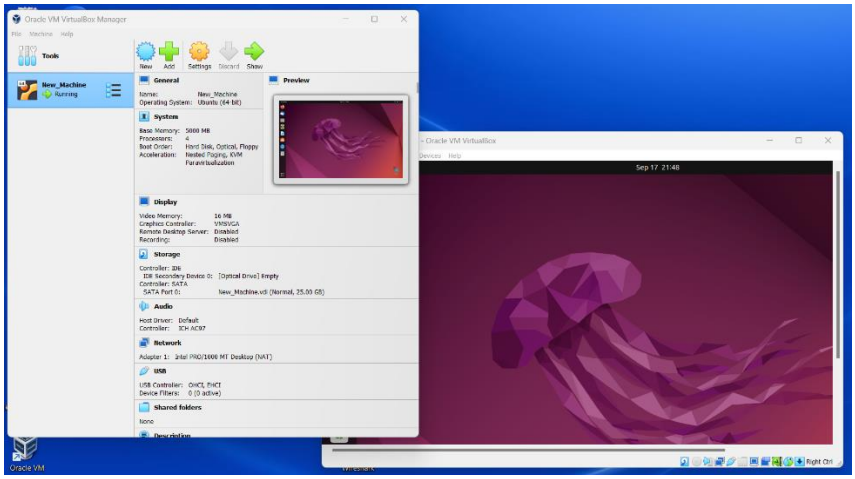
Create a Buffer Overflow on your Ubuntu VM: Lab 01

- Turn in screenshots to show your work
- Use docx or pdf
- Turn in only one document

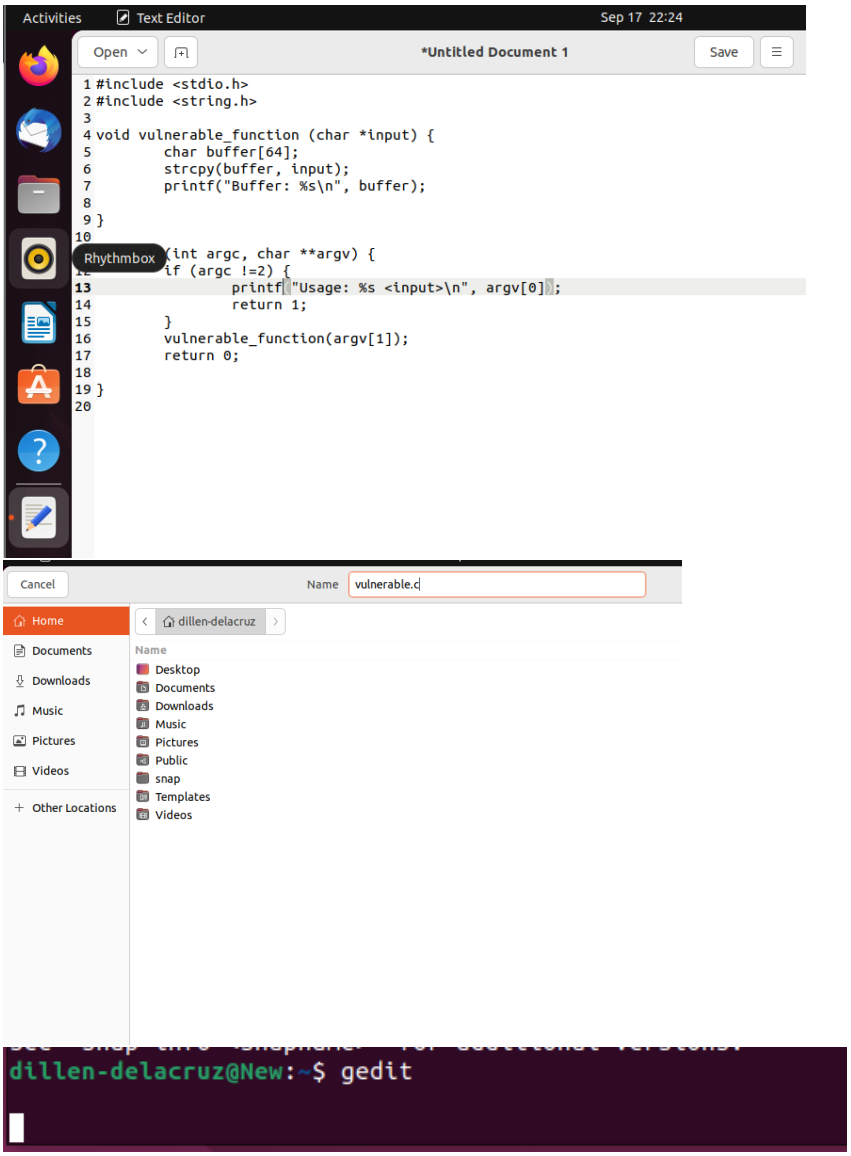
1. Set Up Ubuntu VM: (This part is your Lab 00)



2. Install Development Tools

```
dillen-delacruz@New:~$ sudo apt update
[sudo] password for dillen-delacruz:
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:2 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [765
kB]
Get:6 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [973
 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main i386 Packages [321 k
B]
Get:8 http://us.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [486
kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [165
kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metada
ta [43.1 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadat
a [11.3 kB]
dillen-delacruz@New:~$ sudo apt install build-essential gdb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
build-essential is already the newest version (12.9ubuntu3).
gdb is already the newest version (12.1-0ubuntu1~22.04).
gdb set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 75 not upgraded.
dillen-delacruz@New:~$
```

3. Create a Vulnerable C Program



4. Compile the Vulnerable Program

```
dillen-delacruz@New:~$ gcc -o vulnerable vulnerable.c -fno-stack-protector -m32
In file included from vulnerable.c:1:
/usr/include/stdio.h:27:10: fatal error: bits/libc-header-start.h: No such file
or directory
   27 | #include <bits/libc-header-start.h>
      |          ^~~~~~~~~~~~~~~~~~~~~~~~~~
compilation terminated.
dillen-delacruz@New:~$ sudo apt-get install gcc-multilib
[sudo] password for dillen-delacruz:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
Setting up gcc-11-multilib (11.4.0-1ubuntu1~22.04) ...
Setting up gcc-multilib (4:11.2.0-1ubuntu1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
dillen-delacruz@New:~$ gcc -o vulnerable vulnerable.c -fno-stack-protector -m32
dillen-delacruz@New:~$
```

5. Disable ASLR

```
dillen-delacruz@New:~$ sudo sysctl -w kernel.randomize_va_space=0
kernel.randomize_va_space = 0
dillen-delacruz@New:~$
```

6. Test the Vulnerable Program

```
dillen-delacruz@New:~$ ./vulnerable $(python -c 'print("A" * 80)')
Command 'python' not found, did you mean:
  command 'python3' from deb python3
  command 'python' from deb python-is-python3
Usage: ./vulnerable <input>
dillen-delacruz@New:~$ ./vulnerable $(python3 -c 'print("A" * 80)')
Buffer: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAA
Segmentation fault (core dumped)
dillen-delacruz@New:~$
```

7. **And 8. Set Up GDB for Debugging and Debug the Vulnerable Program**

```
dillen-delacruz@New:~$ gdb -q ./vulnerable
Reading symbols from ./vulnerable...
(No debugging symbols found in ./vulnerable)
(gdb)
(gdb)
(gdb) run $(python3 -c 'print("A" * 80)')
Starting program: /home/dillen-delacruz/vulnerable $(python3 -c 'print("A" * 80)')
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Buffer: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAA

Program received signal SIGSEGV, Segmentation fault.
0x41414141 in ?? ()
(gdb)
```

9. **Cleanup**

```
dillen-delacruz@New:~$ sudo sysctl -w kernel.randomize_va_space=2
[sudo] password for dillen-delacruz:
kernel.randomize_va_space = 2
dillen-delacruz@New:~$
```