



FA2023 Week 03 • 2023-09-17

# Reverse Engineering Setup

Pete Stenger

# Announcements

- We qualified for CSAW!!
  - Congrats, we will be sending a team to New York!



ctf.sigpwny.com

sigpwny{everything\_is\_open\_source}



# Table of Contents

- Tools to install
- Starter commands
- Get started

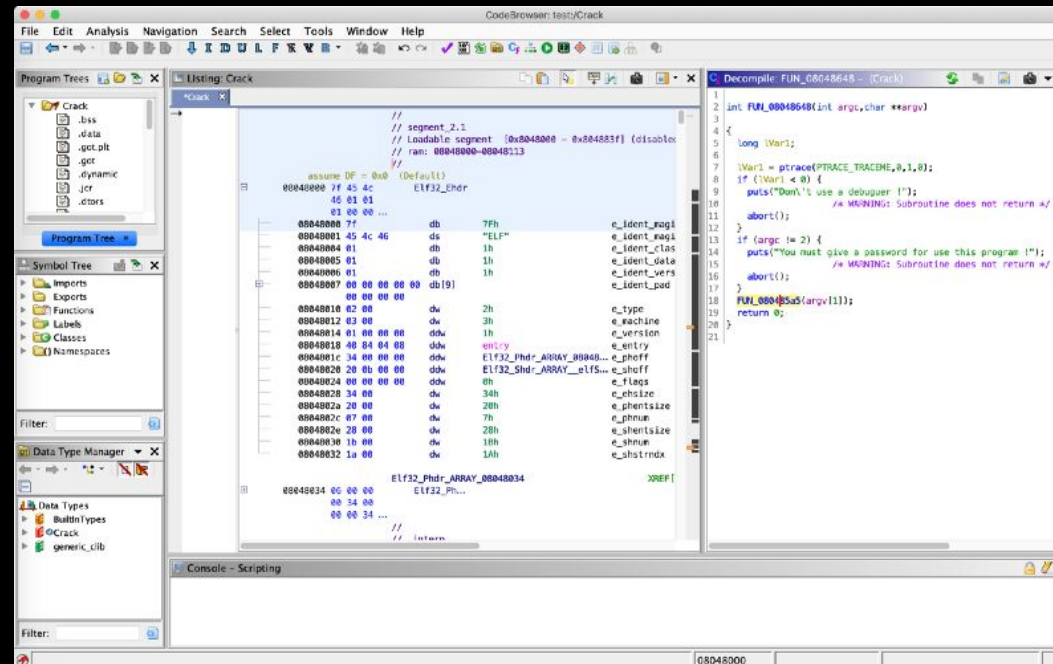


# Tool Installation



# What is Ghidra?

- Ghidra is a reverse engineering toolkit developed by the NSA and made open source
- Allows you to disassemble applications - essentially turn an unreadable application into readable code



# JDK on Windows / Mac



# Check if you have Java

Java should NOT be <11

```
Last login: Sat Sep 16 22:50:17 on tty3003
~ > java -version
openjdk version "20.0.1" 2023-04-18
OpenJDK Runtime Environment Homebrew (build 20.0.1)
OpenJDK 64-Bit Server VM Homebrew (build 20.0.1, mixed mode, sharing)
~ > █
```





# Installing Java Developer Kit

Install JDK 11+ (**not JRE!**) from Oracle

- The latest version should work

<https://www.oracle.com/java/technologies/downloads/#jdk20-windows>

**Java 20 and Java 17 available now**

JDK 20 is the latest release of Java SE Platform and JDK 17 LTS is the latest long-term support release for the Java SE platform.

[Learn about Java SE Subscription](#)

[JDK 20](#) [JDK 17](#) [GraalVM for JDK 20](#) [GraalVM for JDK 17](#)

**JDK Development Kit 20.0.2 downloads**

JDK 20 binaries are free to use in production and free to redistribute, at no cost, under the [Oracle No-Fee Terms and Conditions](#).

JDK 20 will receive updates under these terms, until September 2023 when it will be superseded by JDK 21.

[Linux](#) [macOS](#) [Windows](#)

Product/file description	File size	Download
x64 Compressed Archive	180.99 MB	<a href="https://download.oracle.com/java/20/latest/jdk-20_windows-x64_bin.zip">https://download.oracle.com/java/20/latest/jdk-20_windows-x64_bin.zip</a> (sha256)
x64 Installer	160.12 MB	<a href="https://download.oracle.com/java/20/latest/jdk-20_windows-x64_bin.exe">https://download.oracle.com/java/20/latest/jdk-20_windows-x64_bin.exe</a> (sha256)
x64 MSI Installer	158.90 MB	<a href="https://download.oracle.com/java/20/latest/jdk-20_windows-x64_bin.msi">https://download.oracle.com/java/20/latest/jdk-20_windows-x64_bin.msi</a> (sha256)

# JDK on Linux

Note that we recommend installing JDK and Ghidra on Windows  
(**not** WSL)



# Installing JDK

```
sudo apt update
```

```
sudo apt install openjdk-19-jdk
```



# Downloading Ghidra

<https://github.com/NationalSecurityAgency/ghidra/releases>




or Google "github ghidra release"








Download the public archive in assets for the latest release (ghidra\_X.X.X\_PUBLIC\_XXXXXXXXX.zip, not Source code.zip)

## Ghidra 10.1.5 Latest

- [What's New](#)
- [Change History](#)
- [Installation Guide](#)
- SHA-256: `17db4ba7d411d11b00d1638f163ab5d61ef38712cd68e462eb8c855ec5cfb5ed`

▼ Assets 3

 ghidra_10.1.5_PUBLIC_20220726.zip	328 MB	Jul 26, 2022
 Source code (zip)		Jul 26, 2022
 Source code (tar.gz)		Jul 26, 2022

  62  7  31  26  12  7

89 people reacted



# Running Ghidra

## Windows:

Double click `ghidraRun.bat`

## Mac/Linux:

Open Terminal, navigate to the directory where Ghidra is downloaded using something like ``cd ~/Downloads/ghidra_XX``

Make ghidraRun executable: ``chmod +x ./ghidraRun``

Launch Ghidra: ``./ghidraRun``



# Running Ghidra - Mac

After installation, Ghidra needs permission to run the decompiler binaries

1. Open an x86 binary, and run through the default decompiler
2. When you receive an error, go back to the "Privacy & Security" tab of settings, and hit "allow" on the binary that appears there
3. Repeat until you receive no errors when decompiling!



# Python and Pwntools



# What is pwntools?

pwntools is a CTF framework and exploit development library.  
Intended to **make exploit writing as simple as possible.**

```
>>> sh = process('/bin/sh')
>>> sh.sendline(b'sleep 3; echo hello world;')
>>> sh.recvline(timeout=1)
b''
>>> sh.recvline(timeout=5)
b'hello world\n'
>>> sh.close()
```





# Installing Python (Basic)

## Mac:

```
brew install python  
python3 -m ensurepip
```

Wait for the next slide if you  
would like a more robust  
setup!

## Windows (WSL)/Linux:

```
sudo apt update  
sudo apt install python3 python3-pip
```

We recommend Windows users use Python/pwntools in WSL  
rather than native Windows



# Installing Python (pyenv)

pyenv - version manager for Python

- easily switch between Python versions as needed

```
curl https://pyenv.run | bash
```



# Installing Pwntools

```
python3 -m pip install pwntools
```

If you get "command not found" you may need to refresh the shell environment

```
source ~/.bashrc
```

```
(source ~/.zshrc)
```



# Installing Pwntools

Apple silicon "unicorn error"

```
$ git clone https://github.com/unicorn-engine/unicorn.git
$ brew install cmake
$ brew install qemu
$ cd unicorn/bindings/python
$ python setup.py install
```



# What is GDB + GEF?

## GDB - Gnu DeBugger

- Debug x86 programs (we will teach you how in **Rev II: x86 Reversing**)

## GEF - GDB Enhanced Features

- Adds lots of nice features useful for binary exploitation and reverse-engineering

## [pwndbg](#) - Alternative for GEF

- More advanced than GEF, slightly different syntax



# Installing GDB + GEF

## Mac:

Just use Docker container, preinstalled

## Windows (WSL)/Linux:

```
sudo apt install gdb
```

```
bash -c "$(curl -fsSL https://gef.blah.cat/sh)"
```



# x86 Docker Container

For debugging and running x86 applications



# Installation (Mac M1/M2 only)

Enable Rosetta:

```
/usr/sbin/softwareupdate --install-rosetta  
--agree-to-license
```

Download Docker Desktop

- [docker.com/products/docker-desktop](https://docker.com/products/docker-desktop)

MUST BE **4.16.0 or newer** to work on Apple Silicon

- enable 'Use Virtualization Framework' in 'Settings > General'
- enable 'Use Rosetta for x86/amd64 on Apple Silicon' in 'Settings > Features in Development'

Clone our Docker Container

```
git clone https://github.com/sigpwny/pwn-docker.git  
cd pwn-docker
```

**You must be  
running macOS  
12.3 or newer!**





# Usage

`./start.sh`

Run to initialize your container. Type 'y' to initialize a permanent container, 'n' for a temporary container

`./run.sh`

Connect to your permanent container after it has been closed

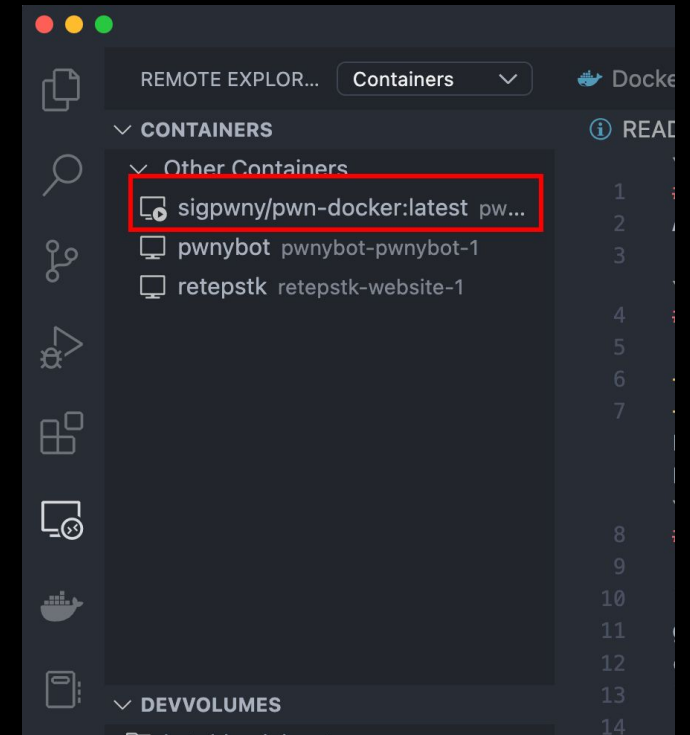
GDB does not work yet :(

– We are trying - it should be theoretically possible, but it is hard to setup



# Visual Studio Code

- Install the "Dev Containers" extension
- Or, work inside the ~/ctf directory (shared with docker)



# Next Steps

- Solve `first_re`, and look at other Rev challenges in the Vault
- Work on CTF problems!



# Next Meetings

## 2023-09-21 • This Thursday

- Reverse Engineering I
- Interpreter reverse engineering (Python/JavaScript)

## 2023-09-23 • This Saturday!

- **Fall CTF 2023**
- First 350 registered people to show up ([sigpwny.com/register23](https://sigpwny.com/register23)) get an electronic badge!  
Also, free shirt + pizza!



ctf.sigpwny.com

**sigpwny{everything\_is\_open\_source}**

**Meeting content can be found at**  
**[sigpwny.com/meetings](https://sigpwny.com/meetings).**

