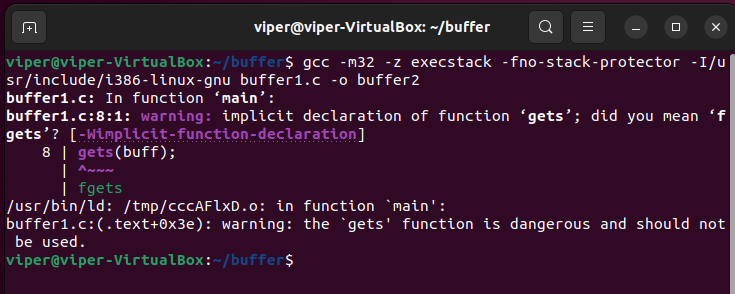
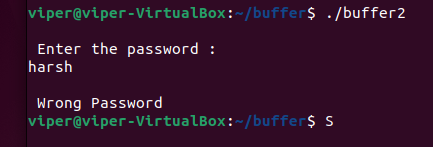
**Buffer Overflow**

**Compilation:**

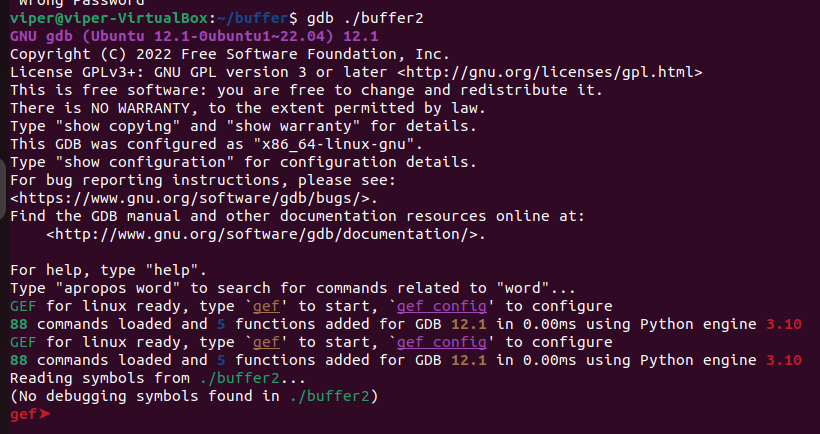




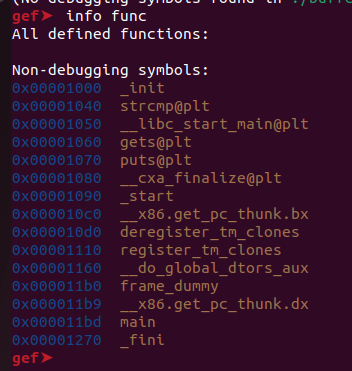
**Execution:**



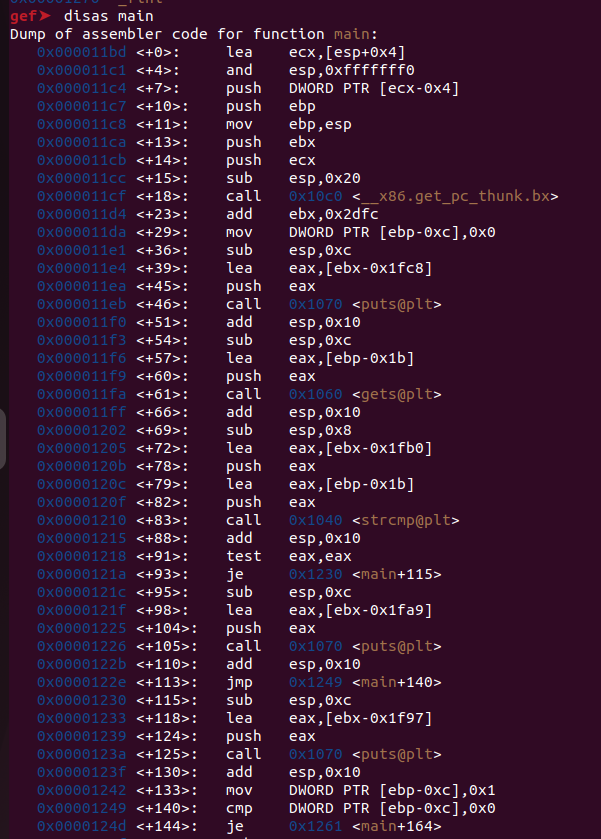
**Opening in GDB:**



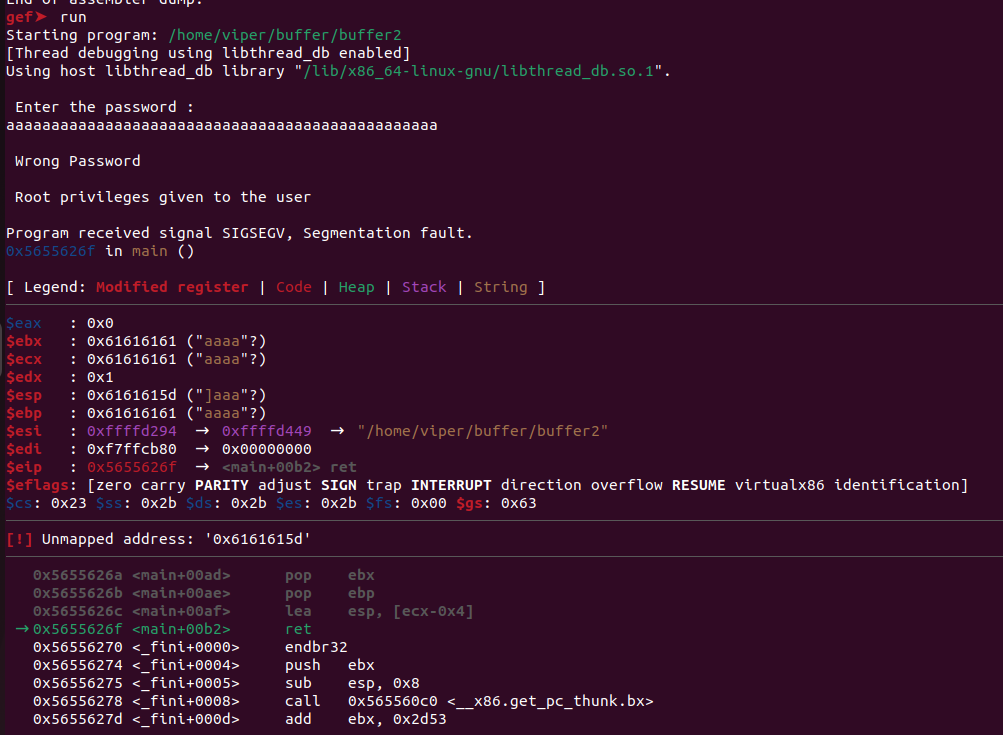
**Info function:**

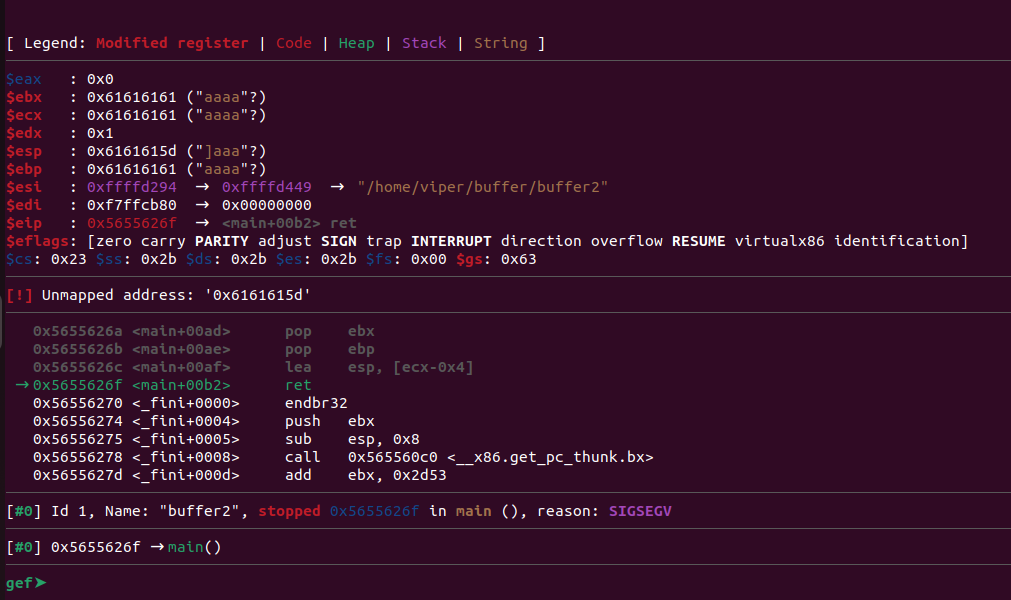


**Disassemble main:**

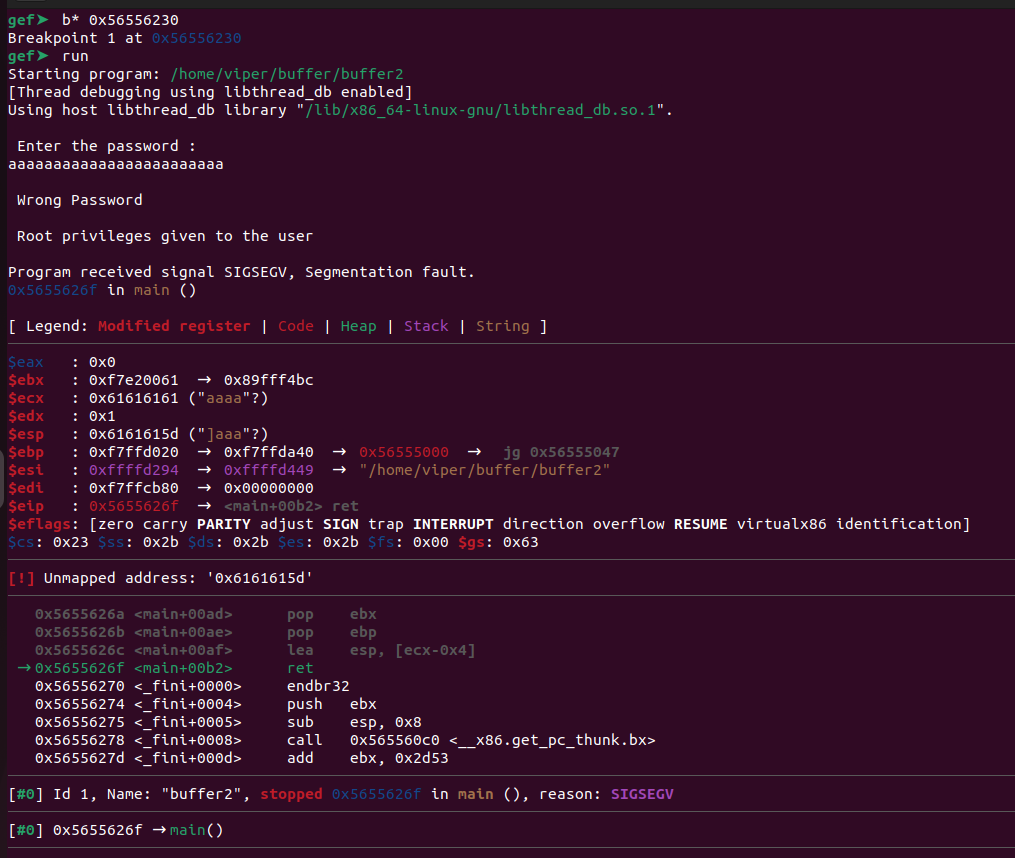


**Run:**

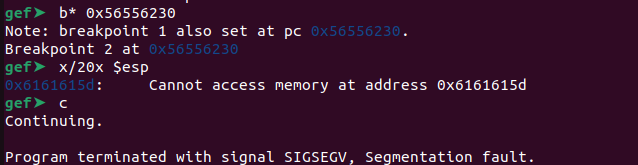




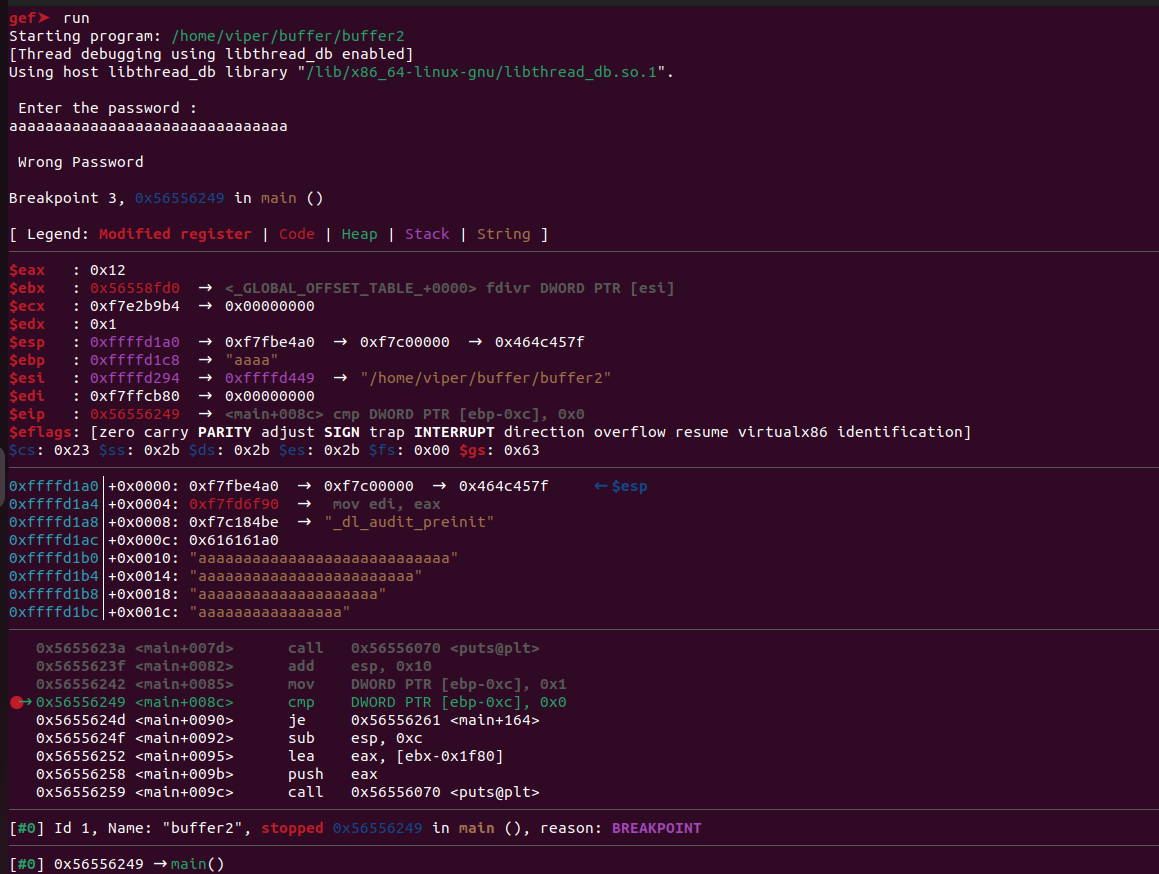
**Create Breakpoint:**

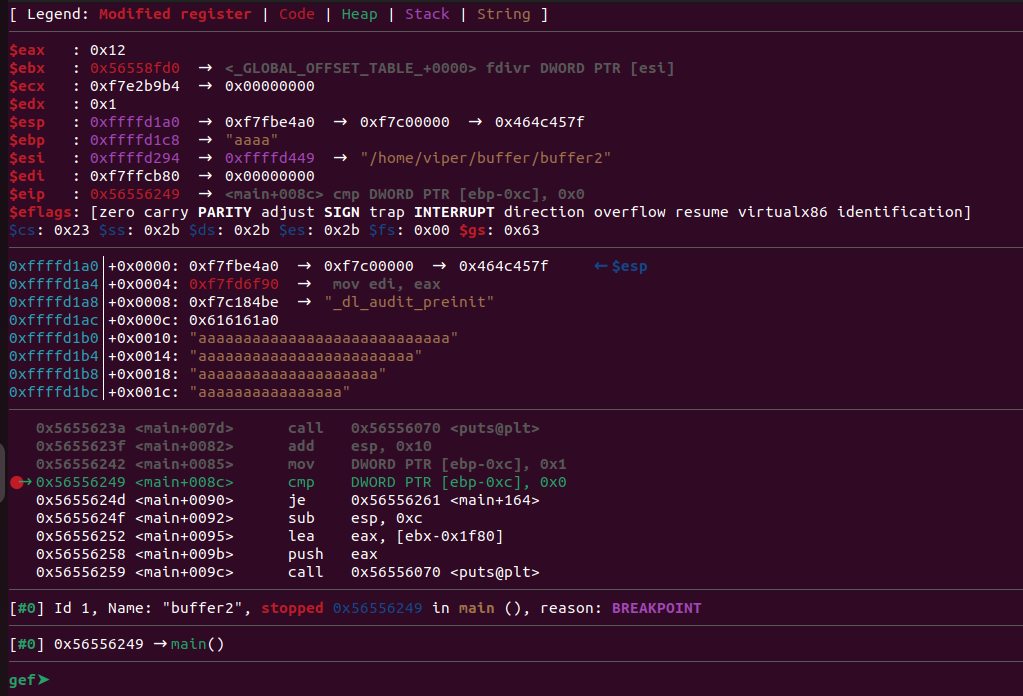


**20 memory locations:**

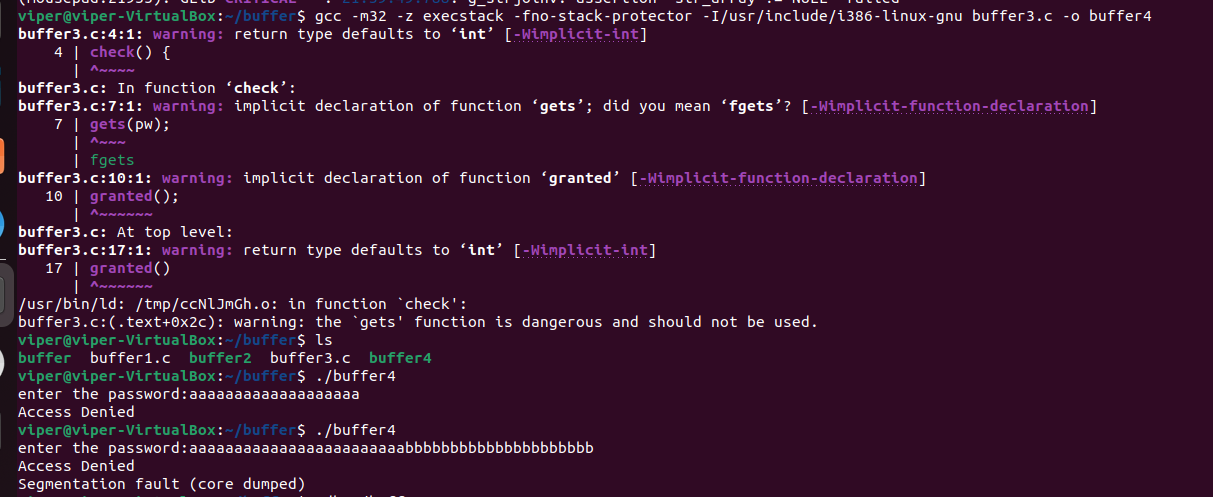


**Run:**

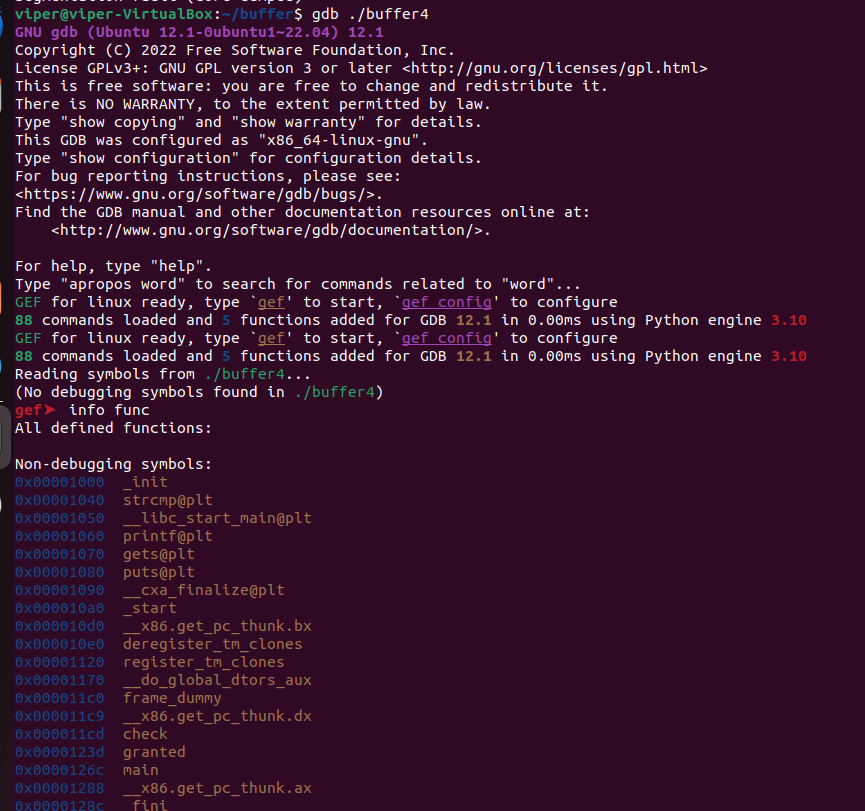




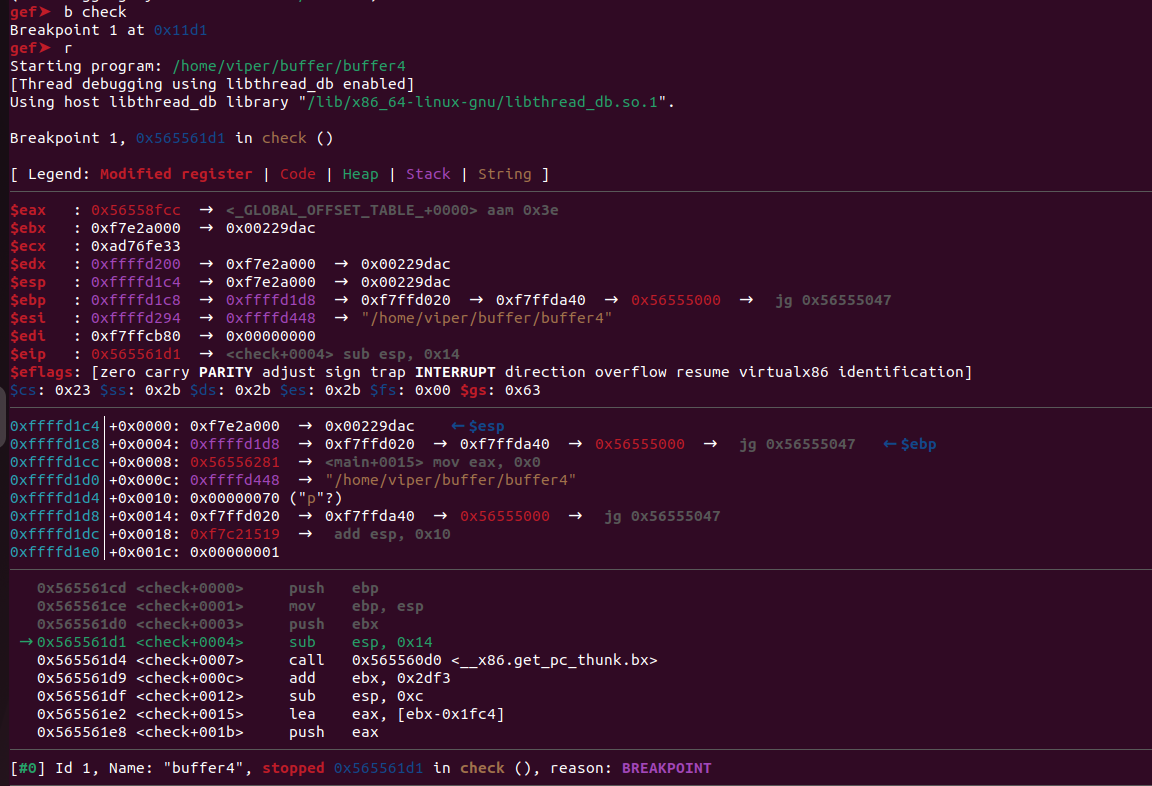
**Second code:**

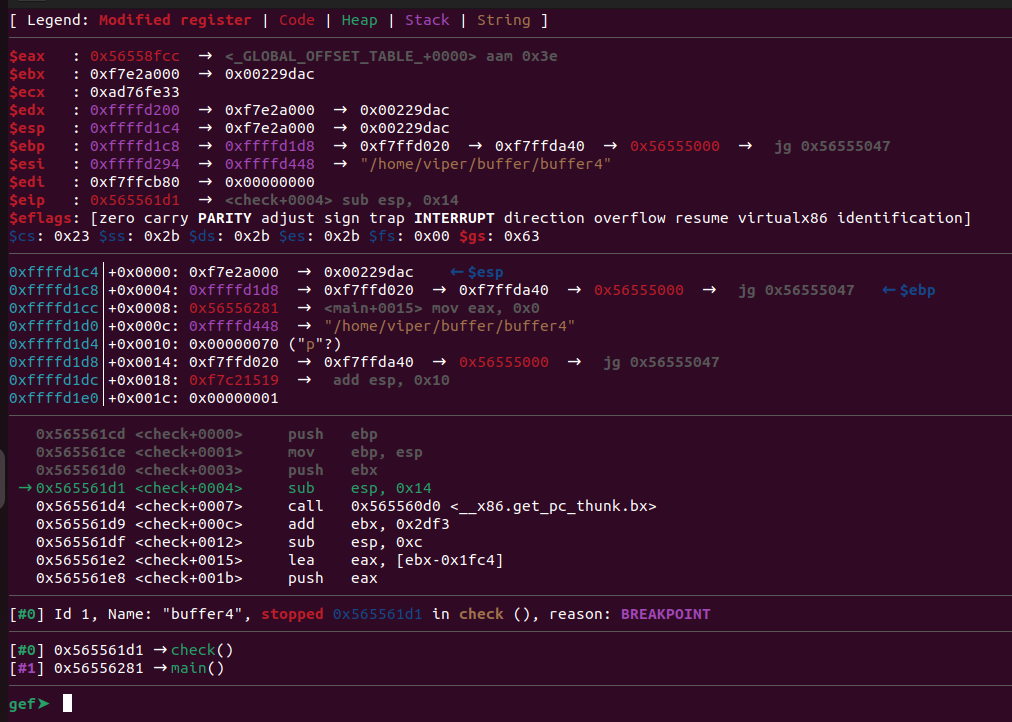


**Execute in GDB:**

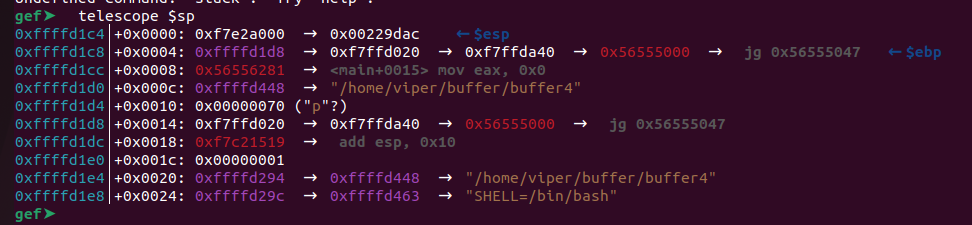


**Breakpoint:**

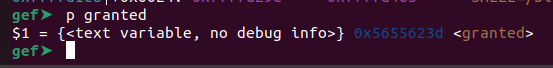


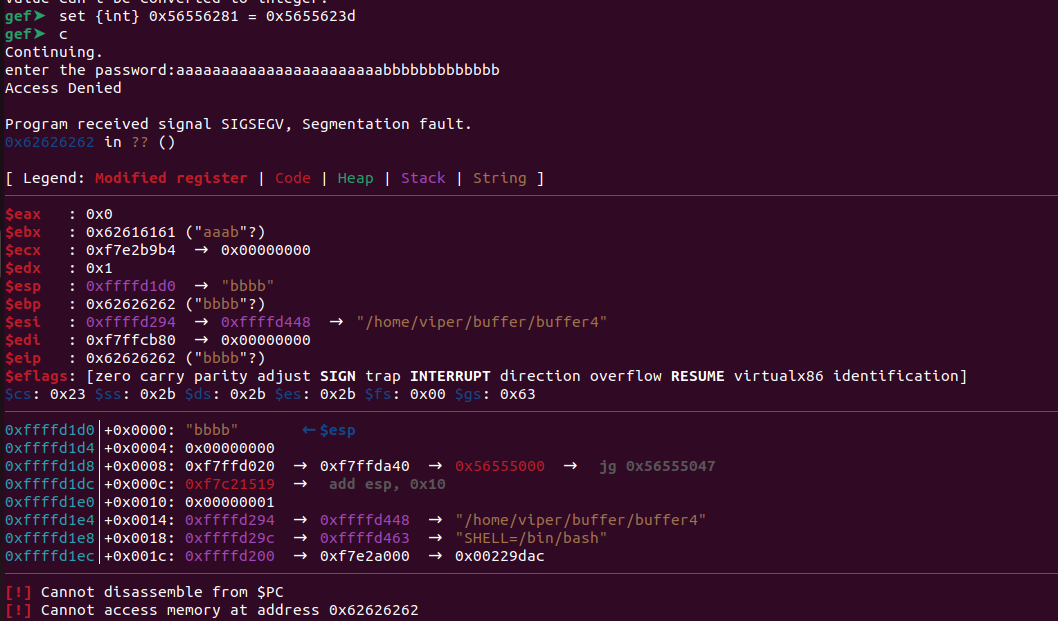


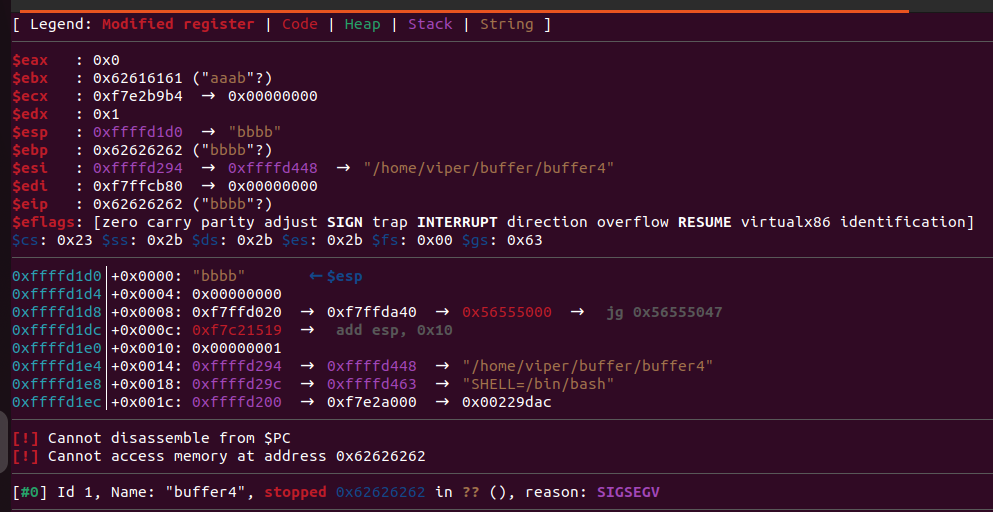
**Examine Stack:**



**Address of Granted function:**







endbr32: This instruction is used in Control Flow Enforcement Technology (CET) to enforce forward-edge control flow integrity. It indicates that the next instruction is a target of a direct call or jump.

lea: This instruction stands for "Load Effective Address". It calculates the effective address of the source operand and stores it in the destination operand.

ecx: This is a general-purpose register in the x86 architecture. It is often used for various purposes, including holding data or memory addresses.

esp: This register, known as the "Stack Pointer", holds the memory address of the top of the stack. It is used for stack operations such as pushing and popping data onto and off the stack.

push: This instruction pushes data onto the stack.

ebp: This register, known as the "Base Pointer", is commonly used as a frame pointer in function prologues and epilogues to access function parameters and local variables.

mov: This instruction is used to move data from one location to another.

ebx: Similar to ecx, ebx is a general-purpose register in the x86 architecture.

sub: This instruction is used to subtract one operand from another.

call: This instruction is used to call a subroutine (function) by transferring control to the specified address.

DWORD PTR: This specifies the size of the memory operand. DWORD stands for "Double Word", which is 32 bits (4 bytes) on x86 systems.

eax: This register, known as the "Accumulator", is used for arithmetic and logical operations. It is also used to store return values from function calls.

add: This instruction is used to add one operand to another.

test: This instruction performs a bitwise AND operation between two operands but does not store the result. It sets the processor flags based on the result.

je: This instruction stands for "Jump if Equal". It transfers control to the specified address if the zero flag is set (indicating that the previous operation resulted in zero).

pop: This instruction removes the top value from the stack and stores it in the specified destination.

ret: This instruction is used to return from a subroutine. It pops the return address from the stack and transfers control to that address.

WHAT IS EIP?

EIP (Extended Instruction Pointer) is a 32-bit register in the x86 architecture that points to the next instruction to be executed by the CPU. In other words, it holds the memory address of the next instruction in the instruction stream.