MAINTRACK

Integrando Plugins Libres a Ghidra: Ghidra Compatibility Layer

LAUTARO LOMBARDI

¡Hola! Soy Lautaro

- Licenciatura en Cs de la Computación, UNC
- Intern @eclypsium
- Slides @lautalom en Github: Presentations/GCL22

UNC Universidad Nacional de Córdoba

FAMAF Facultad de Matemática, Astronomía, Física y Computación

eclypsium

GCL

Cómo integrar extensiones libres a Ghidra

Agenda

1. Motivación. Objetivo. Restricciones.
2. Arquitectura.
3. main.py
4. Ejemplo: Findcrypt.
5. Ejemplo: Syms2elf.
6. Ejemplo: UEFI REtool.
7. Pros and Cons
8. Trabajo Futuro
9. Q&A

100+ plugins

En IDA

100+ plugins

En Binary Ninja

Ideas similares

Daenerys, Binary Ninja

EKOPARTY

GCL

Objetivo

Ejecución de extensiones de código abierto de plataformas distintas de Ghidra (IDA, Binary Ninja, etc)...en Ghidra.

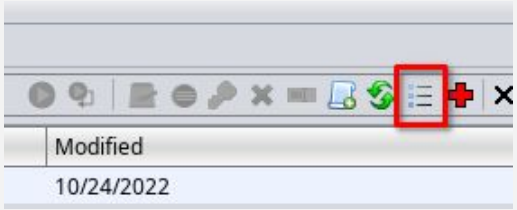
Restricción

No alterar el código fuente.

Procedimiento

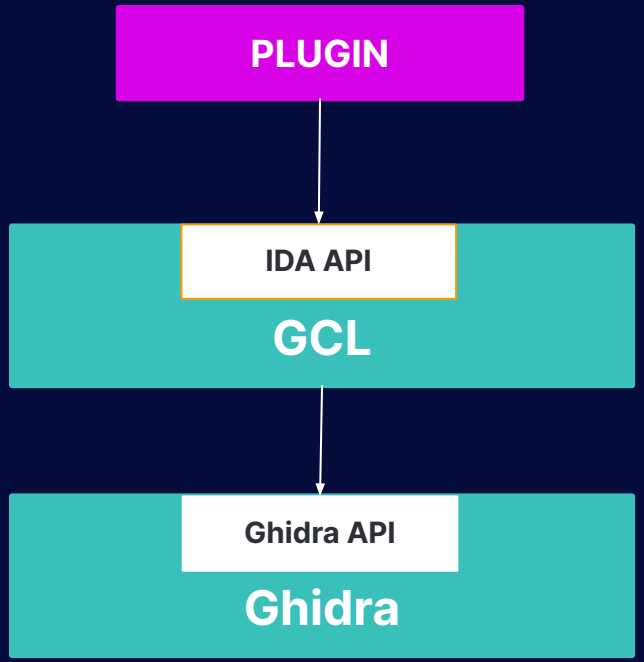
1. Añades el plugin de interés en el Administrador de Scripts de Ghidra.
2. Añades los archivos necesarios de la GCL. (*Solo pueden estar en 1 directorio a la vez).
3. Corres el plugin (como un Script en Ghidra).

1.



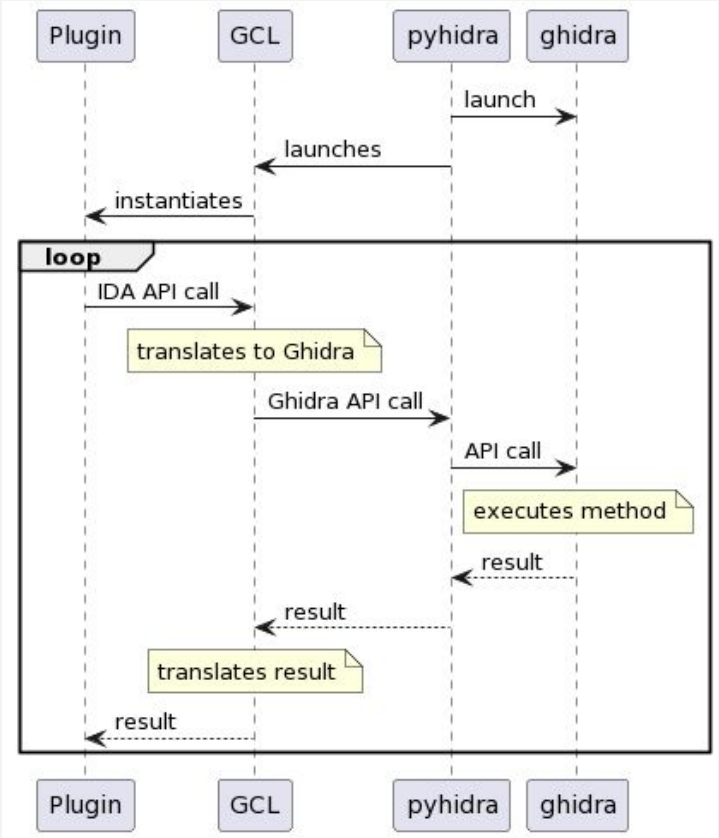
2.

Arquitectura - System Context



EKOPARTY

Arquitectura - Secuencia



EKOPARTY

main.py

```
1.  if __name__ == '__main__':
2.      try:
3.          import importlib.util
4.          import sys
5.          import os
6.          import cp
7.      except:
8.          sys.path.append(os.path.dirname(__file__))
9.          import cp
10.     finally:
11.         cp.currentProgram = currentProgram # export currentProgram
12.         plugin = "someplugin"
13.         path = os.path.dirname(__file__)+'/'+plugin+'.py' # manual import
14.         spec = importlib.util.spec_from_file_location(plugin,path)
15.         module = importlib.util.module_from_spec(spec)
16.         spec.loader.exec_module(module)
17.         module.PLUGIN_ENTRY().run(0) # call entrypoint
```

Findcrypt

Yara rule

```
rule CRC32c_poly_Constant {
meta:
        author = "_pusher_"
        description = "...."
strings:
        $c0 = { 783BF682 }
condition:
        $c0
}
```

```
#include <stdio.h>

const char poly[] = { '\x78','\x3b', '\xf6', '\x82', '\0'};
int main() {
    printf("%s\n", poly);
    return 0;
}
```

Findcrypt Essentials

Módulos necesarios

import **idaapi**
import **idautils**
import **ida_bytes**
import **ida_diskio**
import **idc**
...etc

Mocking

class YaraSearchResultChooser(**idaapi.Choose**):

Tratar de ser tan general como se pueda a la hora de definir clases y componentes.

Entrypoint

```
# register IDA plugin
def PLUGIN_ENTRY():
    return Findcrypt_Plugin_t()
```

```
┌─lautaro@latitude ~/scripts/layer ‹pyhidra›
└─$ pyhidra -g
INFO  Using log config file: jar:file:/home/lautaro/ghidra_10.1.4_PUBLIC/Ghidra/Framework/Generic/lib/Gener
oggingInitialization)
INFO  Using log file: /home/lautaro/.ghidra/.ghidra_10.1.4_PUBLIC/application.log (LoggingInitialization)
INFO  Loading user preferences: /home/lautaro/.ghidra/.ghidra_10.1.4_PUBLIC/preferences (Preferences)
INFO  Class search complete (1046 ms) (ClassSearcher)
INFO  Initializing SSL Context (SSLContextInitializer)
INFO  Initializing Random Number Generator... (SecureRandomFactory)
INFO  Random Number Generator initialization complete: NativePRNGNonBlocking (SecureRandomFactory)
INFO  Trust manager disabled, cacerts have not been set (ApplicationTrustManagerFactory)
INFO  User lautaro started Ghidra. (GhidraRun)
INFO  Opening project: /home/lautaro/programs/holis_ghidra/holis_ghidra (DefaultProject)
INFO  Packed database cache: /tmp/lautaro-Ghidra/packed-db-cache (PackedDatabaseCache)
Could not get bytes
Could not get bytes
>>> start yara search
<<< end yara search
Address: 8196 | Rules file: global | Name: CRC32c_poly_Constant_2004 | String: $c0 | Value: b'x;\xf6\x82' |
```

EKOPARTY

Algunas traducciones

```
def _get_memory(self):
    result = bytearray()
    segment_starts = [ea for ea in idautils.Segments()]
    if condition:
     result += ida_bytes.get_bytes(start, end - start)
```

idautils.py

```
import cp
def Segments():
  """returns a list of segment starting offsets"""
  blocks = cp.currentProgram.getMemory().getBlocks()
  minAddress = currentProgram.minAddress.getOffset()
  ans = [i.getStart()-minAddress for i in blocks if i.isRead()]
    return ans
```

EKOPARTY

Algunas traducciones

```
def _get_memory(self):
  result = bytearray()
  segment_starts = [ea for ea in idautils.Segments()]
  if condition:
    result += ida_bytes.get_bytes(start, end - start)
```

Ida_bytes.py #pseudocode

```
def get_bytes(start, length):
 """Returns length bytes from start of a
segment"""
 res = b""
 minAddress = cp.currentProgram.minAddress
 fcp = FlatProgramAPI(cp.currentProgram)
 start += minAddress
 try:
  res = fcp.getBytes(toAddr(start),length + 1)
  res = list(res)
  res = [i if i >= 0 else (256 + i) for i in
res]
 except:
  print("Could not get bytes")
 return bytes(res)
```

Syms2elf: Essentials

Imports

import idaapi

import idc

import ida_nalt

import ida_kernwin

import traceback? error 404

except:

    log(traceback.format_exc())

```
lautaro@latitude ~/scripts/syms2elf ‹329c2ce●›
$ python
Python 3.10.7 (main, Sep  6 2022, 21:22:27) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import pyhidra
>>> pyhidra.run_script(r"/home/lautaro/programs/holis",r"/home/lautaro/scripts/syms2elf/main.py")
[syms2elf] Exporting symbols to ELF...
[syms2elf] 0x00101000 - 0x0000001b - _init - 1/99 - 18
[syms2elf] 0x00101040 - 0x00000026 - _start - 7/99 - 18
[syms2elf] 0x00101070 - 0x00000021 - deregister_tm_clones - 14/99 - 18
[syms2elf] 0x001010a0 - 0x00000032 - register_tm_clones - 16/99 - 18
[syms2elf] 0x001010e0 - 0x00000036 - __do_global_dtors_aux - 54/99 - 18
[syms2elf] 0x00101130 - 0x00000009 - frame_dummy - 76/99 - 18
[syms2elf] 0x00101139 - 0x0000001a - main - 88/99 - 18
[syms2elf] 0x00101154 - 0x0000000d - _fini - 93/99 - 18
[syms2elf] ELF saved to: /home/lautaro/scripts/syms2elf/symbols.elf
>>>
```

EKOPARTY

Algunas traducciones

Source

```
def ida_fcn_filter(func_ea):
if idc.get_segm_name(func_ea) not in ("extern",".plt"):
        return True
    return False

def get_ida_symbols():
 symbols = []
 for f in filter(ida_fcn_filter, Functions()):
     func     = get_func(f)
     seg_name = idc.get_segm_name(f)
     fn_name  = idc.get_func_name(f)
     symbols.append(Symbol(fn_name, STB_GLOBAL_FUNC,
       int(func.start_ea), int(func.size()),seg_name))
```

Idc.py #pseudocode

```
def get_segm_name(ea):
 res = ""
 minAddress = currentProgram.minAddress
 fcp = FlatProgramAPI(currentProgram)
 res = getBlock(toAddr(ea+minAddress)).getName()
  if res == "EXTERNAL":
     res = "extern"
  return res
```

EKOPARTY

Algunas traducciones

Source

```
def ida_fcn_filter(func_ea):
if idc.get_segm_name(func_ea) not in ("extern",".plt"):
        return True
    return False

def get_ida_symbols():
 symbols = []
 for f in filter(ida_fcn_filter, Functions()):
     func     = get_func(f)
     seg_name = idc.get_segm_name(f)
     fn_name  = idc.get_func_name(f)
     symbols.append(Symbol(fn_name, STB_GLOBAL_FUNC,
       int(func.start_ea), int(func.size()),seg_name))
```

idautils.py #pseudocode

```
class FunWrapper:
 def __init__(self, f):
  minAddress = cp.minAddress
  lng = cp.getListing()
  fun = lng.getFunctionAt(toAddr(minAddress+f))
  self.start_ea = fun.getBody().getMinAddress()
  self.fsize = fun.getBody().getFirstRange().length

 def size(self):
  return self.fsize

def get_func(f):
    func = FunWrapper(f)
    return func
```

EKOPARTY

UEFI RETool (WIP)

```
┌─lautaro@latitude ~/scripts/uefi_retool/ida_plugin ‹master›
└─$ ls
README.md  uefi_analyser  uefi_analyser.py
┌─lautaro@latitude ~/scripts/uefi_retool/ida_plugin ‹master›
└─$ ls uefi_analyser
analyser.py      dep_graph.py  __init__.py  log_pp_guids.py   __pycache__  ui.py
dep_browser.py   guids         log_all.py   prot_explorer.py  tables.py    utils.py
```

Status: agregando estructuras UEFI en el DataType Manager de Ghidra

```
└─$ pyhidra -g
INFO  Using log config file: jar:file
oggingInitialization)
INFO  Using log file: /home/lautaro/.
INFO  Loading user preferences: /home
INFO  Class search complete (1094 ms)
INFO  Initializing SSL Context (SSLCo
INFO  Initializing Random Number Gene
INFO  Random Number Generator initial
INFO  Trust manager disabled, cacerts
INFO  User lautaro started Ghidra. (G
INFO  Opening project: /home/lautaro/
INFO  Closed project: holis_ghidra (F
INFO  Opening project: /home/lautaro/
INFO  Opened project: spidump (FileAc
INFO  Packed database cache: /tmp/lau
Boot services:
 * list is empty
Protocols:
 * list is empty
Comments:
 * list is empty
Names:
 * list is empty
Types:
 * list is empty
```

Algunas traducciones

```
if ((idc.print_insn_mnem(ea) == "mov")
   and (idc.get_operand_value(ea, 0) == REG_RAX)
   and (idc.get_operand_value(ea, 1) == BS_OFFSET)
     )
```

idc.py

```
def print_insn_mnem(ea):
 fcp = FlatProgramAPI(cp.currentProgram)
 minAddress = cp.currentProgram.minAddress.getOffset()
 listing = cp.currentProgram.getListing()
 codeUnit = listing.getCodeUnitAt(fcp.toAddr(minAddress+ea))
   if codeUnit is not None:
     return str(codeUnit.getMnemonicString().lower())
   else return “”
```

Pros and Cons

Rapidez de desarrollo

La integración de un nuevo plugin toma cada vez menos esfuerzo (se aprovecha la API ya traducida).

En general es más simple un script que un plugin (aunque también más limitado).

Reutilizar

Impulsar la interoperabilidad de tecnologías de ingeniería reversa.

Nuevas plataformas

Hay lugar para explorar con más plataformas de ingeniería reversa como Binary Ninja.

Versionado de APIs

GCL debe adecuarse al versionado de Ghidra y de las plataformas de RE que integre.

Desde cero

Dependiendo del tamaño del plugin y sus funciones, puede ser parecido incluirlo en Ghidra a reescribirlo.

Internals...muy internals

Algunas funcionalidades son específicas de algún procesador, o de un módulo. Algunas funciones pueden no ser biyectivas y devolver valores ad hoc...

EKOPARTY

Trabajo Futuro

¿Extender la GCL?

Si te interesa el proyecto puedes ayudarme extendiendo nuevas traducciones de las APIs....

Mejorar la arquitectura

¿Se te ocurren formas de mejorar la compatibilidad entre versiones diferentes de APIs de los SDK?

¿Explorar plugins con otros lenguajes?

Si un plugin está un lenguaje distinto a Python, ¿hay forma para incluirlo en GCL?

Q & A

¿Preguntas?

EKOPARTY

THANKS! GRACIAS!

Contacto

linkedin/lautaro-lombardi

lautarolombardi19@gmail.com