HexRaysCodeXplorer: make object-oriented RE easier

Aleksandr Matrosov Eugene 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**
  - Vtable reconstruction
  - Associate vtable object with exact object
  - Class hierarchy reconstruction
C++ Code Reconstruction: the truth is out there
An overview of the Flamer Framework

Vector<Consumer>
- Mobile Consumer
- Cmd Consumer
- Lua Consumer
- Media Consumer

Vector<Command Executor>
- DB_Query
- ClanCmd
- FileCollect
- Driller
- GetConfig

Vector<Task>
- IDLER
- CmdExec
- Sniffer
- Munch
- FileFinder

Vector<DelayedTasks>
- Euphoria
- Share Supplier
- LSS Sender
- Frog
- Beetlejuice
An overview of the Flamer Framework
An overview of the Flamer Framework
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
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
At the heart of the decompiler lies *ctree* structure:

```c
/// 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
};
```
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*
- **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 the **ctree** structure:
- **ctree_visitor_t**
- **ctree_parentee_t**

```c
enum _DWORD_fastcall(_DWORD, _BYTE)
```
HexRaysCodeXplorer: Gapz Position Independent Code

```c
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)
```
The IDA’s “Local Types” is used to represent object type
HexRaysCodeXplorer: Virtual Methods

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

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

var.4 v9

memptr.4 (m=12)
ea: 2766F
int

memptr.4 (m=8)
ea: 27663
STRUCT_IPL_THREAD_2_3 *

var.4 global_struct
ea: FFFFFFFF
STRUCT_IPL_THREAD_1 *
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
HexRaysCodeXplorerer: Object Type REconstruction

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

```
a2->IoControlCode_HookArray[1] = 0xFFDC243F;
a2->IoControlCodeSubCmd_Hook[2] = 12;
a2->IoControlCode_HookArray[2] = 0xFFDC2437;
a2->IoControlCodeSubCmd_Hook[3] = 2;
a2->IoControlCode_HookArray[3] = 0xFFDC240B;
a2->IoControlCode_HookArray[4] = 0xFFDC243B;
a2->IoControlCodeSubCmd_Hook[5] = 3;
a2->IoControlCode_HookArray[5] = 0xFFDC240F;
a2->IoControlCode_HookArray[6] = 0xFFDC242F;
```
// reference of DWORD at offset 12 in buffer a1
*(DWORD *)(a1 + 12) = 0xEFCDAB89;
HexRaysCodeXplorerer: Object Type REconstruction

// reference of DWORD at offset 12 in buffer
*(DWORD *)(a1 + 12) = 0xEFCDAB89;
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
NO TIME for DEMO

Come to
Advanced RE workshop
REhint's
REhints

HexRaysCodeXplorer
Last updated a month ago

REhints.com
Last updated 2 months ago

Publications
Last updated 4 months ago

info@rehints.com
http://REhints.com

Joined on May 07, 2013

3 public repos
3 private repos
3 members
REhint's
@REhints  FOLLOWING
sharing reverse engineering knowledge
REhints.com

44 TWEETS  2 FOLLOWING  459 FOLLOWERS

Following
Thank you for your attention!

Eugene Rodionov
@vxradius

Aleksandr Matrosov
@matrosov