

SP2023 Week 01 • 2023-1-29

Setup Meeting v2

Pete, Minh, Richard



Announcements

- DiceCTF
 - Playing Friday virtually, type `/ctf optin` in #bot-commands to be notified



Pwny CTF (ctf.sigpwny.com)

- Create an account right now!
- Where we put our challenges for you to build hands on experience
- Solve challenges, find flags, submit flags on website



WARNING before we go any further!

(The “Don’t Get Arrested” Slide)

- We will teach you things that you could use unethically & illegally
- <https://www.law.cornell.edu/uscode/text/18/1030>
 - Read it!
- CFAA TLDR
 - Computer Fraud and Abuse Act
 - Attacking “protected” computers
 - Anywhere between a fine and **TWENTY** years in jail.
- If you don’t have EXPLICIT permission to break into it, **DON’T**
- We are NOT lawyers and CAN’T give you legal advice

We are NOT suggesting, telling, or implying you should actually do these things. By participating in this club and agreeing to our Code of Conduct, you agree that your actions are your own and you will deal with the consequences.



Marcus Hutchins, Controversial Hacker who saved the internet, got arrested for past crimes.



ctf.sigpwny.com

sigpwny{setup_v2}



Table of Contents

- What is a shell
- Getting into the shell
- Tools to install
- Starter commands
- Get started



> The Terminal

"It's where things happen" - Ravi



```
CSAW2020 /dev/tty000
ls
bard      grid      kui_blox1_sol.png
bard.hop  grid_solve.py  libc-2.27.so
ezbreezy  krakme.exe    solve_ezbreezy.py
CSAW2020
```

```
mark@linux-desktop: ~
File Edit View Search Terminal Help
mark@linux-desktop:~$
```

```
tquig@THOMAS-PC: ~
tquig@THOMAS-PC:~$
```



Linux



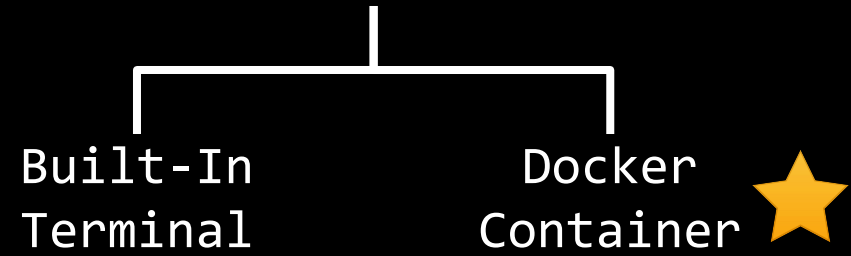
You're good to go!



Windows



macOS



PowerShell? Command Prompt?

- Those are shells too!
- However, they're limited in tools and are Windows-based terminals, not Linux-based



Windows Subsystem for Linux



Installing WSL

- Open command prompt as administrator
 - (Start button → type **cmd** → right click → open as administrator)
- Type **wsl --install**
- Restart computer
- Open command prompt
- Enter WSL by typing **wsl**
- You now have a linux shell

```
Administrator: Windows PowerS... x + v - □ x
PS C:\Users\chris> wsl --install
Installing: Virtual Machine Platform
Virtual Machine Platform has been installed.
Installing: Windows Subsystem for Linux
Windows Subsystem for Linux has been installed.
Downloading: WSL Kernel
Installing: WSL Kernel
WSL Kernel has been installed.
Downloading: GUI App Support
Installing: GUI App Support
GUI App Support has been installed.
Downloading: Ubuntu
[===== 43.3% ]
```



Set a "root" user

Select a username and password for your administrative user.

```
hayden@T470s ~  
Installing, this may take a few minutes...  
Please create a default UNIX user account. The username does not need to match your Windows username.  
For more information visit: https://aka.ms/wslusers  
Enter new UNIX username: hayden  
Enter new UNIX password:  
Retype new UNIX password:  
passwd: password updated successfully  
Installation successful!  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
hayden@T470s:~$
```



macOS Terminal

Command
+ Space



Search "Terminal"



```
→ CSAW2020 ls
bard          grid          kui_blox1_sol.png
bard.hop     grid_solve.py libc-2.27.so
ezbreezy     krakme.exe   solve_ezbreezy.py
→ CSAW2020
```

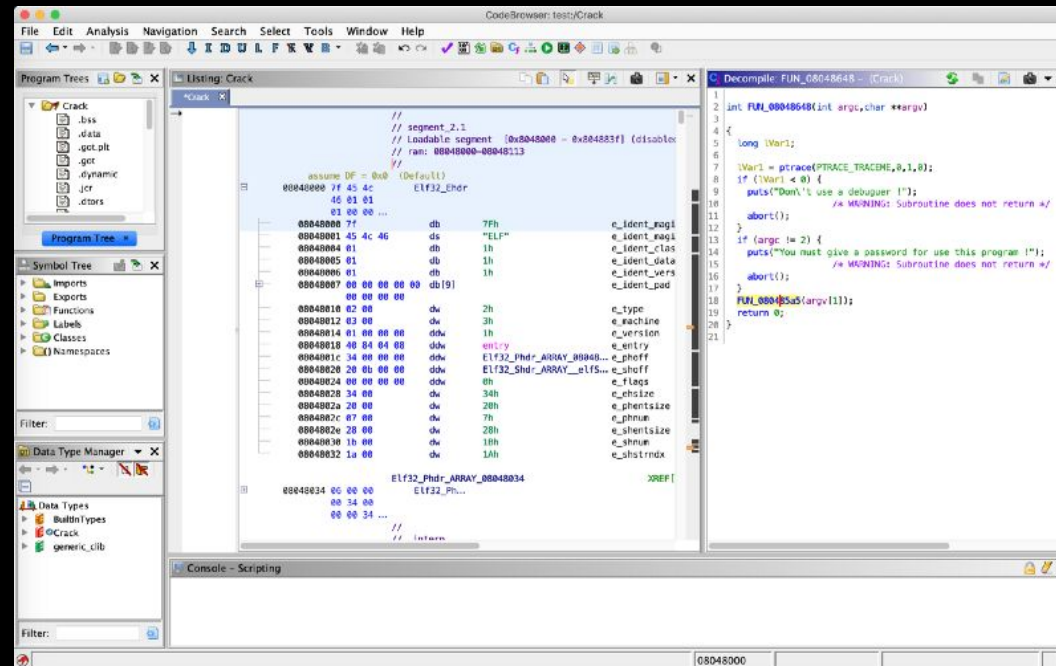


Tool Installation



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



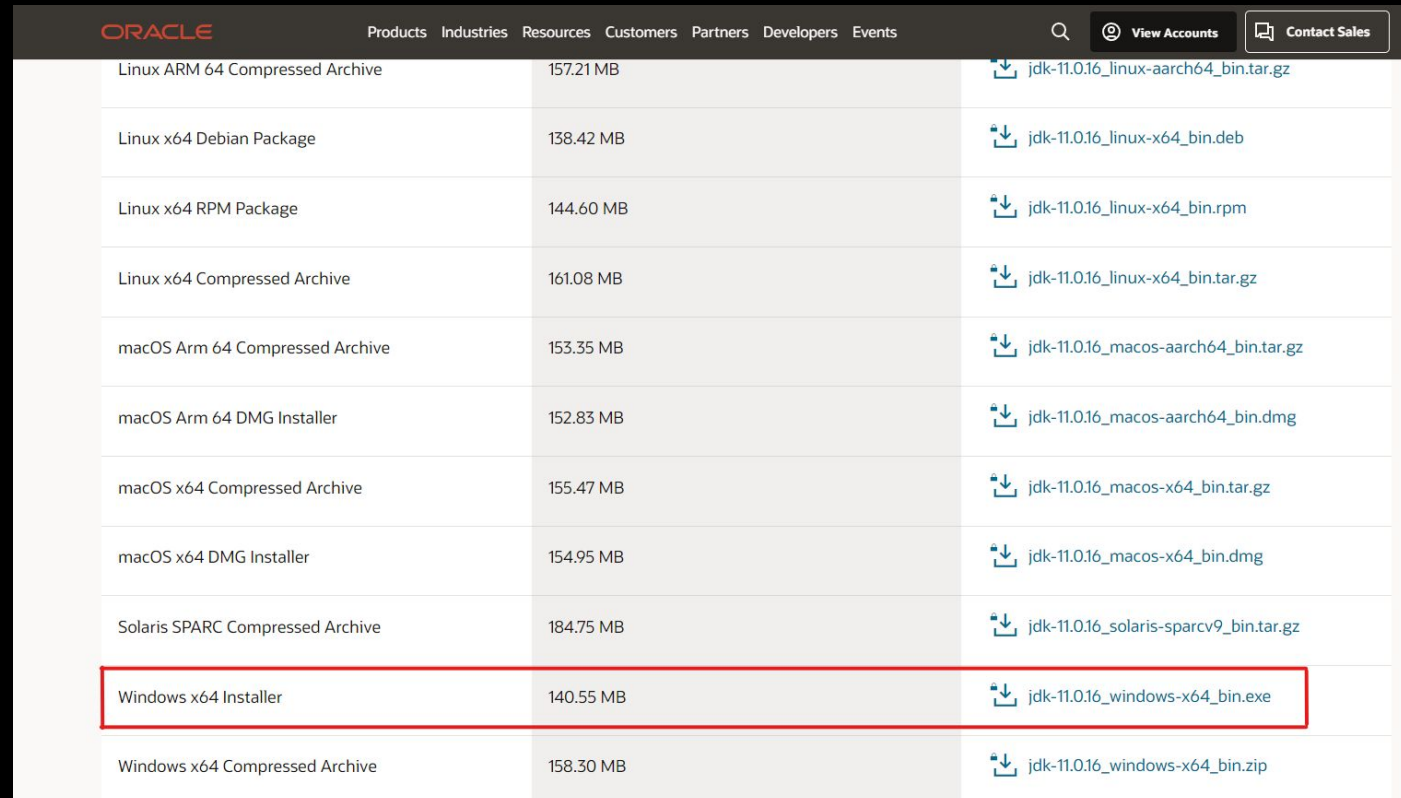
Installing Java Developer Kit

Install JDK 19 (not JRE!) from Oracle

<https://www.oracle.com/java/technologies/javase/jdk19-archive-downloads.html>

or Google

"oracle java se 19"



Operating System / Architecture	Size	Download Link
Linux ARM 64 Compressed Archive	157.21 MB	jdk-11.0.16_linux-aarch64_bin.tar.gz
Linux x64 Debian Package	138.42 MB	jdk-11.0.16_linux-x64_bin.deb
Linux x64 RPM Package	144.60 MB	jdk-11.0.16_linux-x64_bin.rpm
Linux x64 Compressed Archive	161.08 MB	jdk-11.0.16_linux-x64_bin.tar.gz
macOS Arm 64 Compressed Archive	153.35 MB	jdk-11.0.16_macos-aarch64_bin.tar.gz
macOS Arm 64 DMG Installer	152.83 MB	jdk-11.0.16_macos-aarch64_bin.dmg
macOS x64 Compressed Archive	155.47 MB	jdk-11.0.16_macos-x64_bin.tar.gz
macOS x64 DMG Installer	154.95 MB	jdk-11.0.16_macos-x64_bin.dmg
Solaris SPARC Compressed Archive	184.75 MB	jdk-11.0.16_solaris-sparcv9_bin.tar.gz
Windows x64 Installer	140.55 MB	jdk-11.0.16_windows-x64_bin.exe
Windows x64 Compressed Archive	158.30 MB	jdk-11.0.16_windows-x64_bin.zip

JDK on Mac



Installing Java Developer Kit

Go to <https://brew.sh> and run the setup command

```
ret@laptop:~  
~/ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

If it is already installed, make sure to update with `brew update`

Install the latest version of openjdk java or newer: `brew install java`



Linking Java

```
⇒ Caveats
For the system Java wrappers to find this JDK, symlink it with
sudo ln -sfn /opt/homebrew/opt/openjdk@11/libexec/openjdk.jdk /Library/Java/JavaVirtualMachines/openjdk-11.jdk

openjdk@11 is keg-only, which means it was not symlinked into /opt/homebrew,
because this is an alternate version of another formula.

If you need to have openjdk@11 first in your PATH, run:
echo 'export PATH="/opt/homebrew/opt/openjdk@11/bin:$PATH"' >> ~/.zshrc
```

Link your Java JDK

THESE COMMANDS SHOULD BE COPIED FROM END OF BREW OUTPUT

```
sudo ln -sfn /opt/homebrew/opt/openjdk@11/libexec/openjdk.jdk
/Library/Java/JavaVirtualMachines/openjdk-11.jdk
```

Run `java -version` to check that openjdk 11 (or newer) is found

```
🍏 ~/ java -version
openjdk version "11.0.16.1" 2022-08-12
OpenJDK Runtime Environment Homebrew (build 11.0.16.1+0)
OpenJDK 64-Bit Server VM Homebrew (build 11.0.16.1+0, mixed mode)
🍏 ~/ 4█
```

If it isn't found, add jdk11 to your path

```
echo 'export PATH="/opt/homebrew/opt/openjdk@11/bin:$PATH"' >> ~/.zshrc
&& source ~/.zshrc
```



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
```

- Any version newer than JDK 11 is OK

That's it!

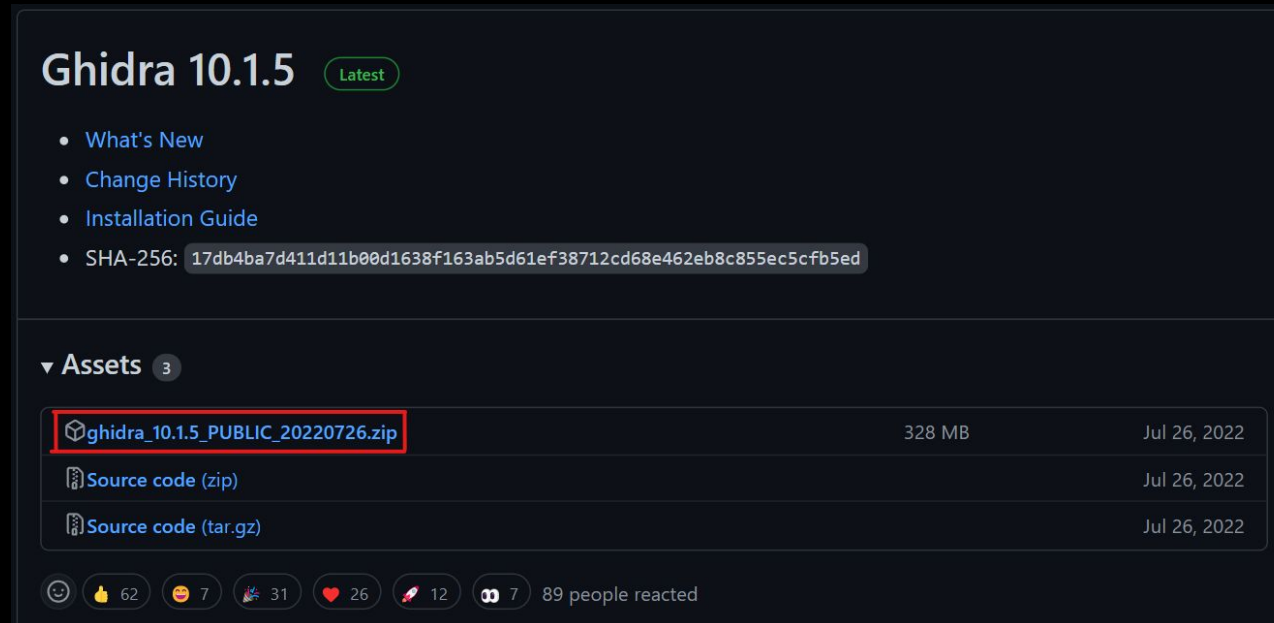


Downloading Ghidra

<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: 17db4ba7d411d11b00d1638f163ab5d61ef38712cd68e462eb8c855ec5c5fb5ed

▼ 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``



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

Mac:

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

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 Pwntools

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

If you get "command not found" you may need to reboot for Python/pip to be added to PATH



Installing GDB + GEF

Mac:

use docker container, not needed

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

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
```



Usage

```
./start.sh
```

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

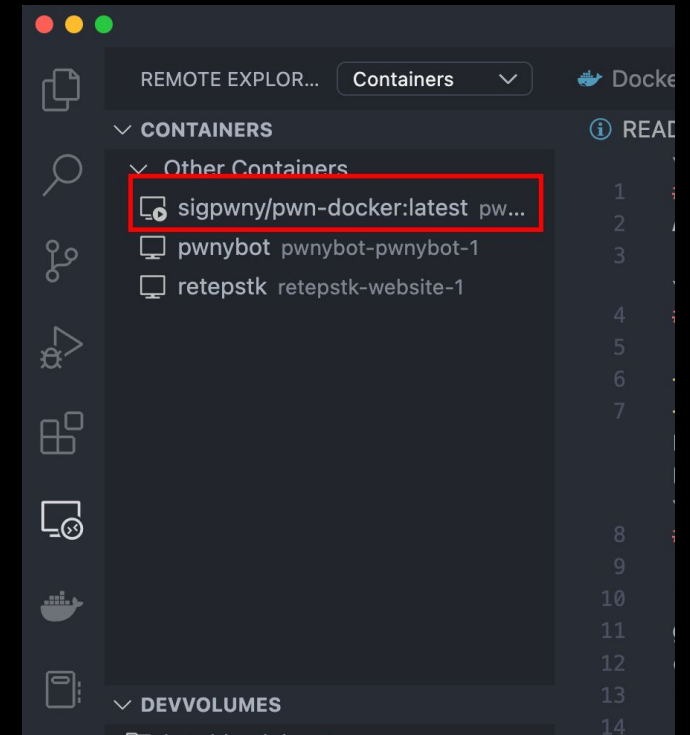
```
./run.sh
```

Connect to your permanent container after it has been closed



Visual Studio Code

- [Install](#) the “Dev Containers” extension
- Or, work inside the ~/ctf directory (shared with docker)



Useful Commands



Filesystem

ls [directory]: lists files in your current directory or specified directory

cd <directory>: changes your current directory to specified directory

mv <source> <dest>: moves file from source to dest (rename), if dest is a directory, move source

rm <file>: removes file (**NOT REVERSIBLE**)

cat <file>: prints the contents of file (sometimes it prints gibberish, think why that might happen)

./file: executes whatever is at file

man <command>: lets you see info about a command and all of its parameters/options

<parameter> means it's a required parameter

[parameter] means it's an optional parameter



Networking

`nc <ip> <port>`: netcat, connect to ip on port port. (First Command - netcat)

`ssh <user@ip> [port]`: secure remote shell, run an instance of a shell as user at the IP address

`ping <ip>`: see if an IP address is up using ICMP (usually blocked by firewalls)

`curl <url>`: network access tool that is mainly used to access websites from the terminal

`wget <url>`: Simplified/modern curl that downloads the file with relevant name



Networking Fundamentals

`nc -l <port>`: open a network socket to listen on specified port

`nc <ip> <port>`: open a connection to the specified IP and port

Ports - communication endpoints on your computer (1-65535)

Remember these two to connect to challenges!



Next Steps



Next Steps - Bandit

```
ssh bandit0@bandit.labs.overthewire.org -p 2220
```

command

user

IP

port



Next Steps - More Practice

Challenges

Meetings **Start Here** Vault

Welcome

Setup

Shell

OverTheWire - bandit

OverTheWire - Natas

OSINT I

Web I

Crypto I

Reverse Engineering I

CryptoHack

Welcome

Discord Authentication 50	SIGPwny Discord 50	Welcome to SIGPwny! 50	Feedback Form 100
------------------------------	-----------------------	---------------------------	----------------------

Setup

Setup Meeting Flag 50	Setup Meeting v2 Flag 50	A Very Special Character 20 beginner	netcat 20 beginner
--------------------------	-----------------------------	--	--------------------------



Next Meetings

2023-02-02 - This Thursday

- Web III (Advanced Web Hacking)
- SSRF, Template Injection, Command Injection

2023-02-05 - Next Sunday

- macOS Privilege Escalation
- Rohit will be talking about his \$XX,XXX bug he found

2023-02-03 - Dice CTF 2023

- Virtual CTF
- Type `/ctf optin` in #bot-commands



```
sigpwny{setup_v2}
```



SIGPwny