How To Auto-Feed Your Dragon: Hatchlings First Ghidra Script

About This Course

- Two-day workshop
- Brief overview and tour of Ghidra itself
- Overview of the script manager
- Writing and running scripts
- Headless Analysis
- Discussion of Java and Python/Jython
- Flat API overview and highlights
- Scripting practice
- Not a comprehensive ‘everything you could ever learn’

About the Instructor

Morgan Whitlow

- Currently:
  - Embedded firmware reverse engineer
  - Hardware enthusiast
  - Sarcastic smartass
- Education:
  - Master of Science in Applied Computer Science
  - B.S. Biology, B.A. Psychology
- Formerly:
  - SOC analyst
  - Lockpicking instructor
  - Nanomaterials researcher
  - Various other stuff

Some (Questionable) Definitions

- Static Analysis → Take file, take file apart.
  - Disassembler →Static. Binary in, assembly out
  - Decompiler →Static. Binary in, higher level code out (usually C)
- Dynamic Analysis → Take file, run file, poke it with a stick and see what happens.
  - Debugger → Dynamic. Take file, run file, attach to it’s process and play with it. Can be risky if you’re not careful.
- Each has pros and cons. Good to be familiar with both

What is Ghidra

- “Software reverse engineering (SRE) framework”
- Free and open-source
- Cross-platform, works on Linux, Windows and Mac
- Can analyze multiple different architectures
- Dissassembler
- Decompiler
- No debugger yet, but supposedly soon
- Java based
- Supports scripting in both Java and Python

Setup and Installation

Setup: Java JDK - Acquire

- Version you want:
  - Java 11 64-bit Runtime and Development Kit (JDK)

```
student@ubuntu:~$ java --version
openjdk 11.0.4 2019-07-16
OpenJDK Runtime Environment (build 11.0.4+11-post-Ubuntu-1ubuntu218.04.3)
OpenJDK 64-Bit Server VM (build 11.0.4+11-post-Ubuntu-1ubuntu218.04.3, mixed mode, sharing)
```

- Find and install:
  - Linux:
    - sudo apt-get install openjdk-11-jdk
  - Windows/Mac:
    - https://adoptopenjdk.net/releases.html?variant=openjdk11&jvmVariant=hotspot

Setup: Java JDK - Path

- Linux users who installed via apt-get and anyone who used a graphical installer can skip this, it should be done for you automatically
  - Sometimes has issues, check in CLI!

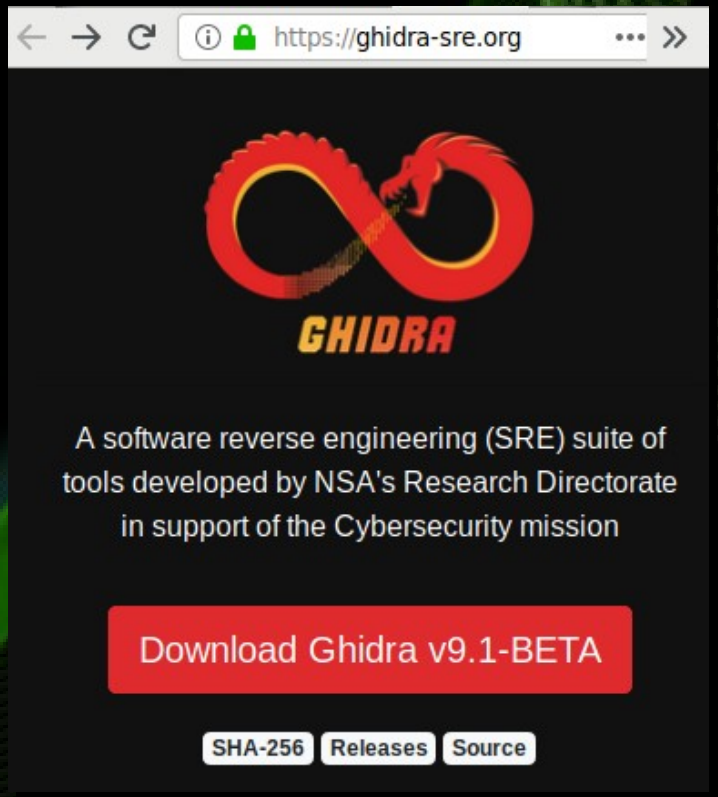
- Otherwise, add the bin folder from the Java directory to your system path. The Ghidra Installation Guide provides detailed instructions:
  - https://ghidra-sre.org/InstallationGuide.html#JavaNotes

Setup: Ghidra - Acquire

- Two places:
  - Ghidra-sre.org
    - https://ghidra-sre.org/

Releases

Version	Link	SHA-256	Notes	Date
9.1	ghidra_9.1-BETA_DEV_20190923.zip	3d61de711b7ea18bdee3ed94c31429e4946603b3e7d082cca5e949bbd651f051	Notes	2019-09-23
9.0.4	ghidra_9.0.4_PUBLIC_20190516.zip	a50d0cd475d9377332811eeae66e94bdc9e7d88e58477c527e9c6b78caec18bf	Notes	2019-05-16
9.0.2	ghidra_9.0.2_PUBLIC_20190403.zip	10ffd65c266e9f5b631c8ed96786c41ef30e2de939c3c42770573bb3548f8e9f	Notes	2019-04-03
9.0.1	ghidra_9.0.1_PUBLIC_20190325.zip	58ffa488e6dc57e2c023670c1dfac0469bdb6f4e7da98f70610d9f561b65c774	Notes	2019-03-25
9.0	ghidra_9.0_PUBLIC_20190228.zip	3b65d29024b9decdbb1148b12fe87bcb7f3a6a56ff38475f5dc9dd1cfc7fd6b2	Notes	2019-02-28



  - GitHub:
    - https://github.com/NationalSecurityAgency/ghidra
    - The repo appears to be current, but the most recent Release isn’t at the time of this writing.

Getting Started: GhidraRun

- No installer, just extract and run
  - Linux & Mac: ghidraRun
  - Windows: ghidraRun.bat

```
student@ubuntu:~/ghidra_9.1-BETA_DEV_20190923/ghidra_9.1-BETA_DEV$ ls
docs  Extensions  Ghidra  ghidraRun  ghidraRun.bat  GPL  LICENSE  licenses  server  support
student@ubuntu:~/ghidra_9.1-BETA_DEV_20190923/ghidra_9.1-BETA_DEV$ ./ghidraRun
```

- Can double-click, or run from commandline
  - Recommend making a softlink/shortcut on the Desktop and in the Home directory

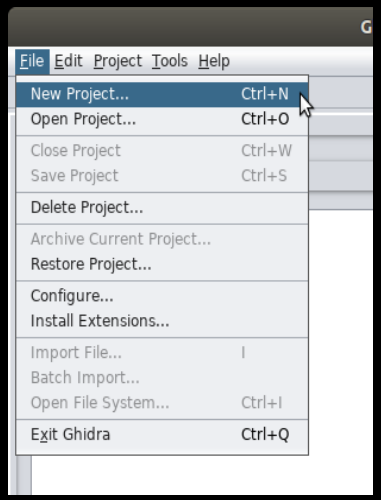
- Click through the User Agreement and move on

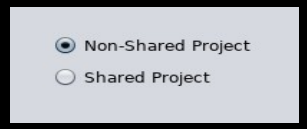
Time For a New Project!

(Like I don’t have enough of those..)

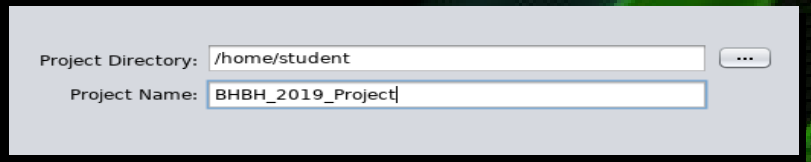
New Project! New Project!

File → New Project
(Or Ctrl-N)



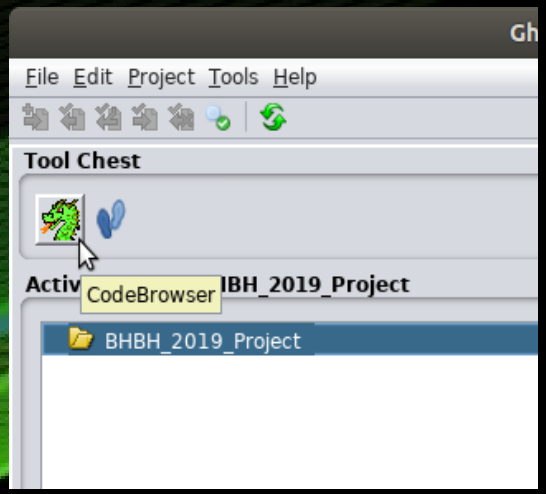


Not covering collaboration features in this course

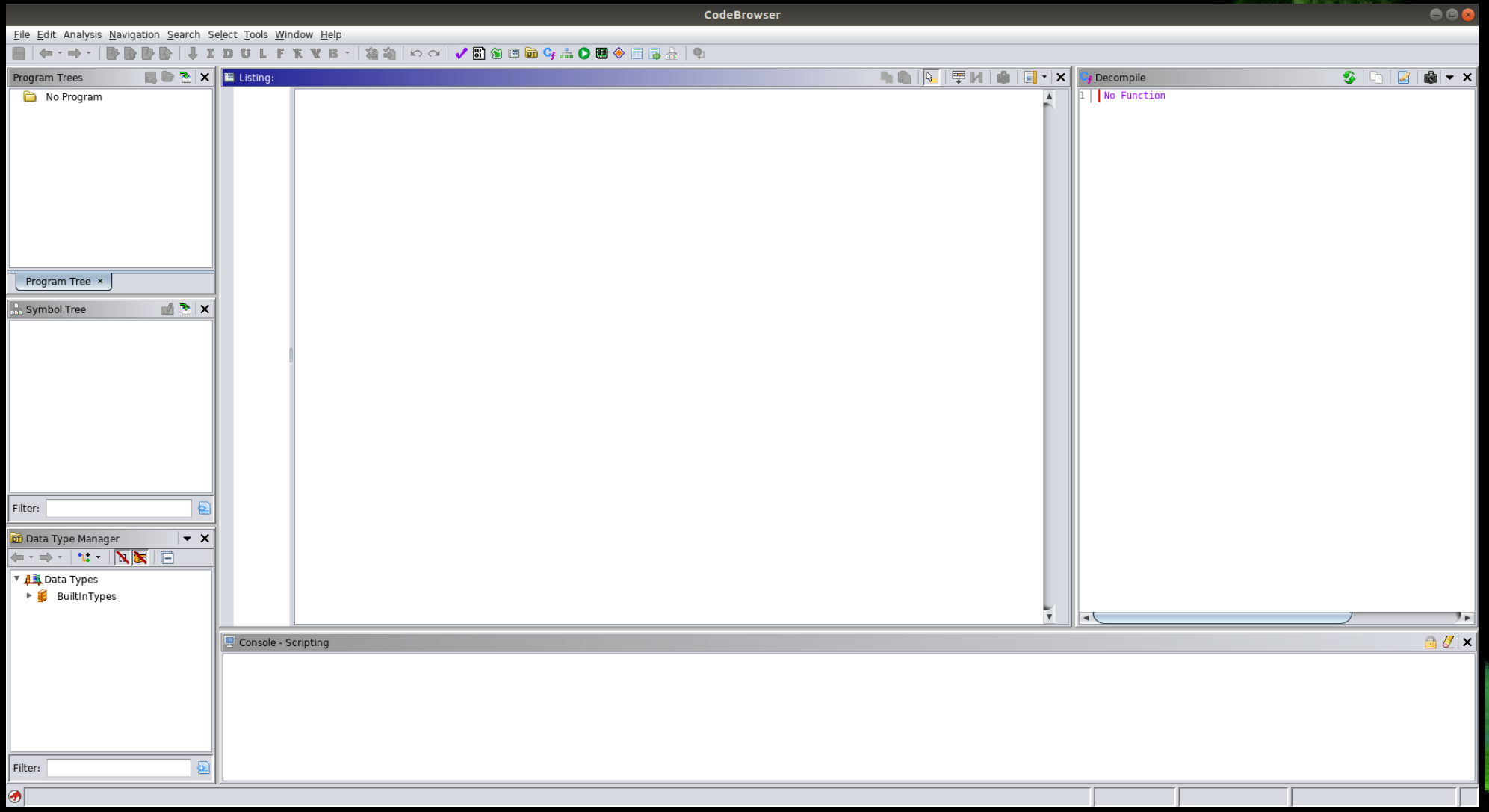


Select whatever folder and project name you like

Once you've made a project, poke the dragon

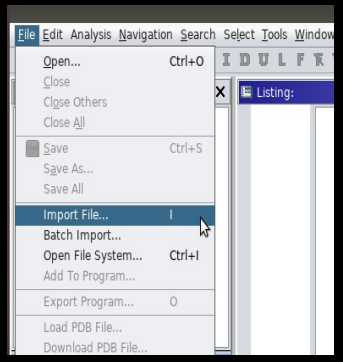


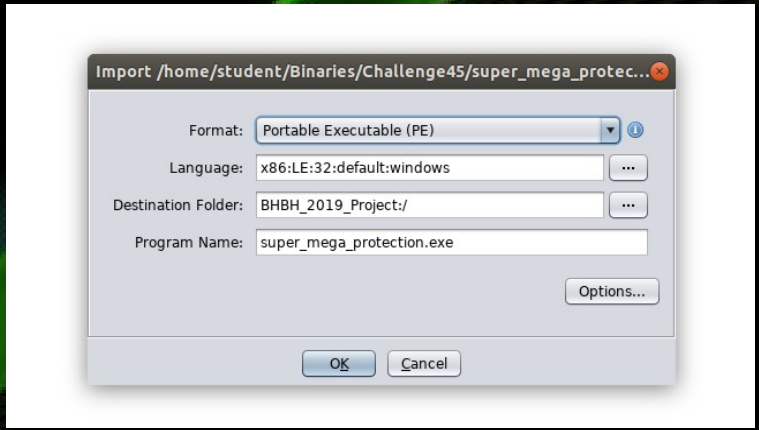
Kinda Empty in Here...



Importing Files

- Ways to import:
  - File → Import File
  - Press I
  - Drag-and-drop

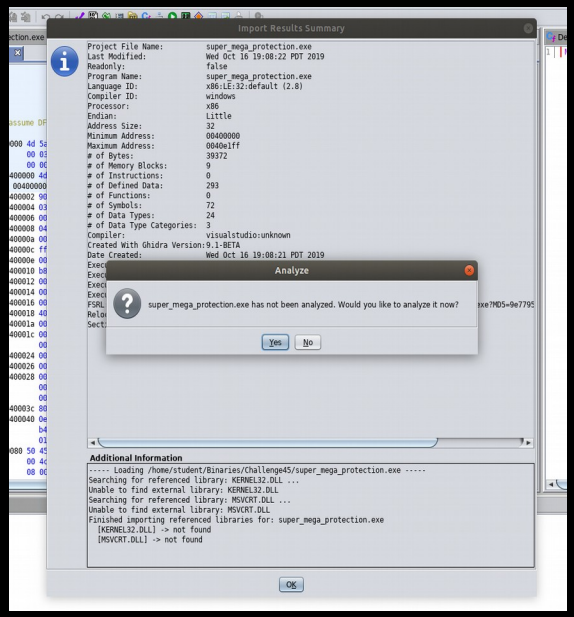


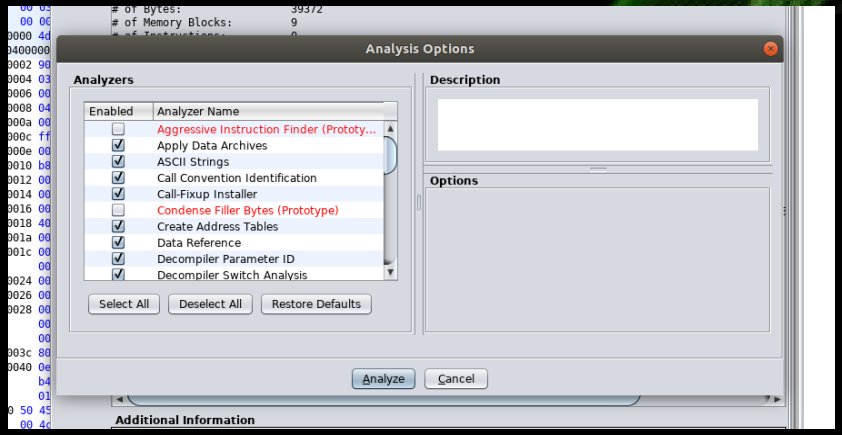


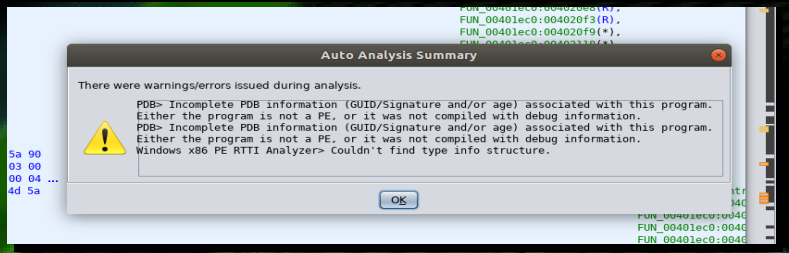
- Common formats and architectures are often detected automatically.
- Unusual or encrypted ones often aren't, at which point you have to make some manual selections

Auto Analysis

- Ghidra will ask if you want it to analyze the file. Usually saying ‘yes’ and running with defaults is fine





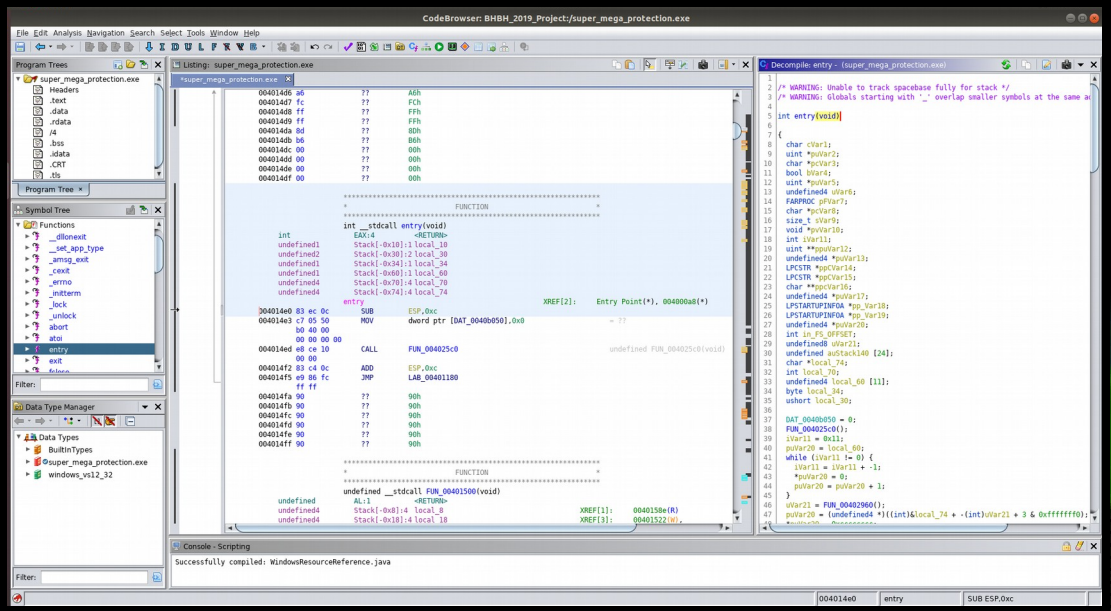


- Skim over the Import Results Summary and hit OK
- Sometimes you’ll get warnings after it finishes, but they’re not always the most useful or important things
- You can ask it to auto-analyze again later at pretty much any point

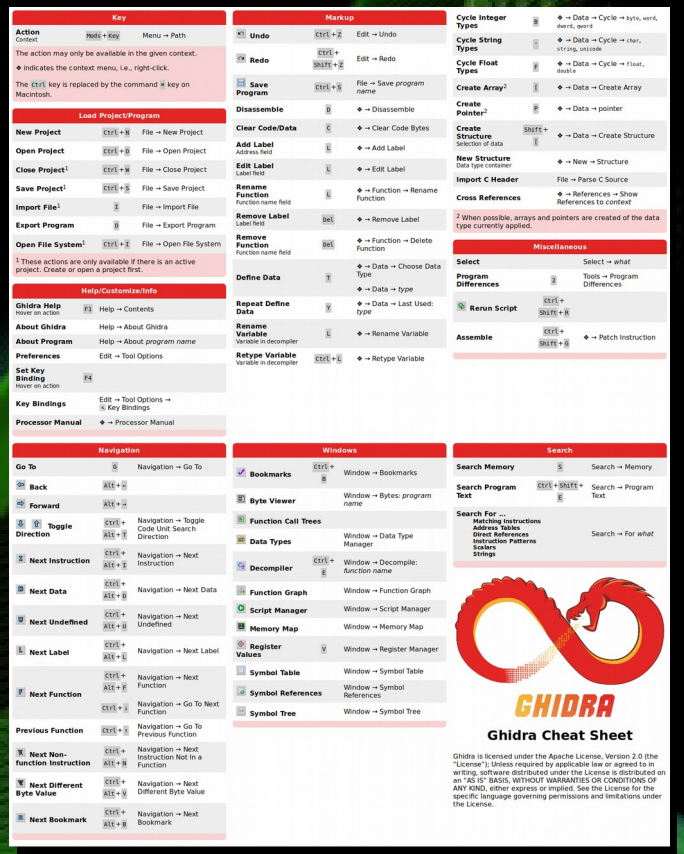
Let’s Take a Tour

Touring the Interface

- The Program is the terrain

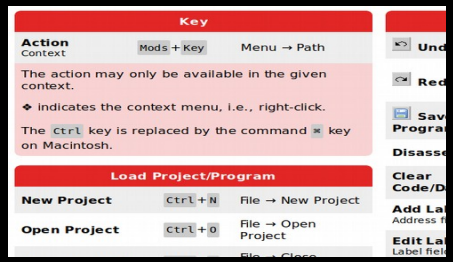


- The cheatsheet is the map
  - Provided by the developers: https://ghidra-sre.org/CheatSheet.html
  - Lets take a look at each section, especially Windows and Searches

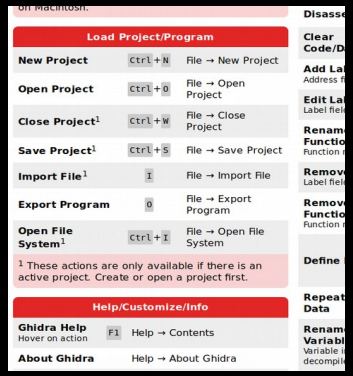


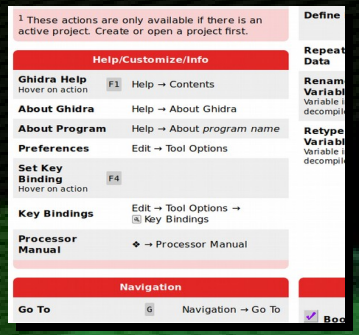
Tour: Key, File Operations, Help

- Key – Pretty straight forward



- Load Project/Program – Most of this we've already covered

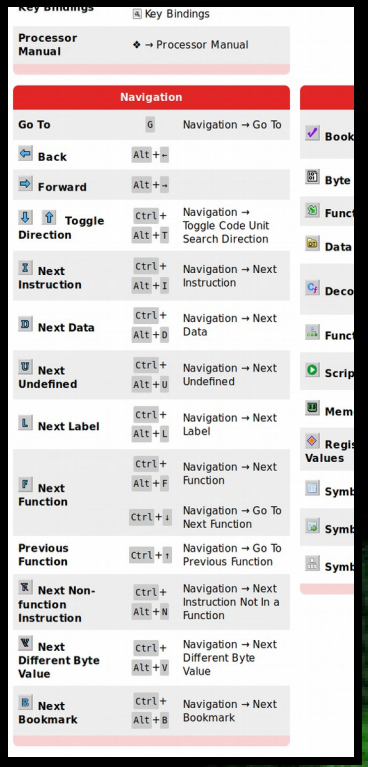


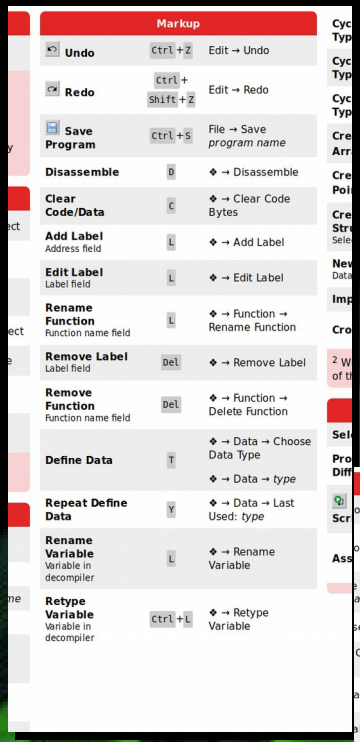


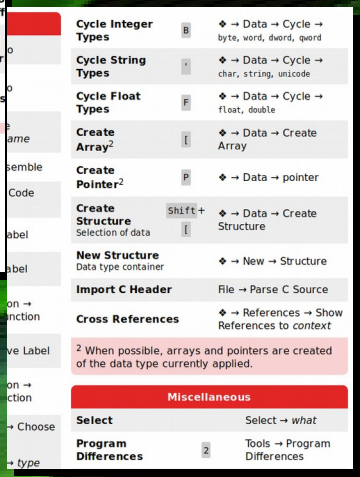
- Help/Customize/Info – Ghidra's help function is actually kind of awesome. You can also add in the processor manuals to be able to access information about the instruction set directly within the program

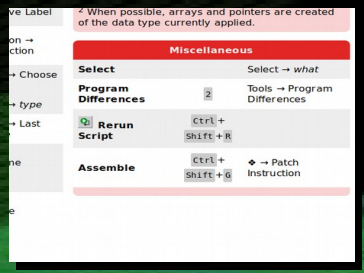
Tour: Navigation, Markup and Misc

- Navigation helps you move through the code
- Markup lets you map and terraform the terrain so to speak







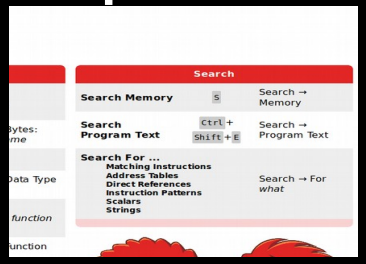


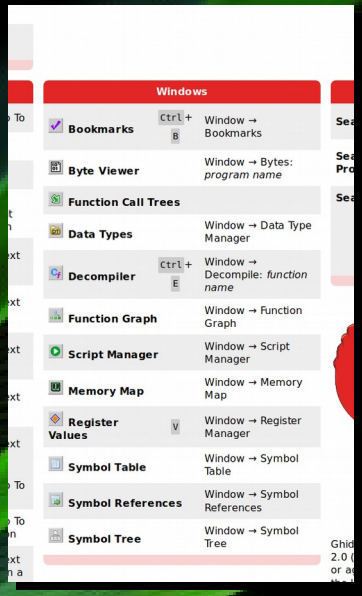
Misc

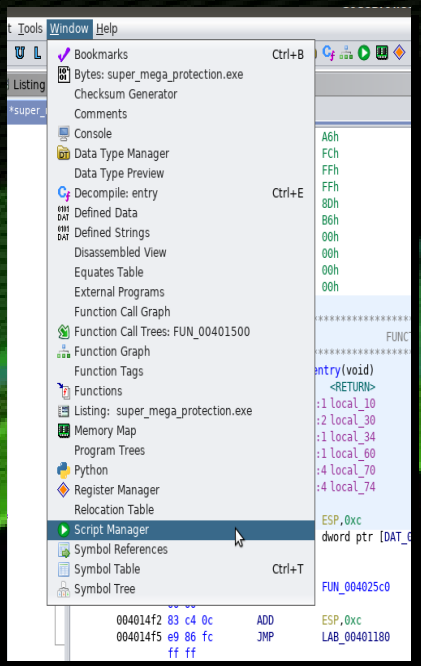
- Program Differences – essentially a side-by-side, visual diff inside Ghidra
- Rerun Script – Exactly what it sounds like
- Assemble – Modify specific instructions in the code

Tour: Different Screens and Searches

- The search functions let you hone in on specific kinds of information quickly
- The different windows lead to different types of data and functionality. While we'll look at each now, the Script Manager will be covered in depth in a moment

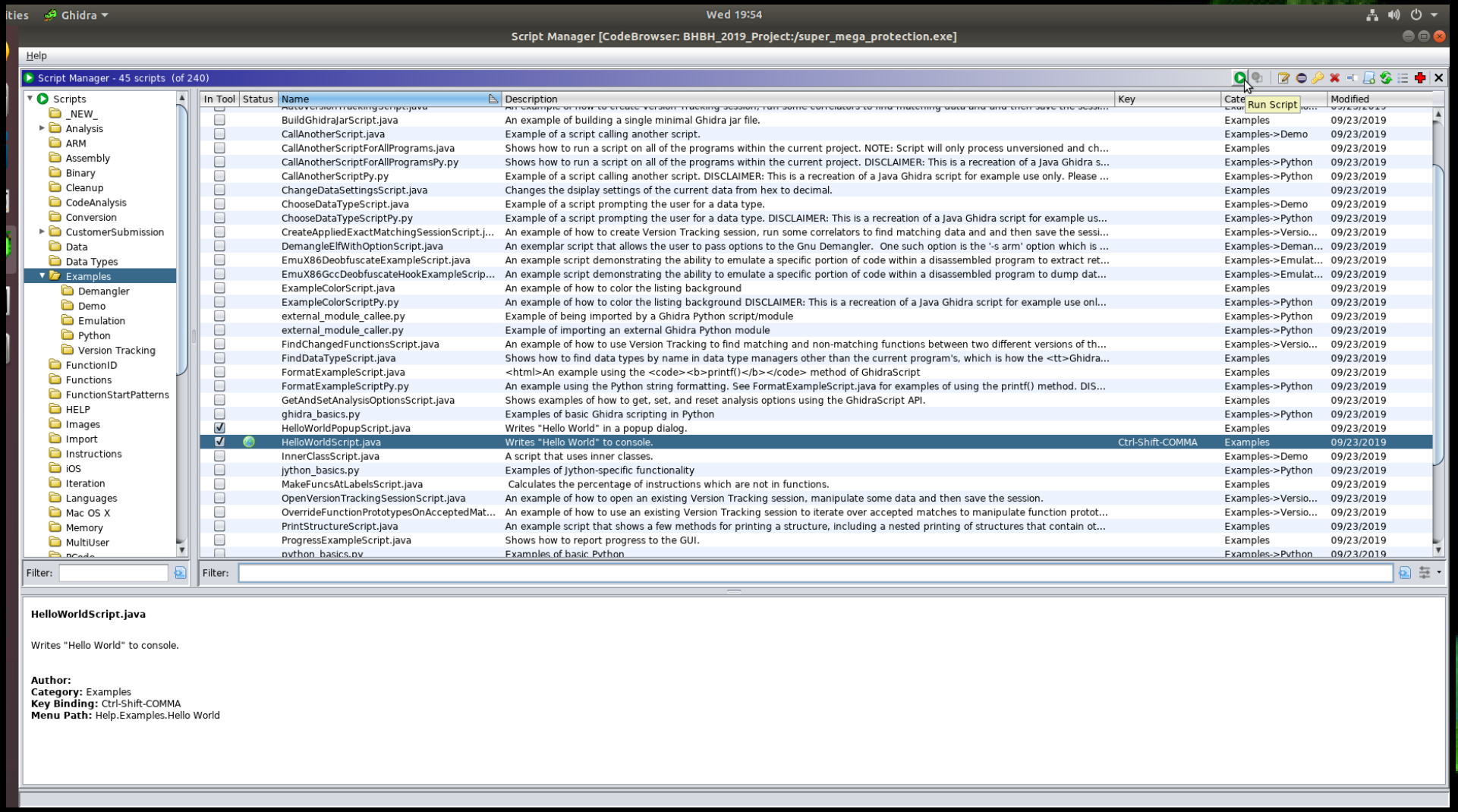






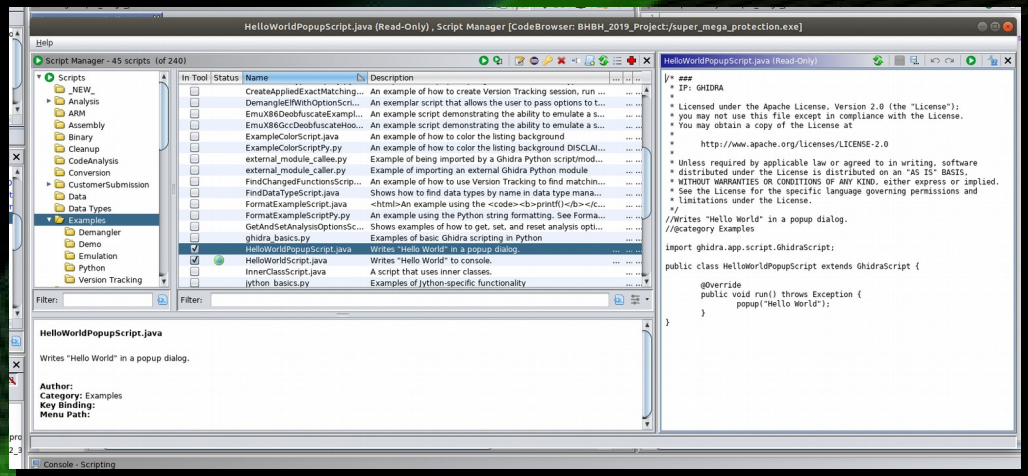
Scripts in Ghidra

Behold, The Script Manager!

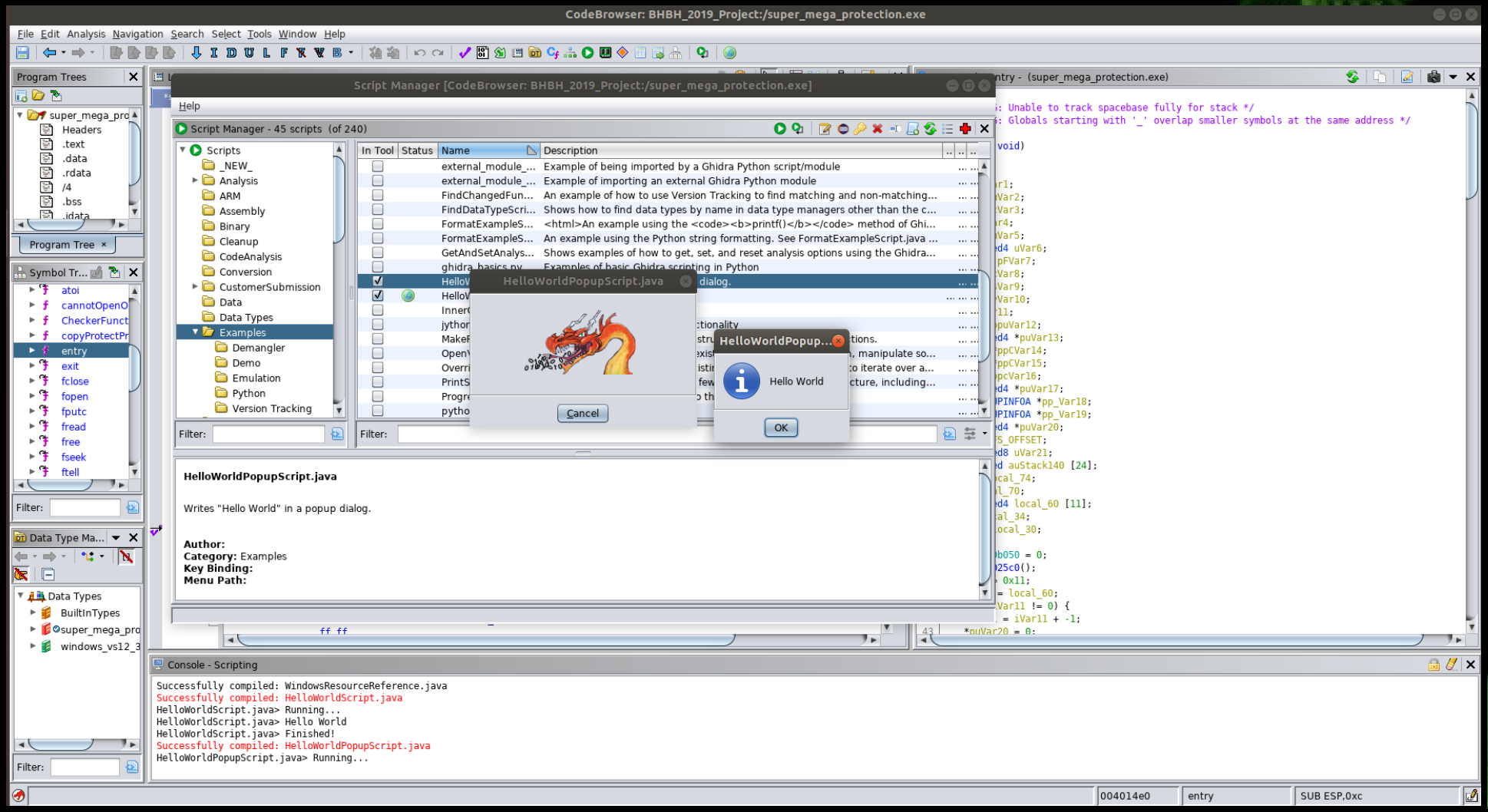


More About Script Manager

- Comes with lots of useful and example scripts
- Can load and run scripts directly from window
- Can assign them to a hotkey for usage anywhere in Ghidra
- Has a built-in editor (can also launch Eclipse)



Hello World


CodeBrowser: BHBH_2019_Project:/super_mega_protection.exe
Script Manager [CodeBrowser: BHBH_2019_Project:/super_mega_protection.exe]
HelloWorldPopupScript.java
Cancel
HelloWorldPopup...
Hello World
OK
HelloWorldPopupScript.java
Writes "Hello World" in a popup dialog.
Author:
Category: Examples
Key Binding:
Menu Path:
Console - Scripting
Successfully compiled: WindowsResourceReference.java
Successfully compiled: HelloWorldScript.java
HelloWorldScript.java> Running...
HelloWorldScript.java> Hello World
HelloWorldScript.java> Finished!
Successfully compiled: HelloWorldPopupScript.java
HelloWorldPopupScript.java> Running...

Java vs Python

- Java
  - Ghidra is built on it, might mesh better
  - Out of the hundreds of scripts prepackaged, most are Java. Less than 20 are in python
  - Object Oriented
- Python
  - Based on Jython (Java Python)
  - Only supports Python 2
  - Probably still easier to learn

Running Headless

- ‘Headless’ ← running from commandline, no GUI
- Scripts can be Java or Python
  - Ex: analyzeHeadless /Users/user/ghidra/projects MyProject -import hello.exe -preScript Script.java

- Useful for going over large amounts of code, files or data and performing some kind of action over and over again. Useful in:
  - Analyzing multiple binaries
  - Processing code or files (renaming, saving, logging, etc.)
  - Configuring things for other processes or later use
  - Anything that could be scripted. The readme has lots of examples.

- Let’s take a quick look at the official readme: https://ghidra.re/ghidra_docs/analyzeHeadlessREADME.html

Headless Exercise

- Try running a couple scripts headless

```
File  Edit  View  Search  Terminal  Help
student@ubuntu:~/ghidra_9.1-BETA_DEV/support$ ./analyzeHeadless ~/BHBH_2019_Project.rep/ BHBH_2019_P
roject -import /usr/lib/firefox/firefox -postscript FindInstructionsNotInsideFunctionScript.java
```

- Keep in mind the scripts won't be able to use the GUI

```
INFO  HEADLESS Script Paths:
    /home/student/ghidra_scripts
    /home/student/ghidra_9.1-BETA_DEV/Ghidra/Features/Base/ghidra_scripts
    /home/student/ghidra_9.1-BETA_DEV/Ghidra/Features/BytePatterns/ghidra_scripts
    /home/student/ghidra_9.1-BETA_DEV/Ghidra/Features/Decompiler/ghidra_scripts
    /home/student/ghidra_9.1-BETA_DEV/Ghidra/Features/FileFormats/ghidra_scripts
    /home/student/ghidra_9.1-BETA_DEV/Ghidra/Features/FunctionID/ghidra_scripts
    /home/student/ghidra_9.1-BETA_DEV/Ghidra/Features/GnuDemangler/ghidra_scripts
    /home/student/ghidra_9.1-BETA_DEV/Ghidra/Features/Python/ghidra_scripts
    /home/student/ghidra_9.1-BETA_DEV/Ghidra/Features/VersionTracking/ghidra_scripts
    /home/student/ghidra_9.1-BETA_DEV/Ghidra/Processors/8051/ghidra_scripts
    /home/student/ghidra_9.1-BETA_DEV/Ghidra/Processors/DATA/ghidra_scripts
    /home/student/ghidra_9.1-BETA_DEV/Ghidra/Processors/PIC/ghidra_scripts (HeadlessAnalyzer)
```

Ghidra Flat API

- Ghidra’s Program API is available, but the Flat API is more stable and intended for users to use. Program is more for developing Ghidra itself.
- Lots and lots of useful functions. Small number of examples:
  - getFunctionContaining()
  - getReferencesTo()
  - getEntryPoint()
  - getFromAddress()
  - getNext()
  - setBackgroundColor()
  - AddressSet()
- Take a look: https://ghidra.re/ghidra_docs/api/ghidra/program/flatapi/FlatProgramAPI.html


Time to get our hands dirty

The Plan

- Write Hello World together in the Script Manager and run it.
- Pick 1 function from the Flat API, make a script and bind it to a hotkey (color coding things is always nice)
- Write a simple headless script and run it

Group Playtime

- CMU Bomb Lab (Classic Reversing Challenge)
  - https://www.cs.cmu.edu/afs/cs/academic/class/15213-s02/www/applications/labs/lab2/bomblab.html
  - https://github.com/luong-komorebi/Binary-Bomb/
- This is a classic reversing challenge where you find the correct input to defuse each stage of a ‘bomb’ (it literally says “boom”)
- Everyone gets a unique bomb, but the way to get to the solutions is always the same
- Scripting Twist: Instead of just solving the challenge, create a script that can go through, find the passcode, and change it to be the name of that file instance instead.
- Feel free to look up walkthroughs, the focus here is on being able to script the process

Bonus Objectives:

- Have the script place a bookmark at the passcode
- Change the background color of the passcode in Ghidra
- Have it change the color of every spot it touches (eg, if you use getNext() or getFunctionContaining(), color where it ‘lands’)
  - Make it proceed chromatically
  - Have the color cycle persist and continue across binaries (hard mode)
- Have it solve/work on additional phases

Hope you enjoyed!

Questions, comments, or just geeking out?

Contact info:

- Twitter: @SynapticRewrite
- MorganWhitlow@Gmail.com