ; toState
.rdata:10267F88 dd offset get_less_power                 ; action
.rdata:10267F8C dd offset sub_10014680                   ; toState
.rdata:10267F90 dd offset sub_10014732                   ; action
.rdata:10267F94 dd 0                                     ; toState
0x10269b48 - 0x10269bbc:  MAIN_VECT_2_HNT_VTABLE   method count: 29
0x10269bc8 - 0x10269c3c:  MAIN_VECT_2_VOLUME_SUPPLIER_VTABLE   method count: 29
0x10269c40 - 0x10269cb4:  MAIN_VECT_2_VIRTUAL_VOLUME_SUPPLIER_VTABLE   method count: 29
0x10269e10 - 0x10269e84:  MAIN_VECT_2_HeadacheConsumer_VTABLE   method count: 29
```

HexRaysCodeXplorer





Third-party plugins

Below is the list of noteworthy public third-party plugins for the decompiler.

- HexRaysCodeXplorer by Aleksandr Matrosov and Eugene Rodionov

  Hex-Rays Decompiler plugin for better code navigation Here is the main features list schedule for first release:
  - navigation through virtual function calls in Hex-Rays Decompiler window;
  - automatic type reconstruction for C++ constructor object;
  - useful interface for working with objects & classes;

- hexrays-python

  Python bindings for the Hexrays Decompiler This is an IDA Pro plugin which provides python bindings around the Hexrays Decompiler SDK API.

- More to come...

Happy analysis!

HexRaysCodeXplorer Features

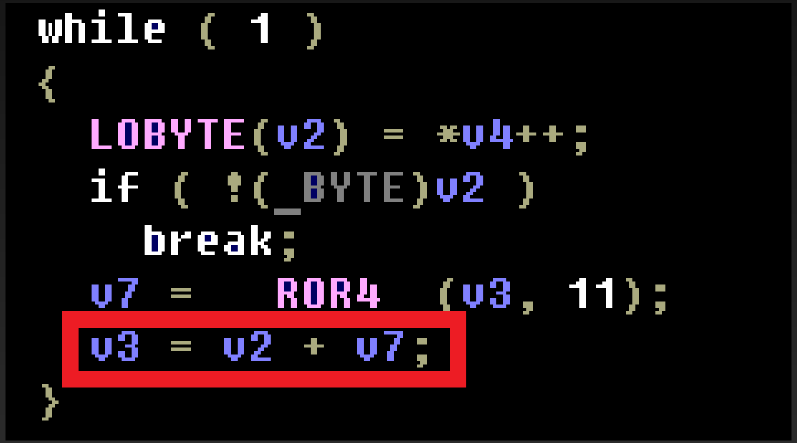
- Hex-Rays decompiler plugin

- The plugin was designed to facilitate static analysis of:
  - ✓ object oriented code
  - ✓ position independent code

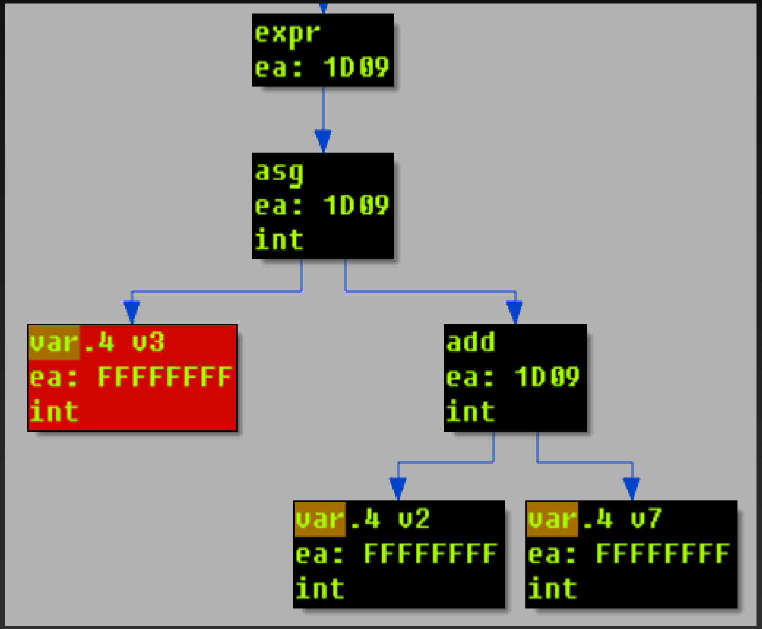
- The plugin allows to:
  - ✓ navigate through decompiled virtual methods
  - ✓ partially reconstruct object type

Hex-Rays Decompiler Plugin SDK

- At the heart of the decompiler lies *ctree* structure:
  - ✓ syntax tree structure
  - ✓ consists of *citem_t* objects
  - ✓ there are 9 maturity levels of the *ctree* structure

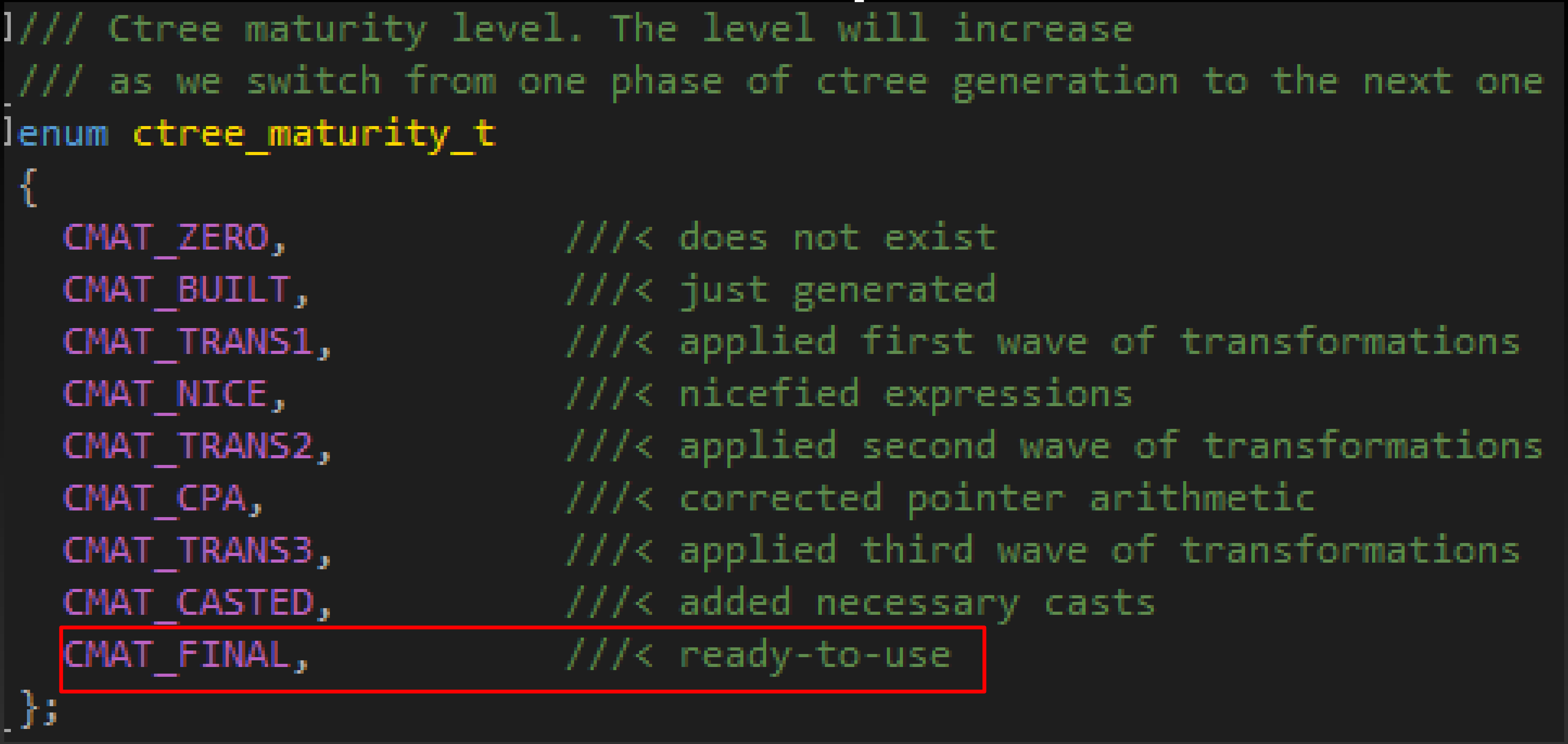


```
while ( 1 )
{
  LOBYTE(v2) = *v4++;
  if ( !(_BYTE)v2 )
    break;
  v7 =   ROR4  (v3, 11);
  v3 = v2 + v7;
}
```



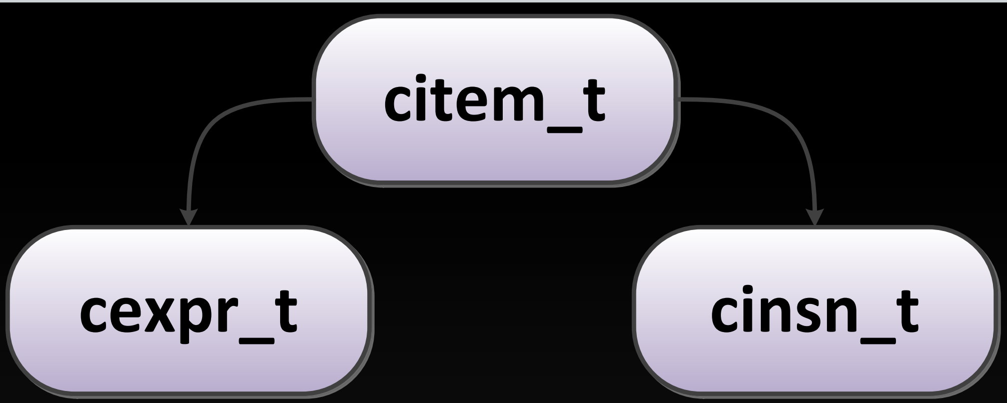
Hex-Rays Decompiler Plugin SDK

- At the heart of the decompiler lies *ctree* structure:



```
/// Ctree maturity level. The level will increase
/// as we switch from one phase of ctree generation to the next one
enum ctree_maturity_t
{
  CMAT_ZERO,            ///< does not exist
  CMAT_BUILT,           ///< just generated
  CMAT_TRANS1,          ///< applied first wave of transformations
  CMAT_NICE,            ///< nicefied expressions
  CMAT_TRANS2,          ///< applied second wave of transformations
  CMAT_CPA,             ///< corrected pointer arithmetic
  CMAT_TRANS3,          ///< applied third wave of transformations
  CMAT_CASTED,          ///< added necessary casts
  CMAT_FINAL,           ///< ready-to-use
};
```

Hex-Rays Decompiler Plugin SDK



- Type *citem_t* is a base class for:
  - ✓ *cexpr_t* – expression type
  - ✓ *cinsn_t* – statement type

- Expressions have attached type information

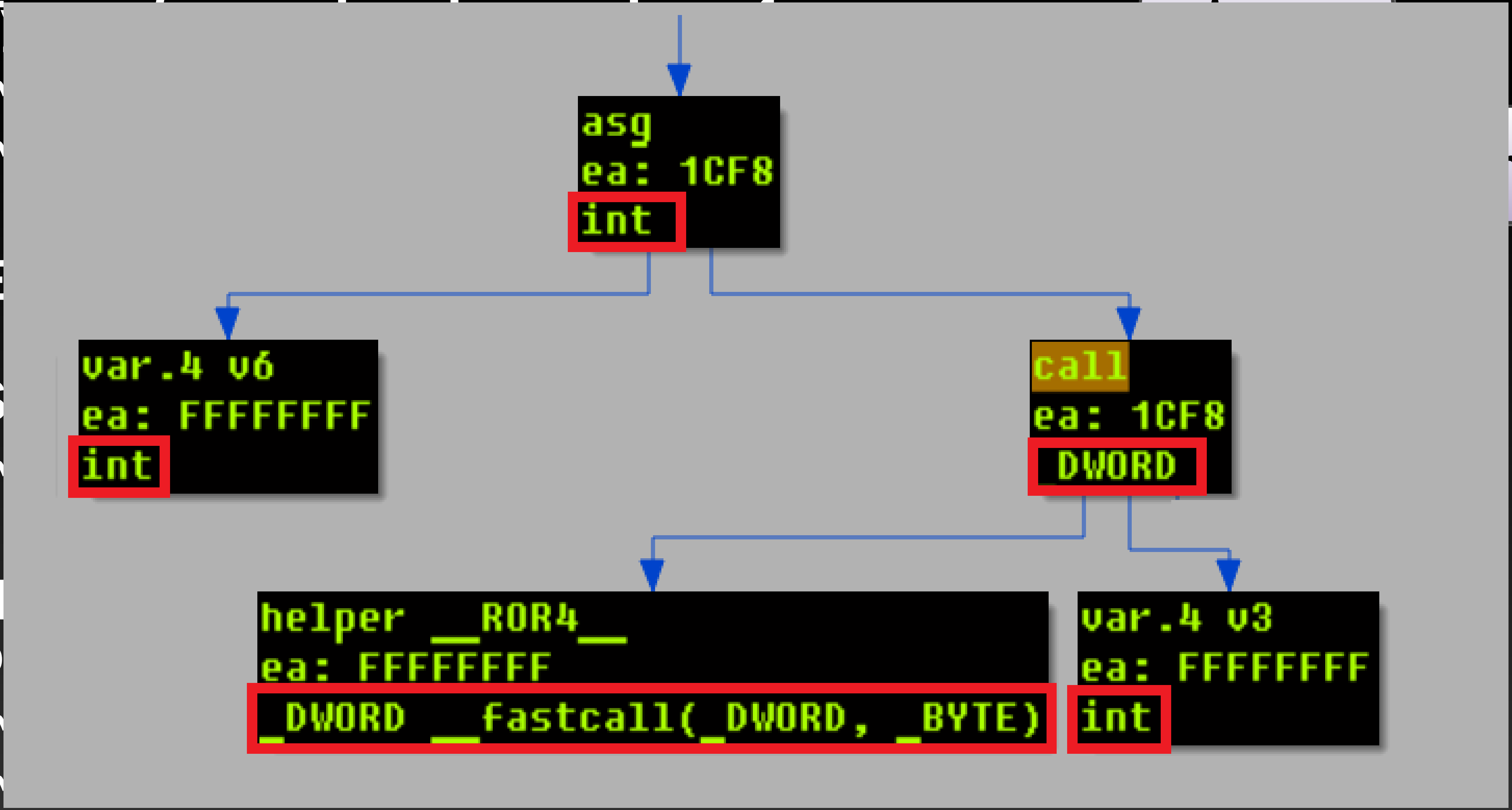
- Statements include:
  - ✓ *block, if, for, while, do, switch, return, goto, asm*

- Hex-Rays provides iterators for traversing the *citem_t* objects within *ctree* structure:
  - ✓ *ctree_visitor_t*
  - ✓ *ctree_parentee_t*

Hex-Rays Decompiler Plugin SDK

- T
- E
- S
- H
  O

n_t



```
asg
ea: 1CF8
int
```

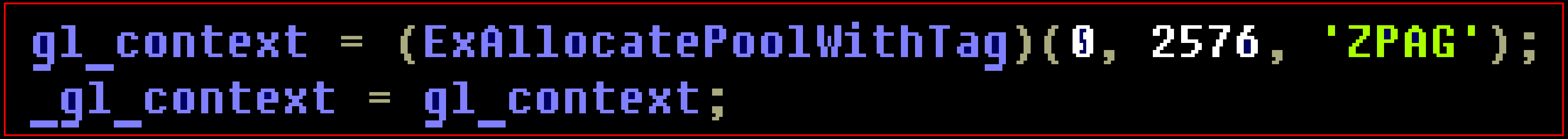
```
var.4 v6
ea: FFFFFFFF
int
```

```
call
ea: 1CF8
DWORD
```

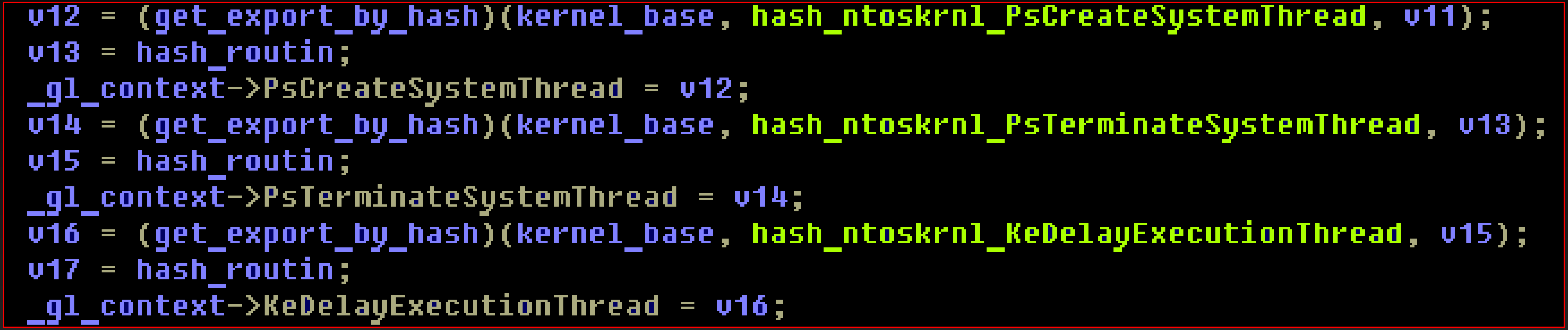
```
helper __ROR4__
ea: FFFFFFFF
_DWORD __fastcall(_DWORD, _BYTE)
```

```
var.4 v3
ea: FFFFFFFF
int
```

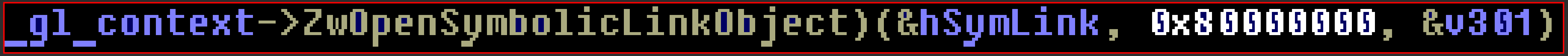
HexRaysCodeXplorer: Gapz Position Independent Code



```
gl_context = (ExAllocatePoolWithTag)(0, 2576, 'ZPAG');
_gl_context = gl_context;
```



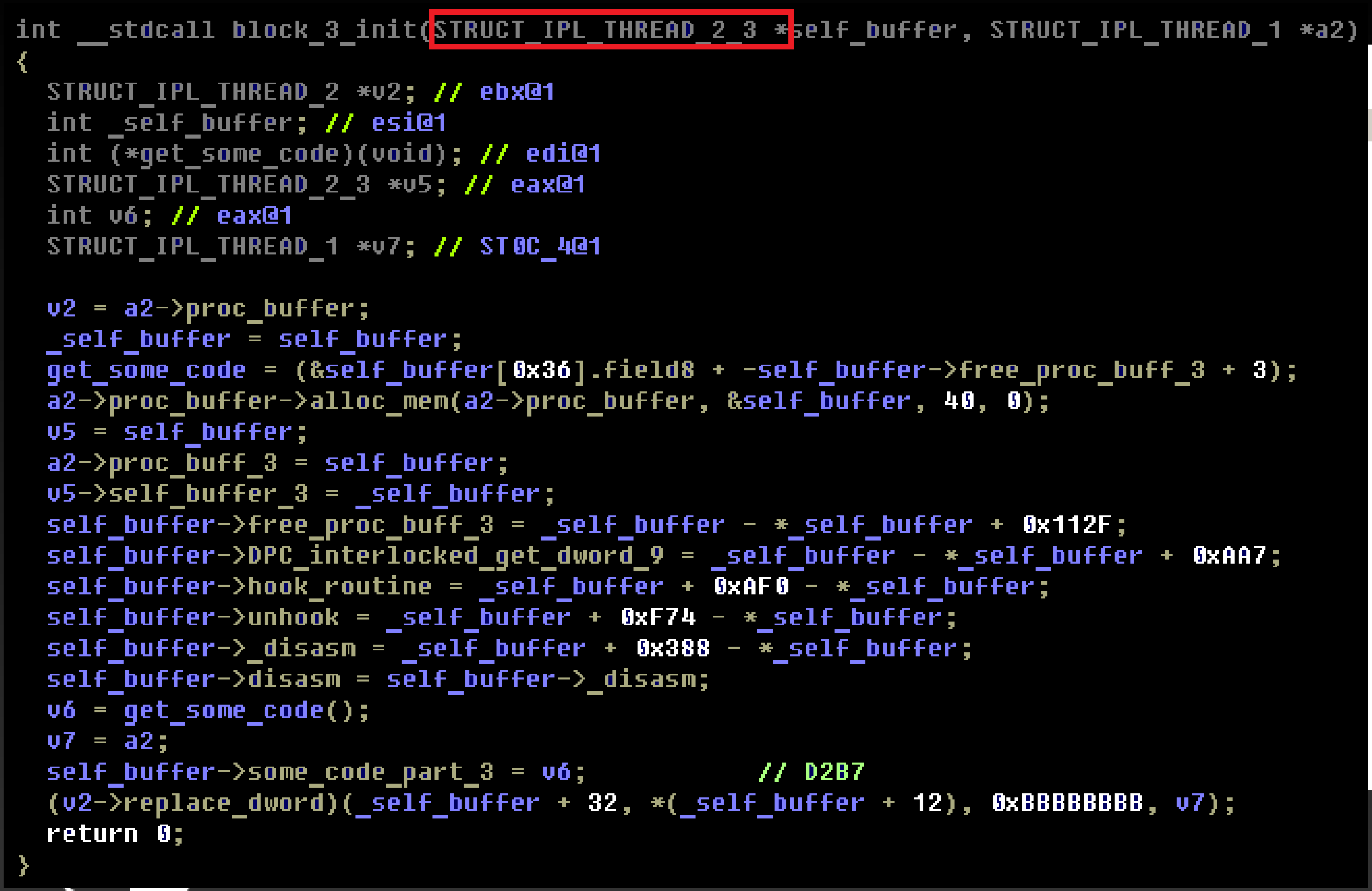
```
v12 = (get_export_by_hash)(kernel_base, hash_ntoskrnl_PsCreateSystemThread, v11);
v13 = hash_routin;
_gl_context->PsCreateSystemThread = v12;
v14 = (get_export_by_hash)(kernel_base, hash_ntoskrnl_PsTerminateSystemThread, v13);
v15 = hash_routin;
_gl_context->PsTerminateSystemThread = v14;
v16 = (get_export_by_hash)(kernel_base, hash_ntoskrnl_KeDelayExecutionThread, v15);
v17 = hash_routin;
_gl_context->KeDelayExecutionThread = v16;
```



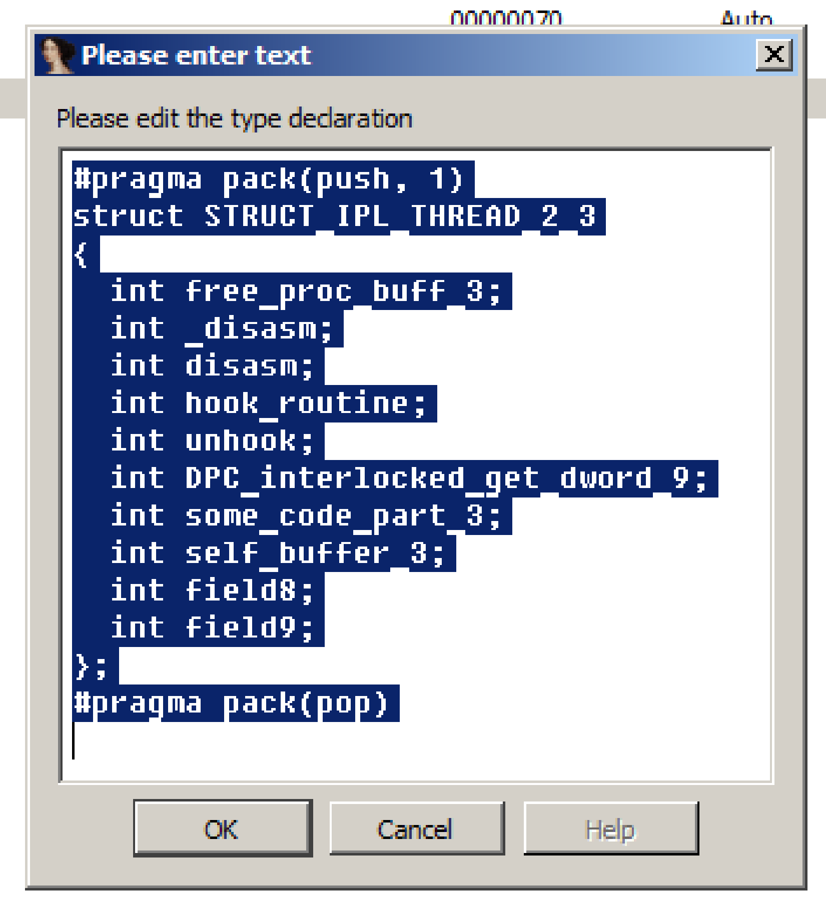
```
_gl_context->ZwOpenSymbolicLinkObject)(&hSymLink, 0x80000000, &v301)
```

HexRaysCodeXplorer: Virtual Methods

- The IDA's "Local Types" is used to represent object type



```
int __stdcall block_3_init(STRUCT_IPL_THREAD_2_3 *self_buffer, STRUCT_IPL_THREAD_1 *a2)
{
  STRUCT_IPL_THREAD_2 *v2; // ebx@1
  int _self_buffer; // esi@1
  int (*get_some_code)(void); // edi@1
  STRUCT_IPL_THREAD_2_3 *v5; // eax@1
  int v6; // eax@1
  STRUCT_IPL_THREAD_1 *v7; // ST0C_4@1

  v2 = a2->proc_buffer;
  _self_buffer = self_buffer;
  get_some_code = (&self_buffer[0x36].field8 + -self_buffer->free_proc_buff_3 + 3);
  a2->proc_buffer->alloc_mem(a2->proc_buffer, &self_buffer, 40, 0);
  v5 = self_buffer;
  a2->proc_buff_3 = self_buffer;
  v5->self_buffer_3 = _self_buffer;
  self_buffer->free_proc_buff_3 = _self_buffer - *_self_buffer + 0x112F;
  self_buffer->DPC_interlocked_get_dword_9 = _self_buffer - *_self_buffer + 0xAA7;
  self_buffer->hook_routine = _self_buffer + 0xAF0 - *_self_buffer;
  self_buffer->unhook = _self_buffer + 0xF74 - *_self_buffer;
  self_buffer->_disasm = _self_buffer + 0x388 - *_self_buffer;
  self_buffer->disasm = self_buffer->_disasm;
  v6 = get_some_code();
  v7 = a2;
  self_buffer->some_code_part_3 = v6;           // D2B7
  (v2->replace_dword)(_self_buffer + 32, *(_self_buffer + 12), 0xBBBBBBBB, v7);
  return 0;
}
```

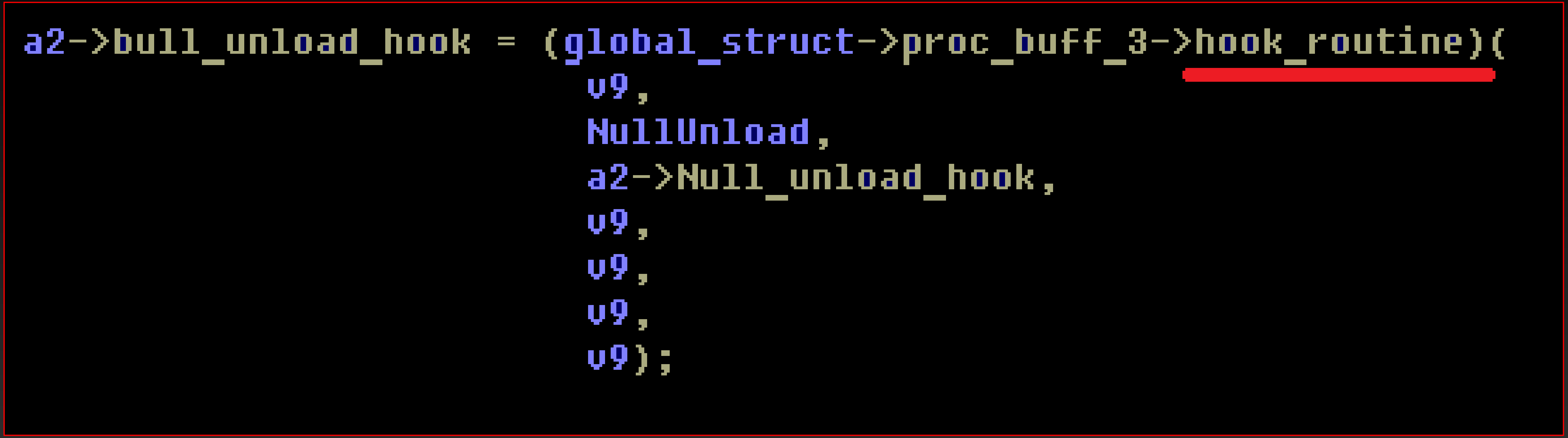


Please edit the type declaration

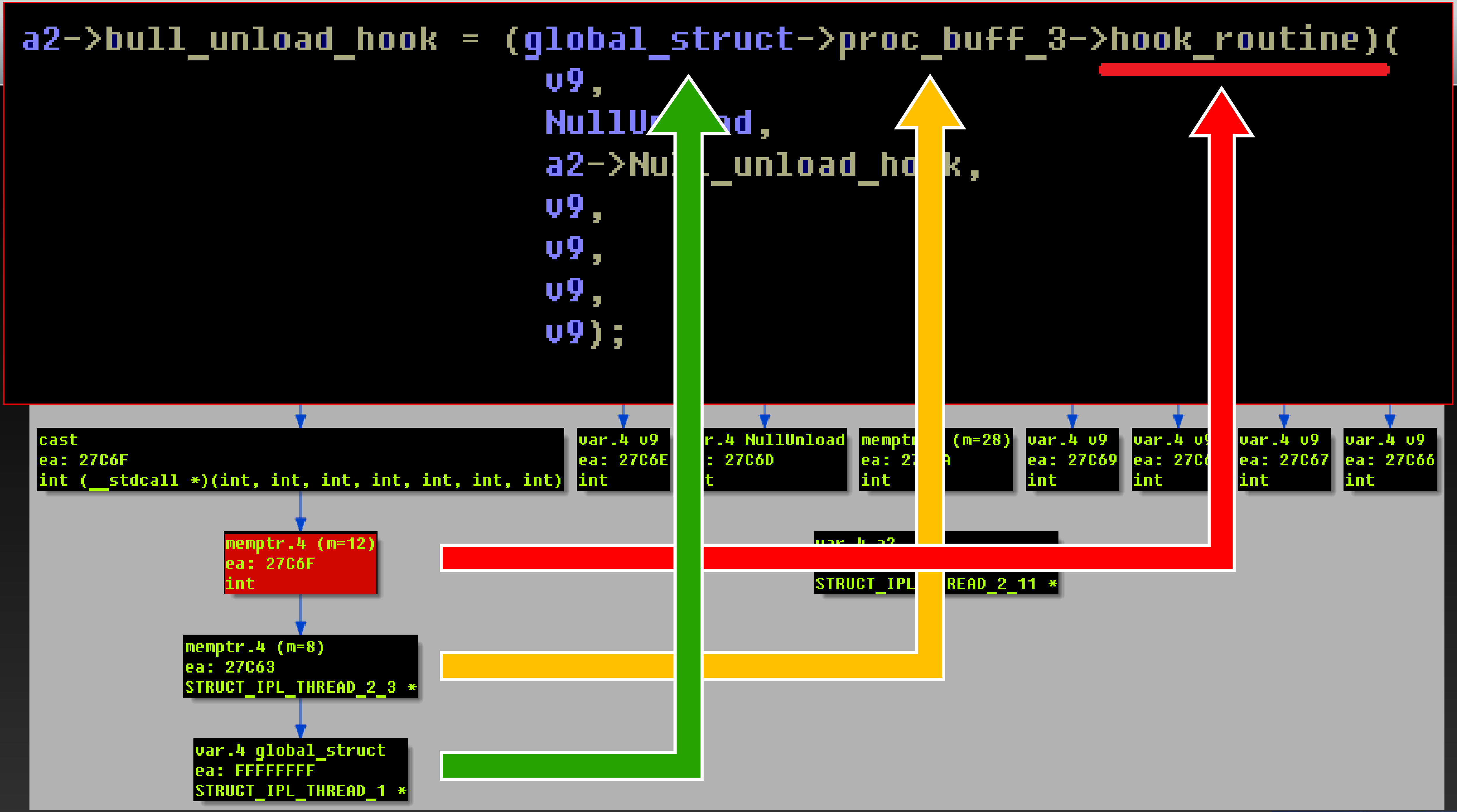
```
#pragma pack(push, 1)
struct STRUCT_IPL_THREAD_2_3
{
  int free_proc_buff_3;
  int _disasm;
  int disasm;
  int hook_routine;
  int unhook;
  int DPC_interlocked_get_dword_9;
  int some_code_part_3;
  int self_buffer_3;
  int field8;
  int field9;
};
#pragma pack(pop)
```

HexRaysCodeXplorer: Virtual Methods

- Hex-Rays decompiler plugin is used to navigate through the virtual methods



```
a2->bull_unload_hook = (global_struct->proc_buff_3->hook_routine)(
                         v9,
                         NullUnload,
                         a2->Null_unload_hook,
                         v9,
                         v9,
                         v9,
                         v9);
```



```
a2->bull_unload_hook = (global_struct->proc_buff_3->hook_routine)(
                        v9,
                        NullUnload,
                        a2->Null_unload_hook,
                        v9,
                        v9,
                        v9,
                        v9);
```

```
cast
ea: 27C6F
int (__stdcall *)(int, int, int, int, int, int, int)

var.4 v9
ea: 27C6E
int

var.4 NullUnload
ea: 27C6D
int

memptr.4 (m=28)
ea: 27
int

var.4 v9
ea: 27C69
int

var.4 v9
ea: 27C6
int

var.4 v9
ea: 27C67
int

var.4 v9
ea: 27C66
int

memptr.4 (m=12)
ea: 27C6F
int

var.4 a2

STRUCT_IPL_THREAD_2_11 *

memptr.4 (m=8)
ea: 27C63
STRUCT_IPL_THREAD_2_3 *

var.4 global_struct
ea: FFFFFFFF
STRUCT_IPL_THREAD_1 *
```

HexRaysCodeXplorer: Object Type REconstruction

- **Hex-Rays's *ctree* structure may be used to partially reconstruct object type based on its initialization routine (constructor)**

- **Input:**
  - ✓ **pointer to the object instance**
  - ✓ **object initialization routine entry point**

- **Output:**
  - ✓ **C structure-like object representation**

HexRaysCodeXplorer: Object Type REconstruction

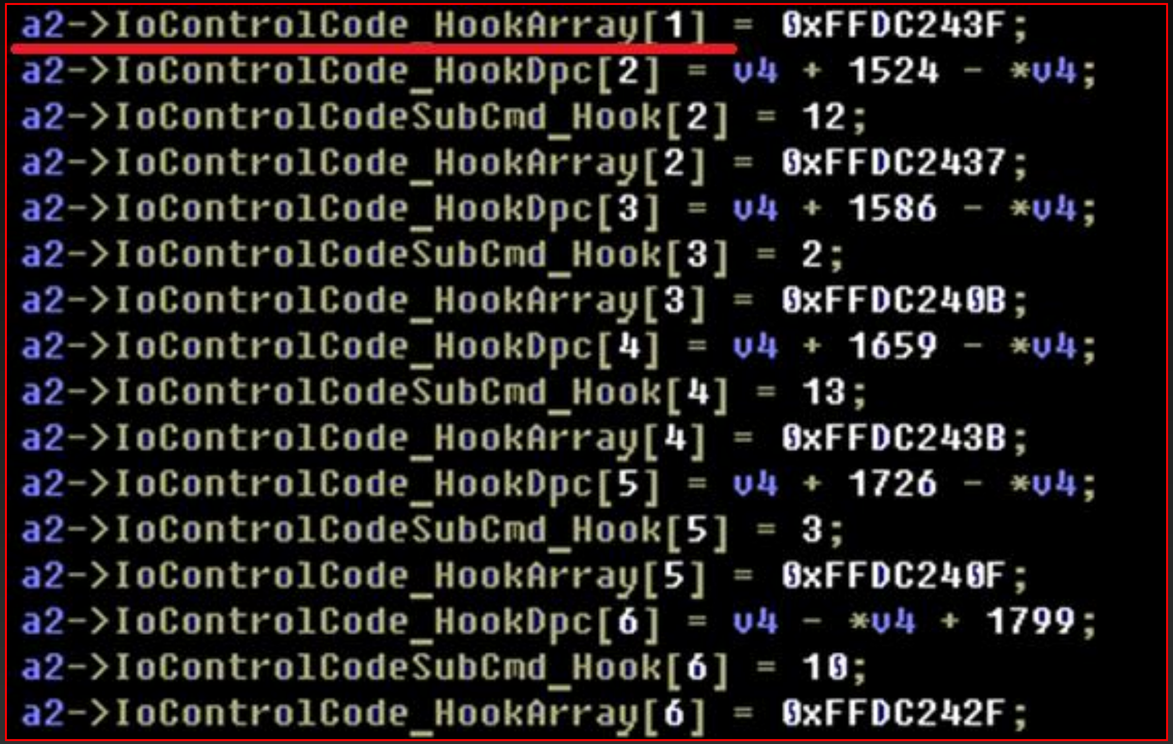
- **Hex-Rays's *ctree* structure may be used to partially reconstruct object type based on its initialization routine (constructor)**

- **Input:**
  - ✓ **pointer to the object instance**
  - ✓ **object initialization routine entry point**

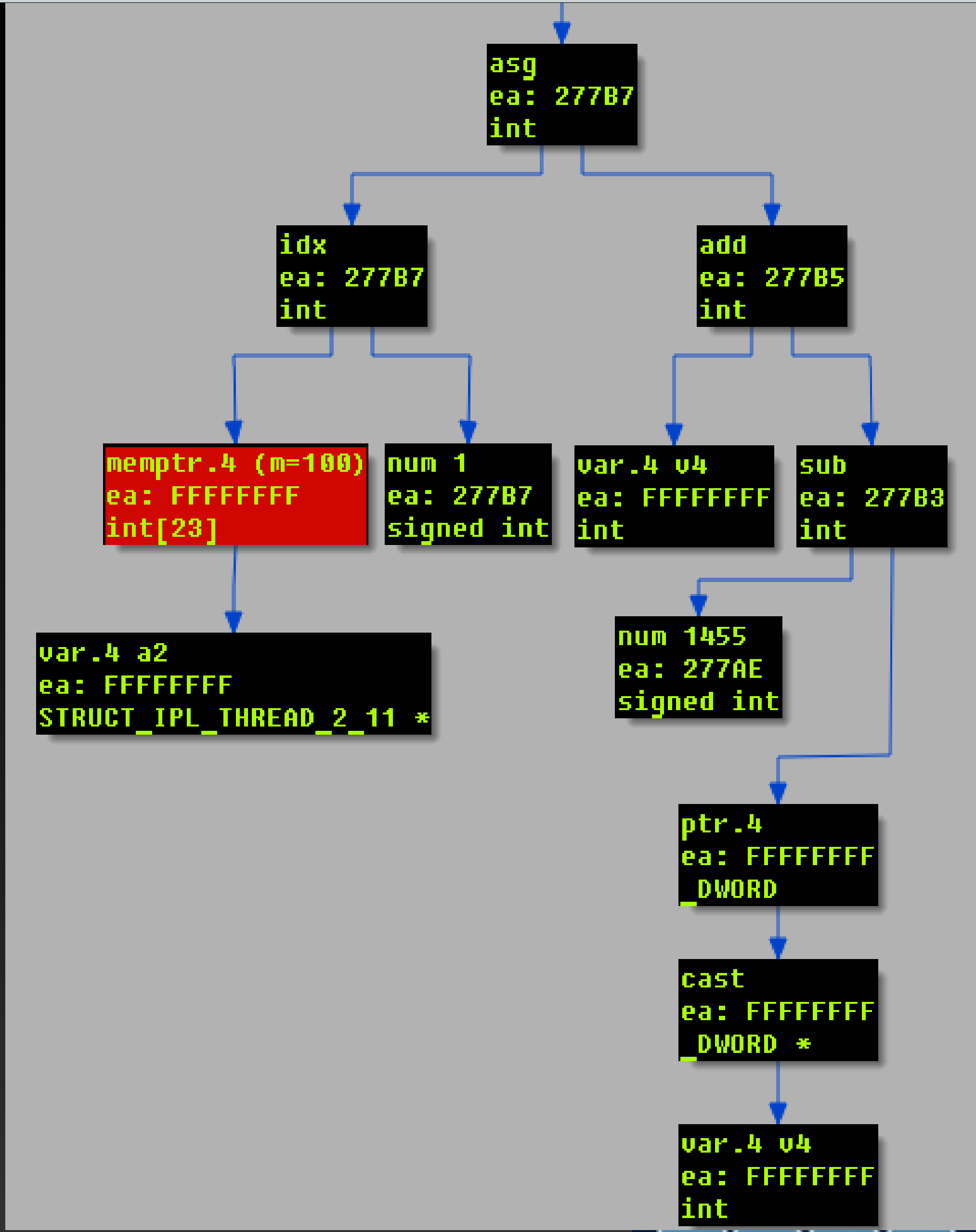
- **Output:**
  - ✓ **C structure-like object representation**

HexRaysCodeXplorer: Object Type REconstruction

- citem_t objects to monitor:
  - ✓ *memptr*
  - ✓ *call (LOBYTE, etc.)*
  - ✓ *idx*
  - ✓ *memref*



```
a2->IoControlCode_HookArray[1] = 0xFFDC243F;
a2->IoControlCode_HookDpc[2] = v4 + 1524 - *v4;
a2->IoControlCodeSubCmd_Hook[2] = 12;
a2->IoControlCode_HookArray[2] = 0xFFDC2437;
a2->IoControlCode_HookDpc[3] = v4 + 1586 - *v4;
a2->IoControlCodeSubCmd_Hook[3] = 2;
a2->IoControlCode_HookArray[3] = 0xFFDC240B;
a2->IoControlCode_HookDpc[4] = v4 + 1659 - *v4;
a2->IoControlCodeSubCmd_Hook[4] = 13;
a2->IoControlCode_HookArray[4] = 0xFFDC243B;
a2->IoControlCode_HookDpc[5] = v4 + 1726 - *v4;
a2->IoControlCodeSubCmd_Hook[5] = 3;
a2->IoControlCode_HookArray[5] = 0xFFDC240F;
a2->IoControlCode_HookDpc[6] = v4 - *v4 + 1799;
a2->IoControlCodeSubCmd_Hook[6] = 10;
a2->IoControlCode_HookArray[6] = 0xFFDC242F;
```



asg
ea: 277B7
int

idx
ea: 277B7
int

add
ea: 277B5
int

memptr.4 (m=100)
ea: FFFFFFFF
int[23]

num 1
ea: 277B7
signed int

var.4 v4
ea: FFFFFFFF
int

sub
ea: 277B3
int

var.4 a2
ea: FFFFFFFF
STRUCT_IPL_THREAD_2_11 *

num 1455
ea: 277AE
signed int

ptr.4
ea: FFFFFFFF
_DWORD

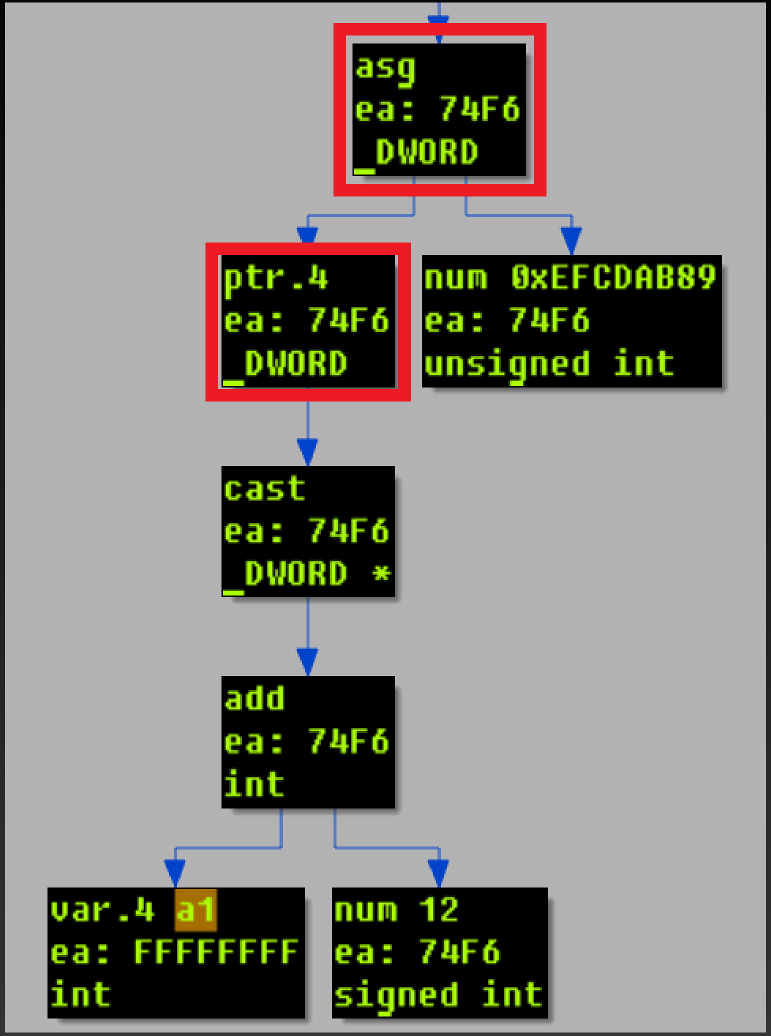
cast
ea: FFFFFFFF
_DWORD *

var.4 v4
ea: FFFFFFFF
int

HexRaysCodeXplorer: Object Type REconstruction

// reference of DWORD at offset 12 in buffer a1

*(DWORD *)(a1 + 12) = 0xEFCDAB89;



asg
ea: 74F6
_DWORD

ptr.4
ea: 74F6
_DWORD

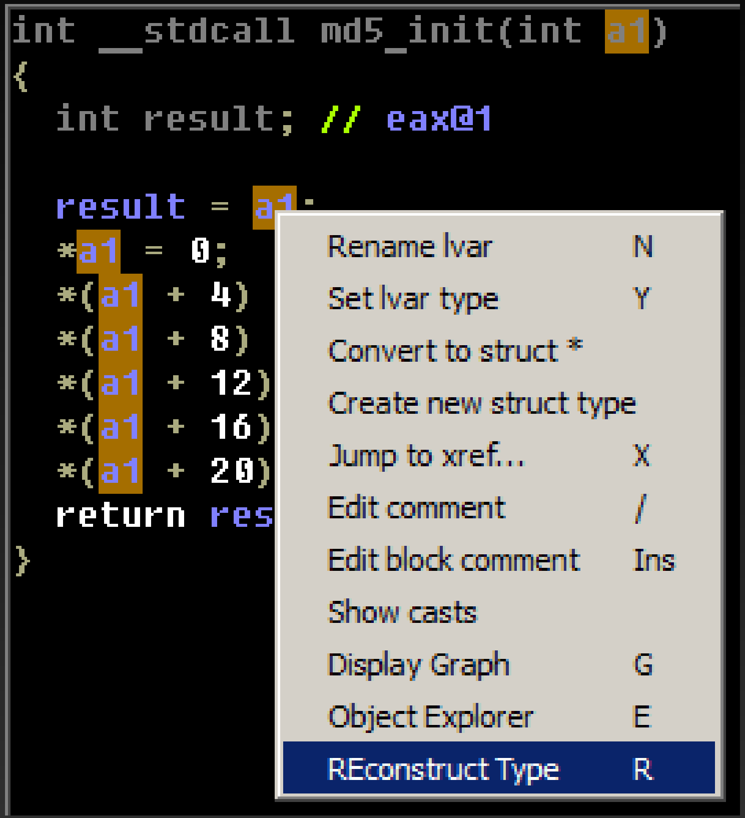
num 0xEFCDAB89
ea: 74F6
unsigned int

cast
ea: 74F6
_DWORD *

add
ea: 74F6
int

var.4 a1
ea: FFFFFFFF
int

num 12
ea: 74F6
signed int

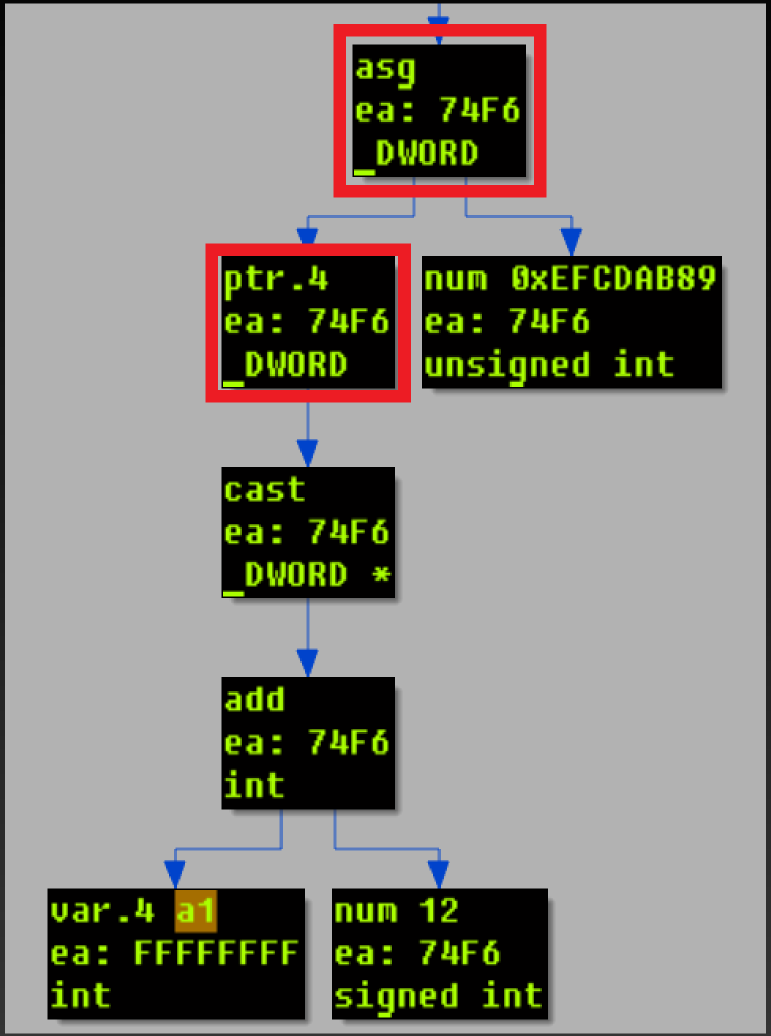


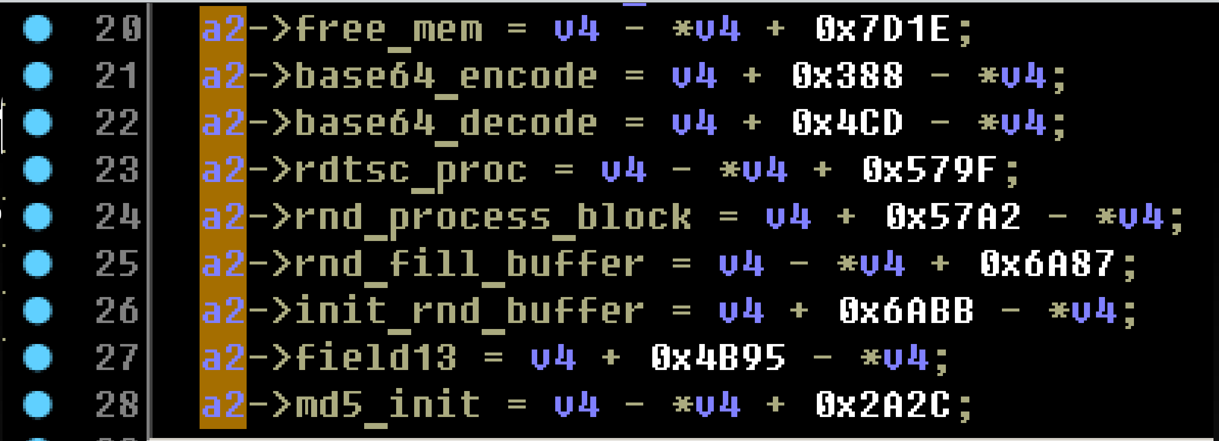
```
int __stdcall md5_init(int a1)
{
  int result; // eax@1

  result = a1;
  *a1 = 0;
  *(a1 + 4)
  *(a1 + 8)
  *(a1 + 12)
  *(a1 + 16)
  *(a1 + 20)
  return res
}
```

HexRaysCodeXplorer: Object Type REconstruction

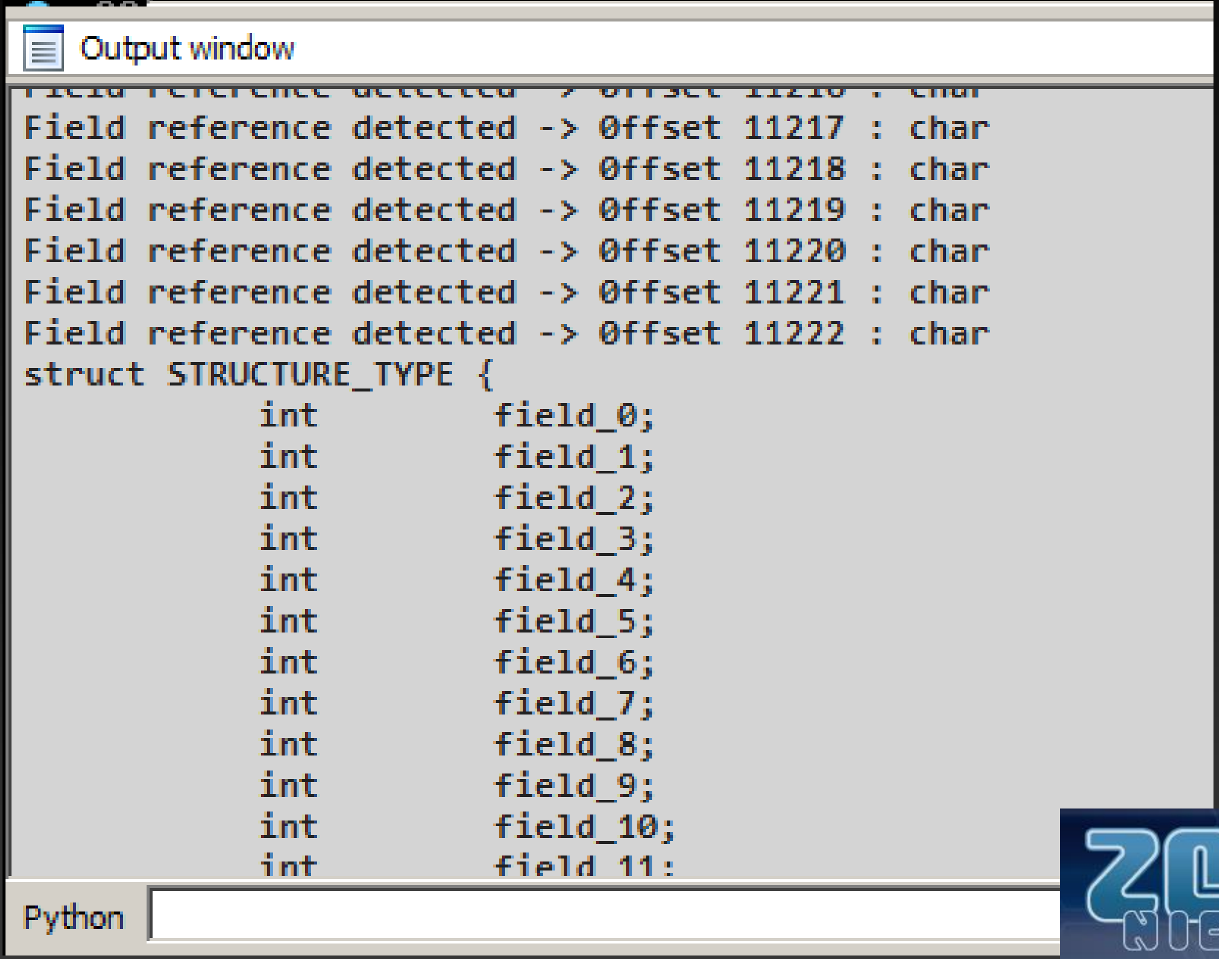
// reference of DWORD at of

*(DWORD *)(a1 + 12) = 0xEF





```
a2->free_mem = v4 - *v4 + 0x7D1E;
a2->base64_encode = v4 + 0x388 - *v4;
a2->base64_decode = v4 + 0x4CD - *v4;
a2->rdtsc_proc = v4 - *v4 + 0x579F;
a2->rnd_process_block = v4 + 0x57A2 - *v4;
a2->rnd_fill_buffer = v4 - *v4 + 0x6A87;
a2->init_rnd_buffer = v4 + 0x6ABB - *v4;
a2->field13 = v4 + 0x4B95 - *v4;
a2->md5_init = v4 - *v4 + 0x2A2C;
```

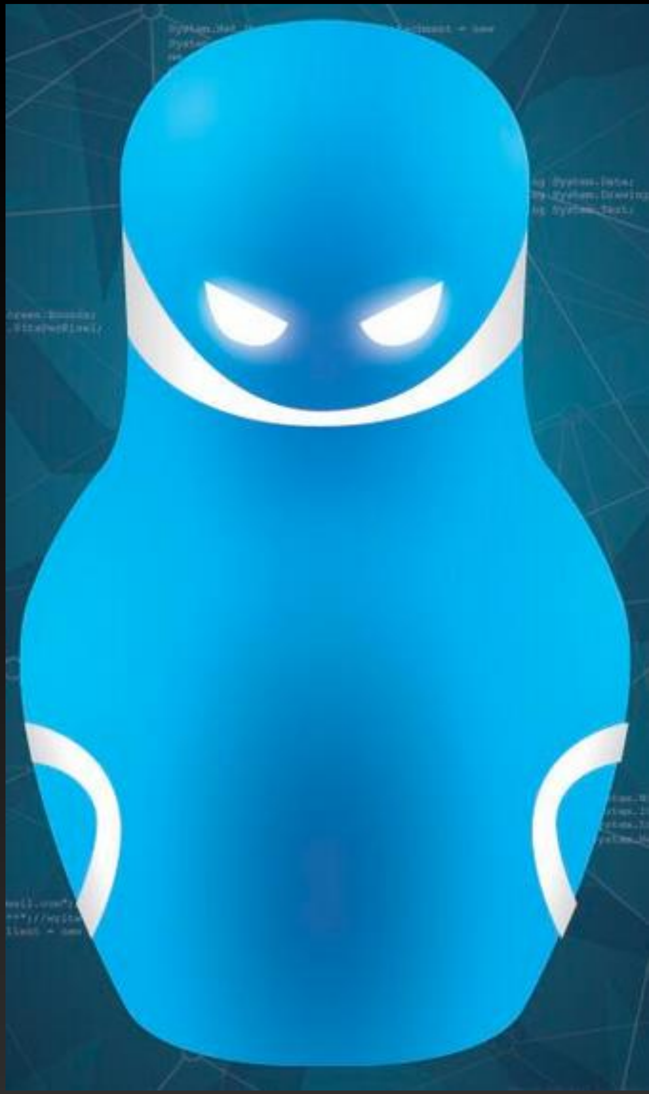


```
Field reference detected -> Offset 11216 : char
Field reference detected -> Offset 11217 : char
Field reference detected -> Offset 11218 : char
Field reference detected -> Offset 11219 : char
Field reference detected -> Offset 11220 : char
Field reference detected -> Offset 11221 : char
Field reference detected -> Offset 11222 : char
struct STRUCTURE_TYPE {
            int         field_0;
            int         field_1;
            int         field_2;
            int         field_3;
            int         field_4;
            int         field_5;
            int         field_6;
            int         field_7;
            int         field_8;
            int         field_9;
            int         field_10;
            int         field_11;
```

HexRaysCodeXplorer 1.1 [ZeroNights Edition]

- **Type Reconstruction:**
  - ✓ reconstruct type into IDA local types
  - ✓ bugfixes =)

- **ObjectExplorer:**
  - ✓ Auto structures for VTBL
  - ✓ Click on VTBL and jump to code
  - ✓ ObjectExplorer hints for VTBL

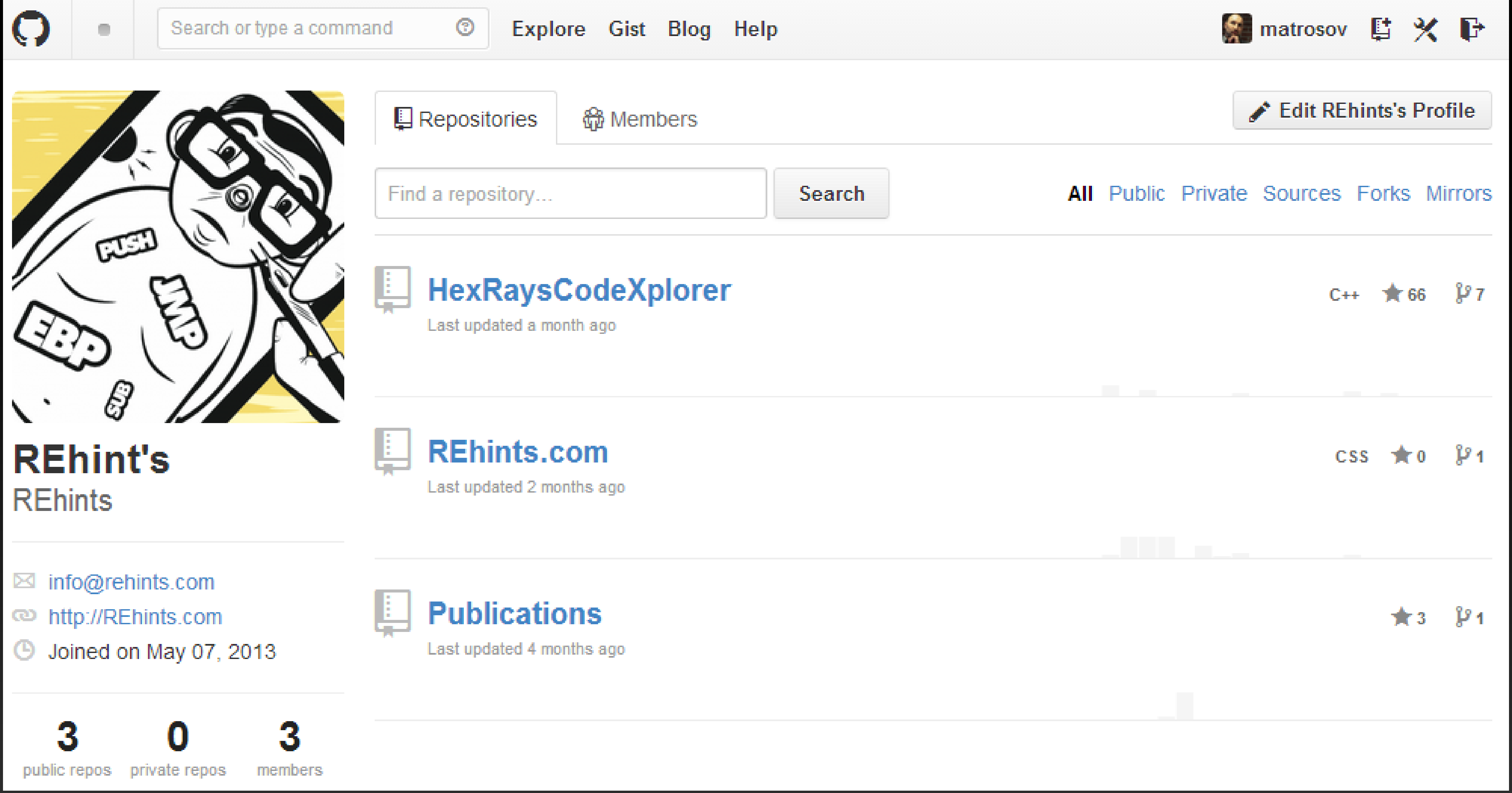


eset

ZERO NIGHTS

NO TIME for DEMO

Come to
Advanced RE workshop

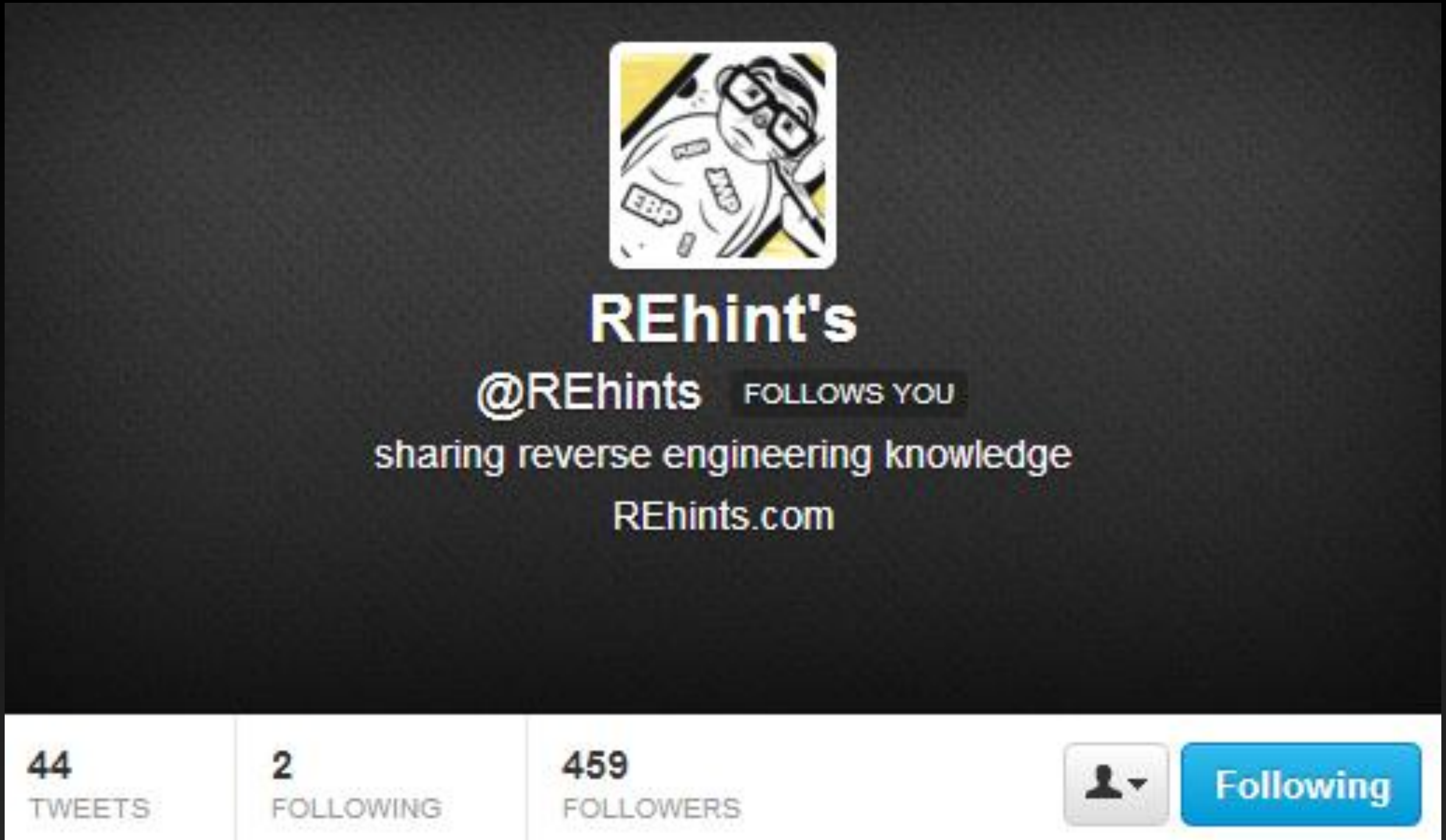


REhint's
REhints

info@rehints.com
http://REhints.com
Joined on May 07, 2013

3	0	3
public repos	private repos	members

Repository	Last updated	Language	Stars	Forks
HexRaysCodeXplorer	Last updated a month ago	C++	66	7
REhints.com	Last updated 2 months ago	CSS	0	1
Publications	Last updated 4 months ago		3	1



REhint's

@REhints FOLLOWS YOU

sharing reverse engineering knowledge

REhints.com

44	2	459
TWEETS	FOLLOWING	FOLLOWERS

Thank you for your attention!

Eugene Rodionov
@vxradius

Aleksandr Matrosov
@matrosov