Hands-On Ghidra

A Tutorial about the Software Reverse Engineering Framework

Roman Rohleder
Thales Group

THALES

Ghidra?

“Gee-druh. The G sounds like the G in goto, great, good, graph and GitHub. The emphasis goes on the first syllable.”

Frequently asked questions

Introduction - Ghidra

- Software Reverse Engineering Framework
- Developed by the National Security Agency
- Public release March 5th 2019
- Open Source, Apache v2 license
- Written in Java*, runs on Linux, Windows & Mac
- **Free**

Overview

- Features
- Extension & Automation
- p-code & SLEIGH format
- Comparison with IDA Pro

Features

- Supports many architectures
- Highly customizable
- Decompiler
- Collaboration/Ghidra Server
- Emulator*
- Thoroughly documented
- Parse C Source & Structure editor
- Built-in Assembler
- Control Flow Graph & Call Graph visualization
- “Version Tracking”

Features – Supported Architectures

6502, 68000, 6805, 80251, 80390, 8048, 8051, 8085, ARM/AArch64, AVR8/32, CR16C, Dalvik, JVM, dsPIC30F/33E/33F, HC05/08/S08/S12, MCS96, MIPS, PA-RISC, PIC-12/16/17/18/24, PowerPC, Sparc, SuperH/H4, TI MSP430/430X, TriCore, x86/64, Z180, Z80

Features – Supported Architectures

6502, 68000, 6805, 80251, 80390, 8048, 8051, 8085, **ARM/AArch64**, AVR8/32, CR16C, **Dalvik**, **JVM**, dsPIC30F/33E/33F, HC05/08/S08/S12, MCS96, **MIPS**, PA-RISC, PIC-12/16/17/18/24, **PowerPC**, **Sparc**, SuperH/H4, TI MSP430/430X, TriCore, **x86/64**, Z180, Z80

Features - Customization

- Modify window layout (add/remove views, re-organize, …)
- Despite (re-)organization of views
  All in sync with current selection
- Modify Hotkeys
- Change fonts, fore-/background colors
- Load & Organize Plug-Ins within the GUI

Features – Decompiler

- **THE** most anticipated feature
- Works for all aforementioned architectures
- Fairly clean
- Different Data Flows highlightable (def-use chain, forward/ backward slice)
- Potential decompilation errors are tagged with special variable names/prefixes (in_, in_stack_, extraout_, unaff_)

Features – Collaboration

- Ghidra client & server
- Share and work on projects with multiple users
- Read/Write/Admin access per user configurable
- Merge conflicts can be resolved with a given tool
- Authentication: Username/Password, Active Directory (Kerberos), PKI, JAAS, SSH preshared key for headless
- Not interactive
- No branches

Features – Emulator*

- Has API for emulation
- Ability to set breakpoints for the emulation
- Sample scripts provided
- However no nice “clicky” interface for out-of-the-box usage*

*yet… (supposedly to be released with an Integrated debugger some time)

Features – Header parser & Struct editor

- Visual struct editor
- Struct/Data previews
- Accumulated data types exportable/importable (Ghidra Data Type Archives .gdt)
- You can provide custom header files to add function signatures, structs, …
- Export all said types to a header file

Features – Built-In Assembler

- Auto-completion (use upper case)
- Immediately alters analysis/decompilation
- Changes only in Ghidra, not file on disk
- Changes can be exported back to file*
- Different stability/coverage ratings per Architecture

Features – Built-In Assembler

- Poor: disPIC30F
- Bronze: AVR32
- Gold: x86-64
- Platinum: x86, ARM/Thumb 32, AArch64, PowerPC, SPARC, MIPS, PA-RISC, AVR8, SuperH-4, 68000, TI MSP430X

Features – Documentation

- Javadoc for Java API available
- Context-sensitive & well described Help pages
  (hover mouse over item in question & press F1)
- GhidraClass: Slide-sets & exercises covering all aspects of Ghidra usage and extension (Beginner, Intermediate, Advanced)
- 245 example scripts (Java & Python),
  showcasing how to use the API
- Instruction reference* & Instruction encoding

*requires prior download of reference manuals to right location

Features – CFGs & Call graphs

- Interactive Control Flow Graphs and Call graphs*
- Both sync with code selection changes
- Flows to/from blocks or loops are highlightable
- Call graphs also representable as Call Tree (quick overview w/o needing much space)

*seem a bit sluggish,
especially on obfuscated code

Extension & Automation

- Java scripts
- Python scripts & Interpreter
- Customized “tools”
- Headless mode

Extension & Automation - Java Scripts

- Integration with Eclipse
  → Auto-completion
  → Debuggability
- Ghidra Program API vs. Script API
- GhidraDev Eclipse plugin
- Can run other java or python scripts from within a script

Extension & Automation - Java Scripts

```
// Description goes here
// and continues here
//@author Author
//@category MyScripts
//@keybinding alt f
//@menupath MyScripts.Fix Disassembly1
//@toolbar logo.png
```

Extension & Automation - Java Scripts

```
// Description goes here
// and continues here
//@author Author
//@category MyScripts
//@keybinding alt f
//@menupath MyScripts.Fix Disassembly1
//@toolbar logo.png
```

Extension & Automation - Java Scripts

```
// Description goes here
// and continues here
//@author Author
//@category MyScripts
//@keybinding alt f
//@menupath MyScripts.Fix Disassembly1
//@toolbar logo.png
```

Extension & Automation - Java Scripts

```
// Description goes here
// and continues here
//@author Author
//@category MyScripts
//@keybinding alt f
//@menupath MyScripts.Fix Disassembly1
//@toolbar logo.png
```

Extension & Automation - Java Scripts

```
// Description goes here
// and continues here
//@author Author
//@category MyScripts
//@keybinding alt f
//@menupath MyScripts.Fix Disassembly1
//@toolbar logo.png
```

Extension & Automation - Java Scripts

```
// Description goes here
// and continues here
//@author Author
//@category MyScripts
//@keybinding alt f
//@menupath MyScripts.Fix Disassembly1
//@toolbar logo.png
```

Extension & Automation - Java Scripts

```
// Description goes here
// and continues here
//@author Author
//@category MyScripts
//@keybinding alt f
//@menupath MyScripts.Fix Disassembly1
//@toolbar logo.png
```

Extension & Automation - Java Scripts

```
// Description goes here
// and continues here
//@author Author
//@category MyScripts
//@keybinding alt f
//@menupath MyScripts.Fix Disassembly1
//@toolbar logo.png
```

Extension & Automation - Python

- Run via Jython
- Tied to Python 2.7.1
- Integrated interpreter
- Auto-completion
- help(COMMAND) → prints corresponding javadoc
- GhidraDev Eclipse plugin + PyDev plugin
- Can run other python or java scripts from within a script

Extension & Automation – Custom “tools”

- Save window layout, key bindings, colors, loaded plugins, etc. as custom “tools”
- Useful to have several tools for different tasks (Not have everything clobbered into one & always adjust the windows etc.)

Extension & Automation – Headless mode

- Run custom scripts before/after analysis or w/o analysis
- Turn On/Off certain analysis passes
- Java scripts/Python scripts – both work
- Run on single file, folders, wildcarded files
- Import into existing projects, keep/delete newly created projects
- Can interact with shared repositories
  (Computation happens locally though)
- Make address selections or pass values to follow-up scripts

p-code & SLEIGH format

- p-code: Ghidras intermediate representation (IR)

  Yes yes… another IR...

- SLEIGH: file format describing

Binary → Assembly → p-code snippet

+ information about registers and adress space

p-code & SLEIGH format

- Register Transfer Language
- “raw p-code” & “Additional p-code”
- No side-effects
- Unlimited temporary registers
- Address space, Varnode & p-code operations
- Pseudo p-code

COPY
LOAD
STORE
BRANCH
CBRANCH
BRANCHIND
CALL
CALLIND
USERDEFINED
RETURN
PIECE
SUBPIECE
INT_EQUAL
INT_NOTEQUAL
INT_LESS
INT_SLESS
INT_LESSEQUAL
INT_SLESSEQUAL
INT_ZEXT
INT_SEXT

INT_ADD
INT_SUB
INT_CARRY
INT_SCARRY
INT_SBORROW
INT_2COMP
INT_NEGATE
INT_XOR
INT_AND
INT_OR
INT_LEFT
INT_RIGHT
INT_SRIGHT
INT_MULT
INT_DIV
INT_REM
INT_SDIV
INT_SREM
BOOL_NEGATE
BOOL_XOR
BOOL_AND

Pseudo p-code

BOOL_OR
FLOAT_EQUAL
FLOAT_NOTEQUAL
FLOAT_LESS
FLOAT_LESSEQUAL
FLOAT_ADD
FLOAT_SUB
FLOAT_MULT
FLOAT_DIV
FLOAT_NEG
FLOAT_ABS
FLOAT_SQRT
FLOAT_CEIL
FLOAT_FLOOR
FLOAT_ROUND
FLOAT_NAN
INT2FLOAT
FLOAT2FLOAT
TRUNC
CPOOLREF
NEW

Additional p-code operations

- MULTIEQUAL
- INDIRECT
- PTRADD
- PTRSUB
- CAST

p-code & SLEIGH format

- SLEIGH format can have file inclusions, macros and other preprocessing
- Defines endianness, alignment, wordsize, access (r/w) and other properties of address spaces
- Complex but generic format further describing the disassembly process

Comparison with IDA Pro

- Architecture support:
  → More disassemblers in IDA
  → More decompilers in Ghidra (All previously mentioned architectures)
- Features:
  → Integrated debugger for all major platforms in IDA
  → Integrated collaboration in Ghidra
- Extensibility:
  → Broad community and many plugins for IDA
  → Thorough documentation and many examples in Ghidra

Comparison with IDA Pro

- Performance
- Documentation
- Decompilation: Comparable, slight differences
- Stability: both similarly good/bad
- “Look & Feel”
- The little things
- **Price: free vs. 52959$**

Future?

Official:

- Debugger
- (Emulator)

Community:

- More plugins to follow...
- P-code → LLVM IR anyone?

Conclusion

- Great all-in-one framework
- Easy to use and extend
- **Free**

Thank you for your attention!

Questions?

Contact:
Roman Rohleder
roman.rohleder@thalesgroup.com

Resources

Ghidra Project page: https://ghidra-sre.org/

Github: https://github.com/NationalSecurityAgency/ghidra