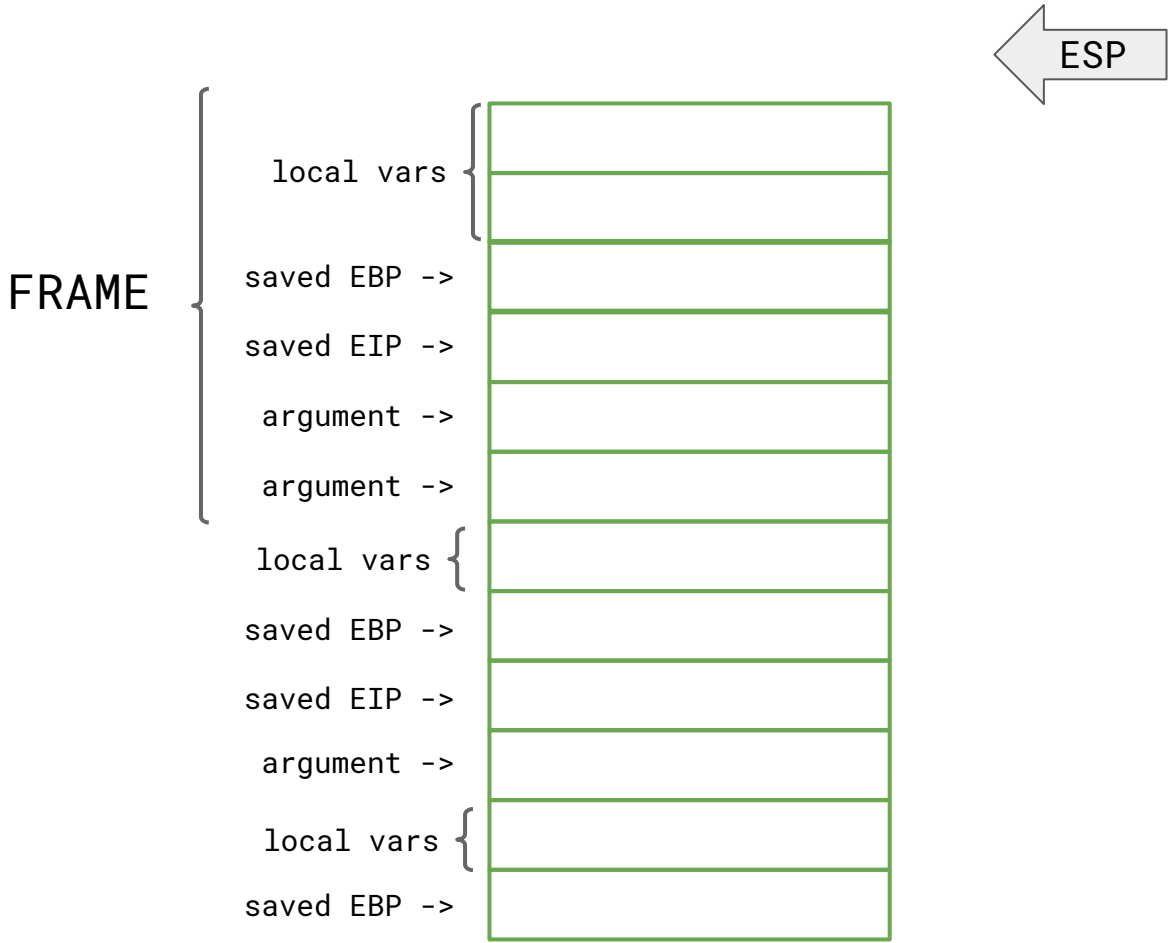
Advanced Cybersecurity Topics

–

Returned Oriented Programming

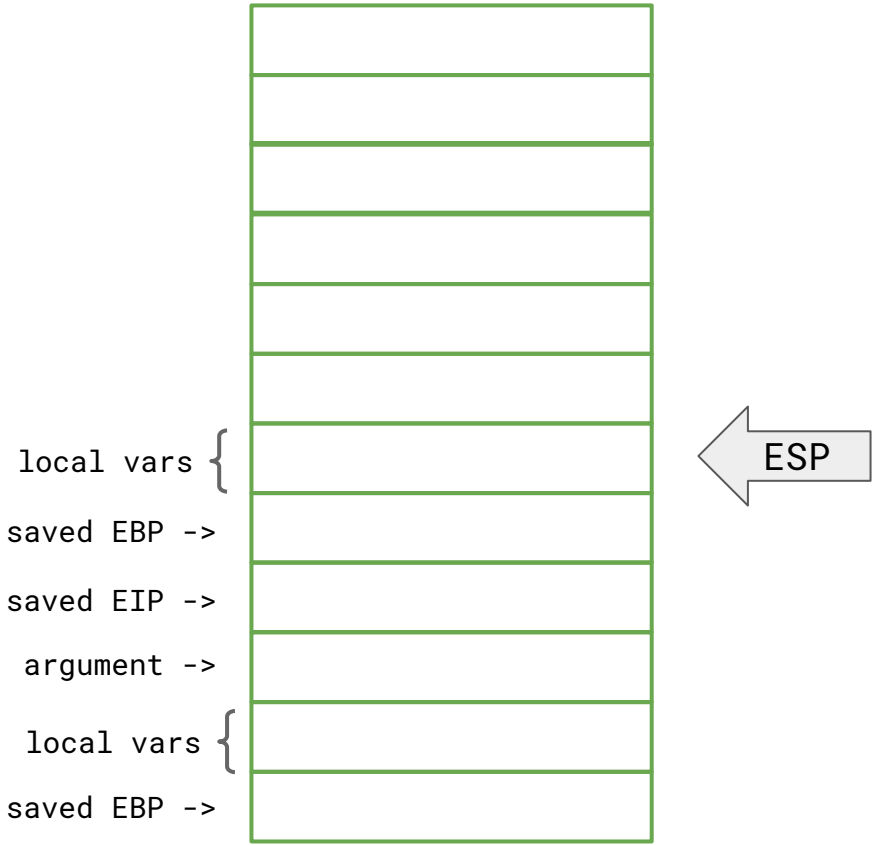
19-20

Call Frame


ESP
FRAME
local vars
saved EBP ->
saved EIP ->
argument ->
argument ->
local vars
saved EBP ->
saved EIP ->
argument ->
local vars
saved EBP ->

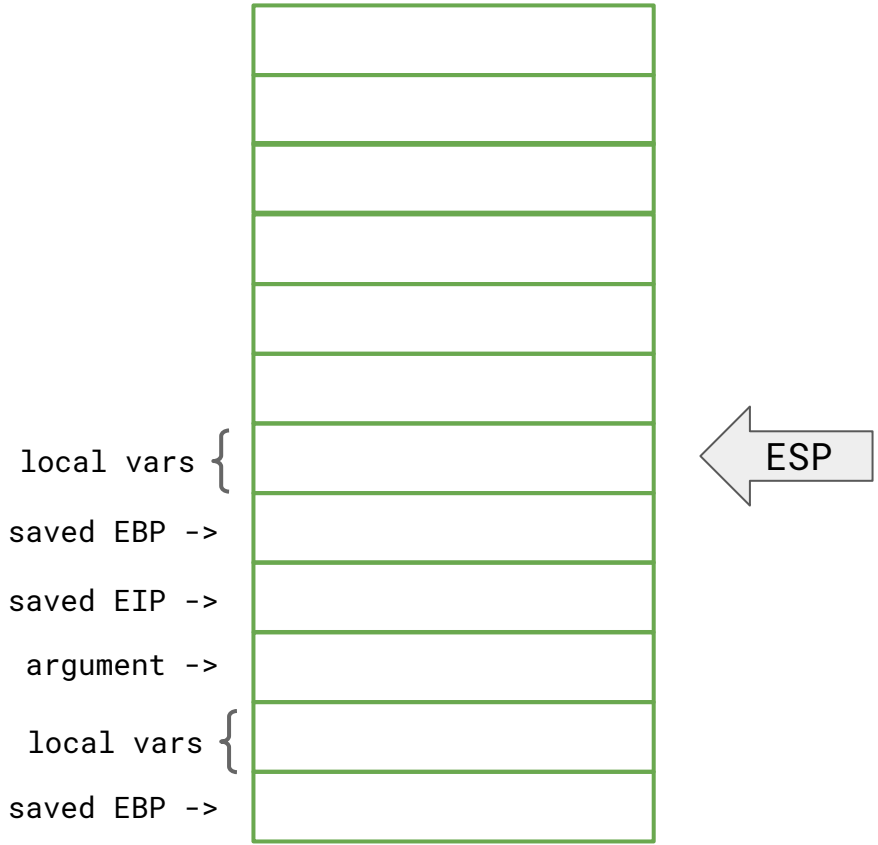
Caller

- PUSH EAX
- PUSH EBX
- CALL
- ADD ESP, 0x8



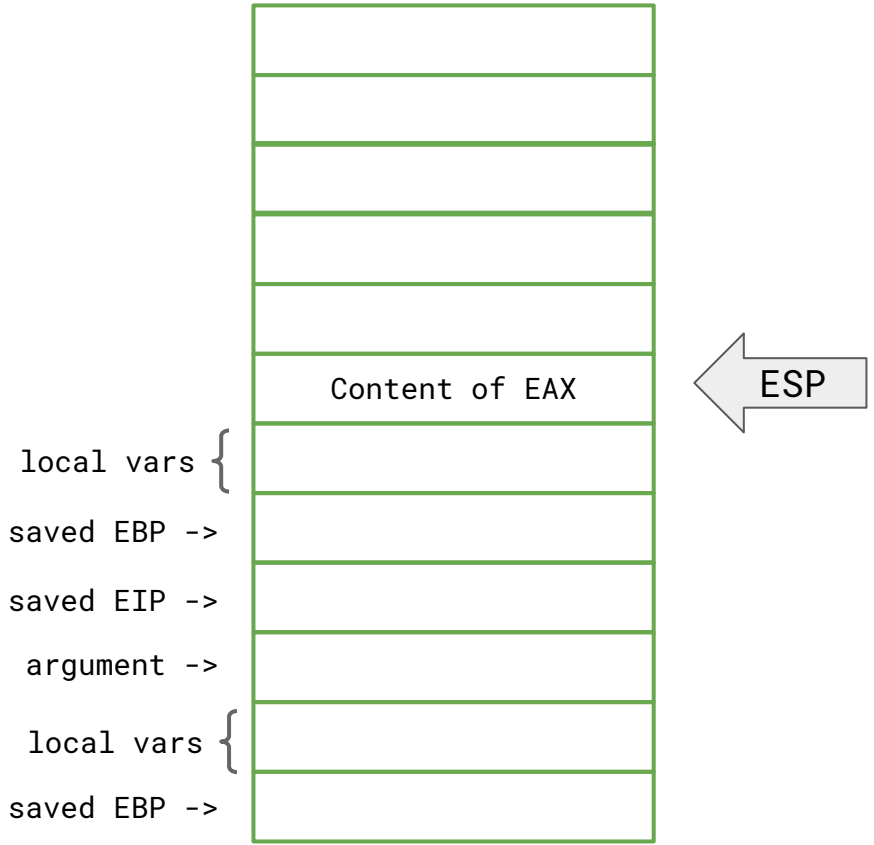
Caller

- **PUSH EAX**
- PUSH EBX
- CALL
- ADD ESP, 0x8



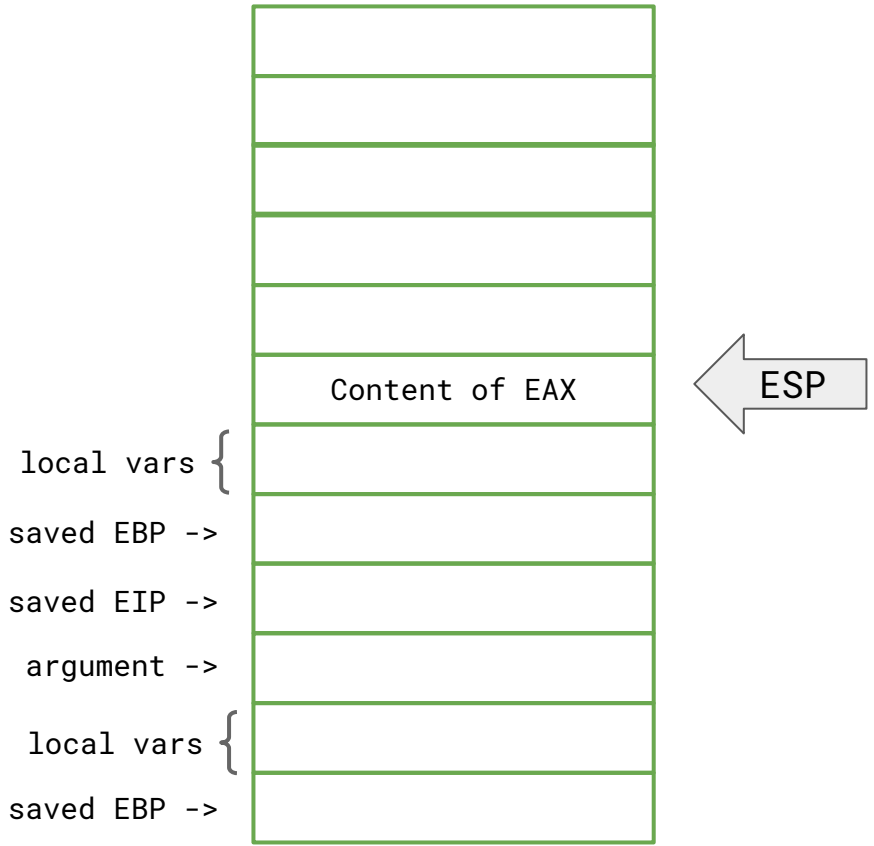
Caller

- **PUSH EAX**
  - SUB ESP, 0x4
  - MOV [ESP], EAX
- PUSH EBX
- CALL
- ADD ESP, 0x8



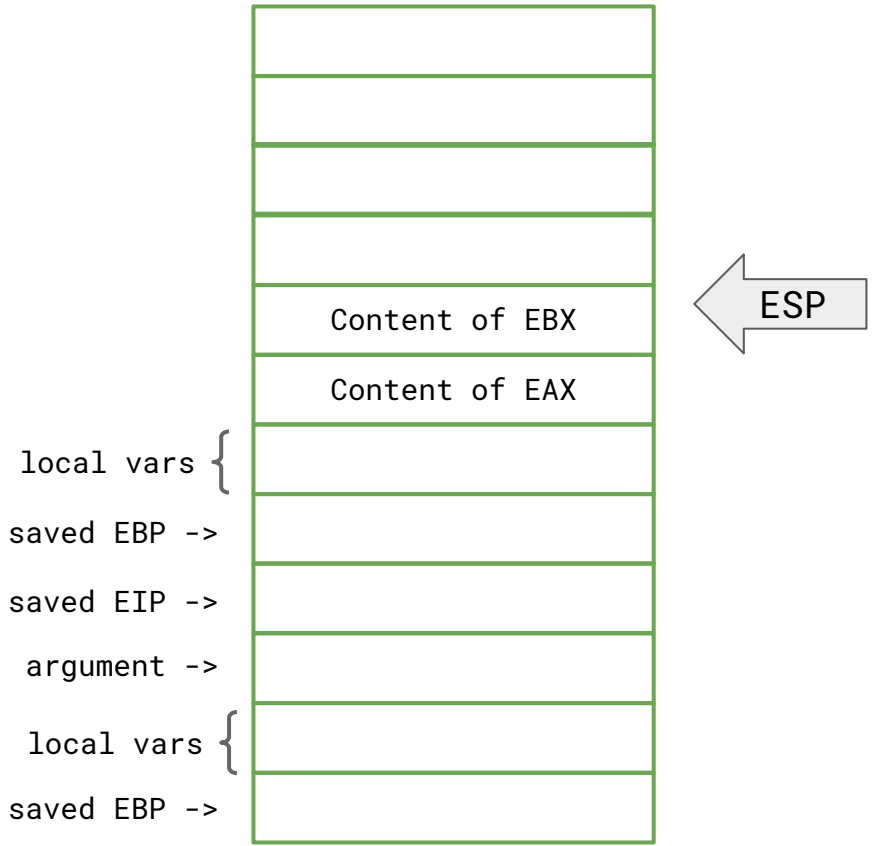
Caller

- PUSH EAX
- **PUSH EBX**
- CALL
- ADD ESP, 0x8



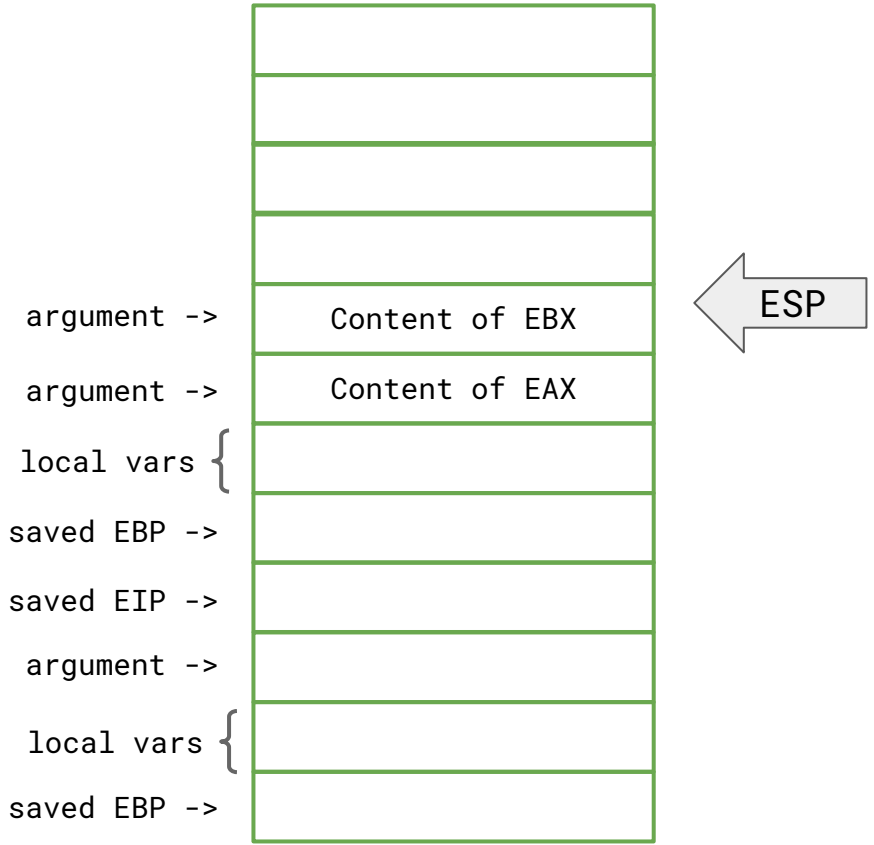
Caller

- PUSH EAX
- **PUSH EBX**
- CALL
- ADD ESP, 0x8



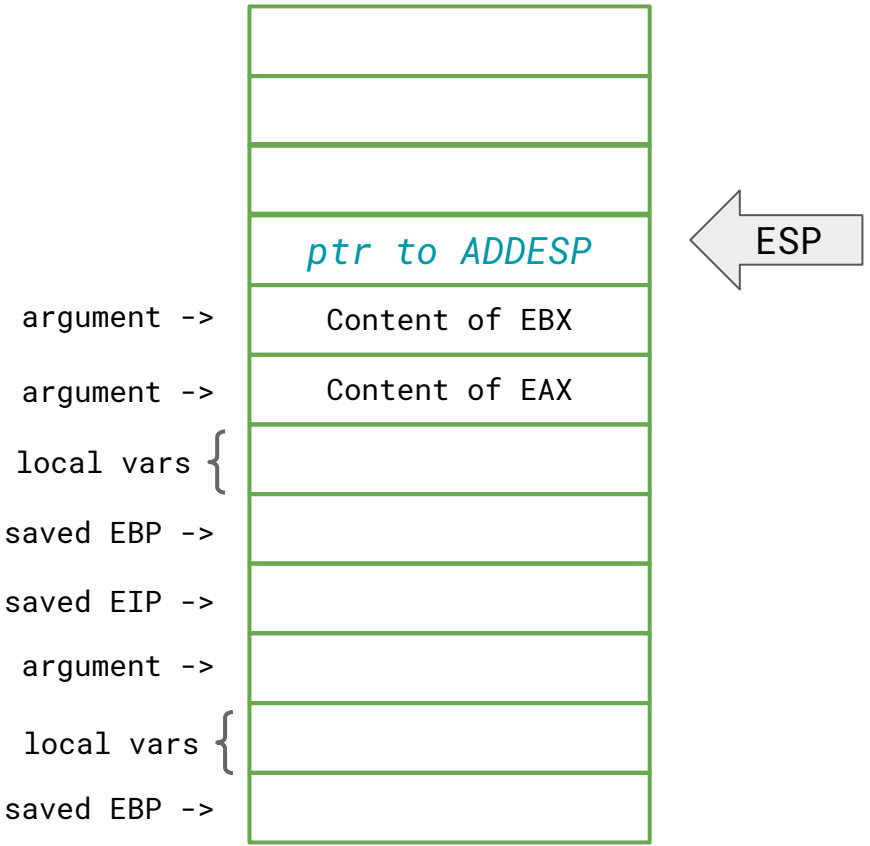
Caller

- PUSH EAX
- **PUSH EBX**
- CALL
- ADD ESP, 0x8



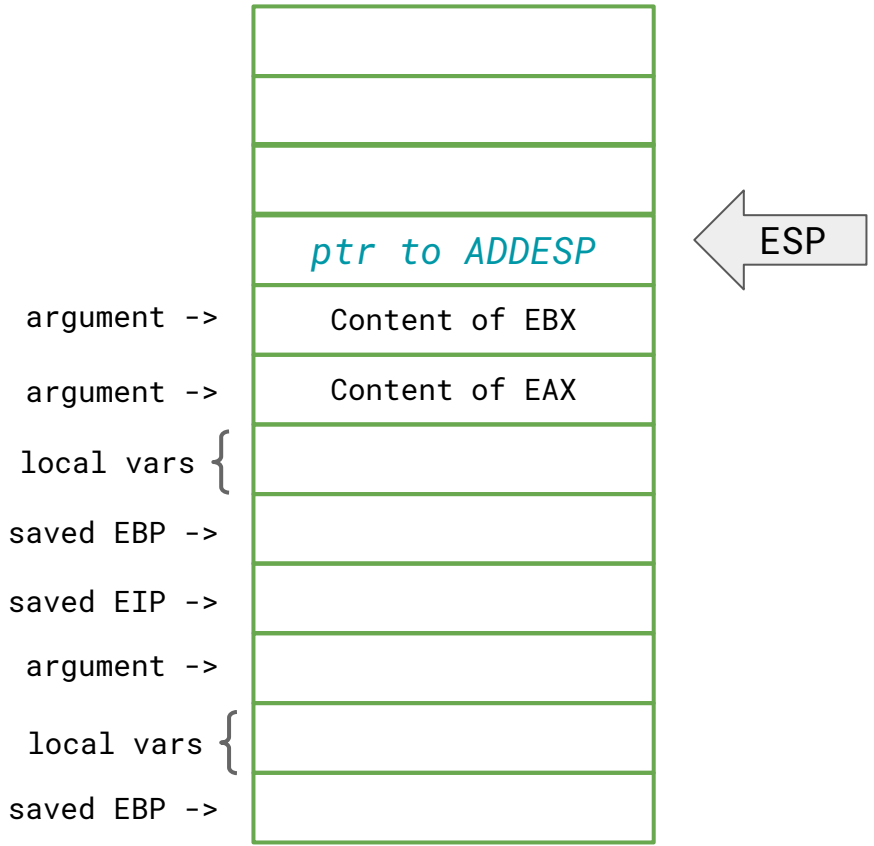
Caller

- PUSH EAX
- PUSH EBX
- **CALL FUN**
  - PUSH EIP
  - JUMP FUN
- ADD ESP, 0x8



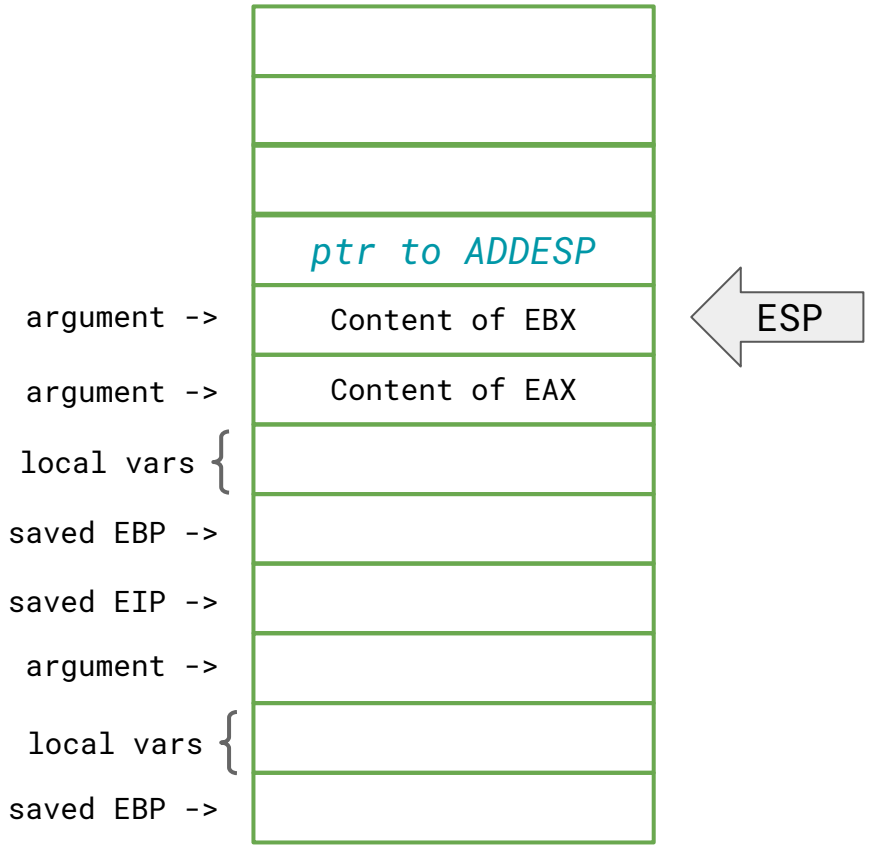
Fun

- RET
  - MOV RIP, [ESP]
  - ADD ESP, 0x4
- After RET the caller clean the stack



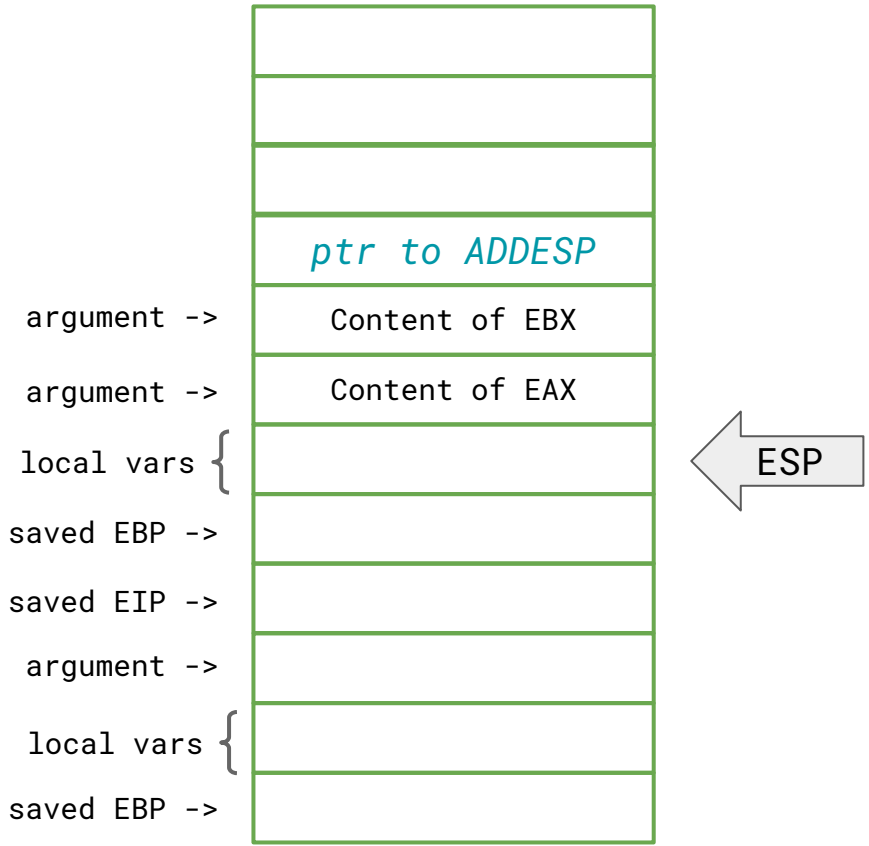
Caller

- PUSH EAX
- PUSH EBX
- CALL FUN
- **ADD ESP, 0x8**



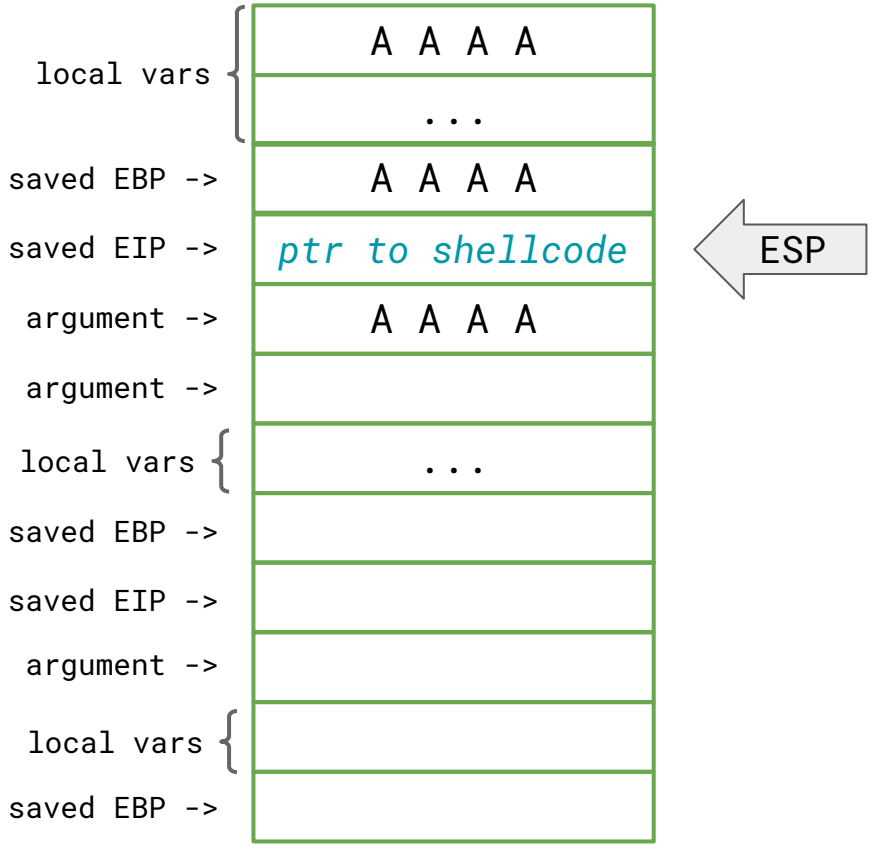
Caller

- PUSH EAX
- PUSH EBX
- CALL FUN
- ADD ESP, 0x8



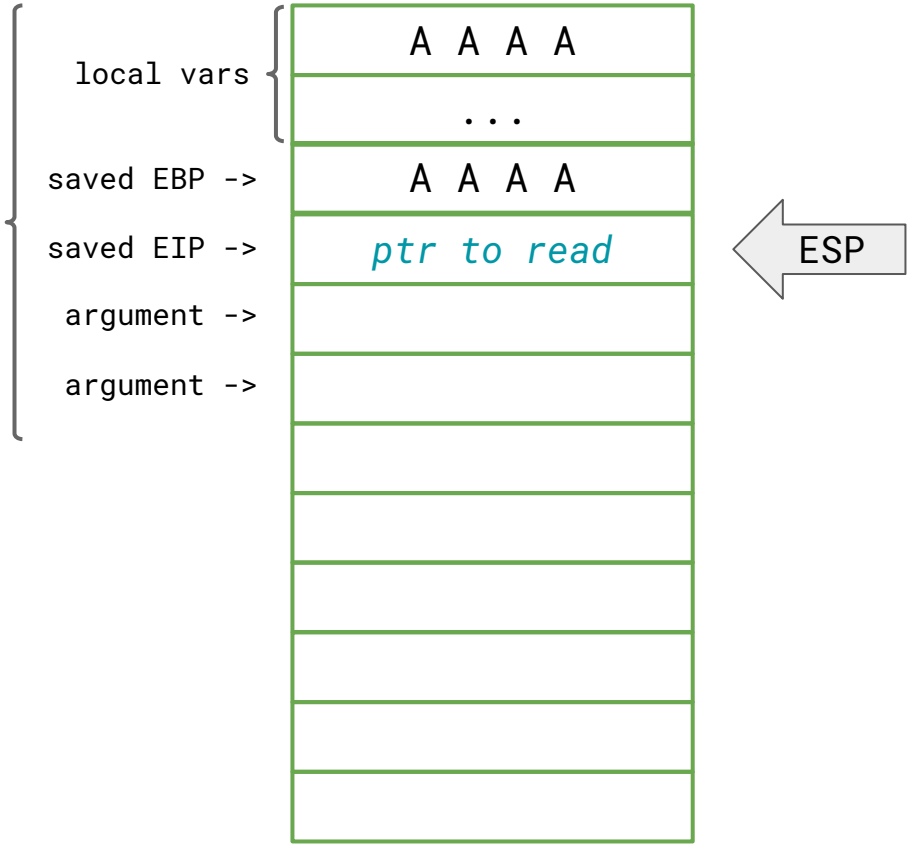
Buffer overflow

- Overwrite EIP and jump to a **shellcode**



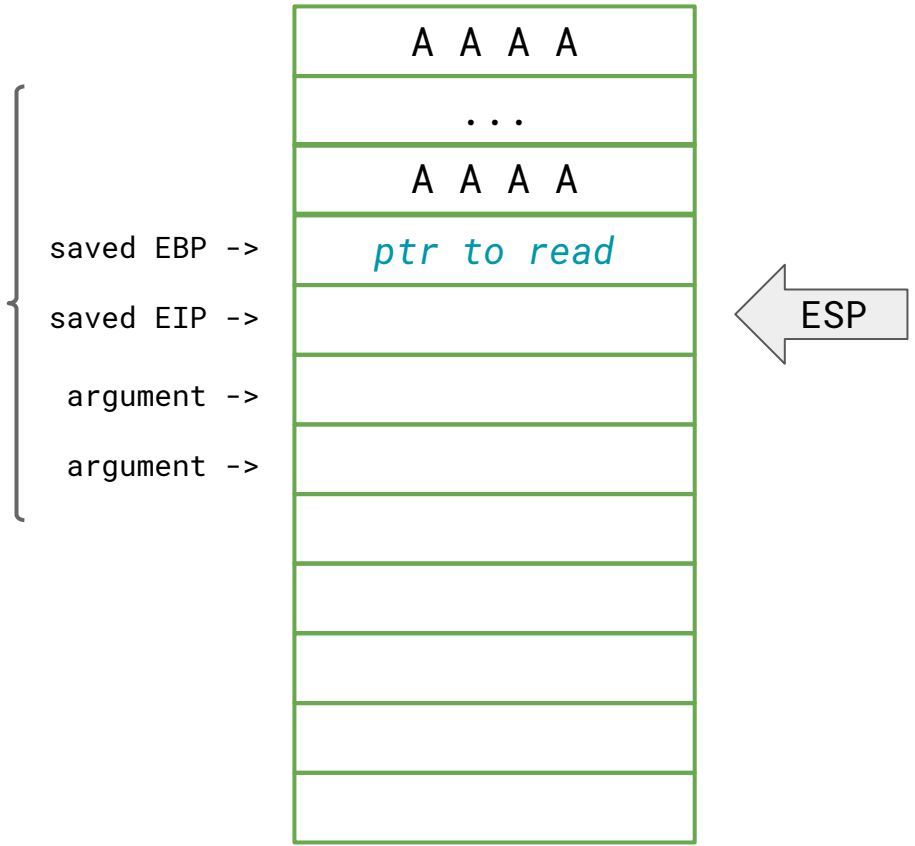
Return to Function (LibC)

- Overwrite EIP and jump to a **function**
- Setup the **arguments**



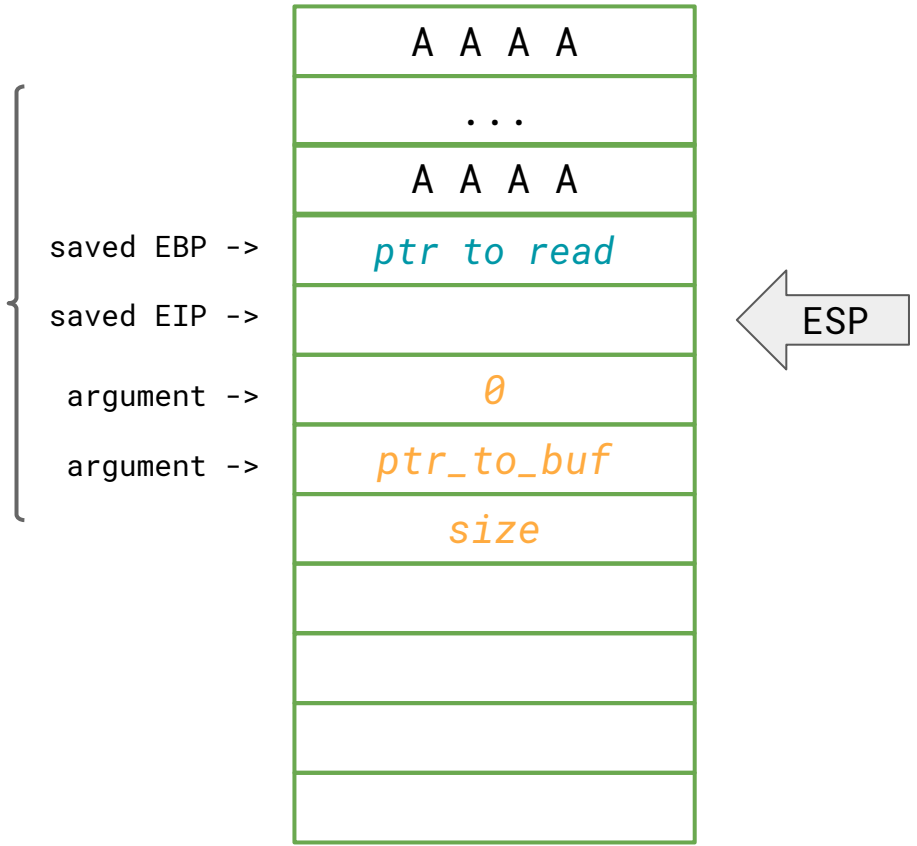
Return to Function (LibC)

- Overwrite EIP and jump to a **function**
- Setup the **arguments**



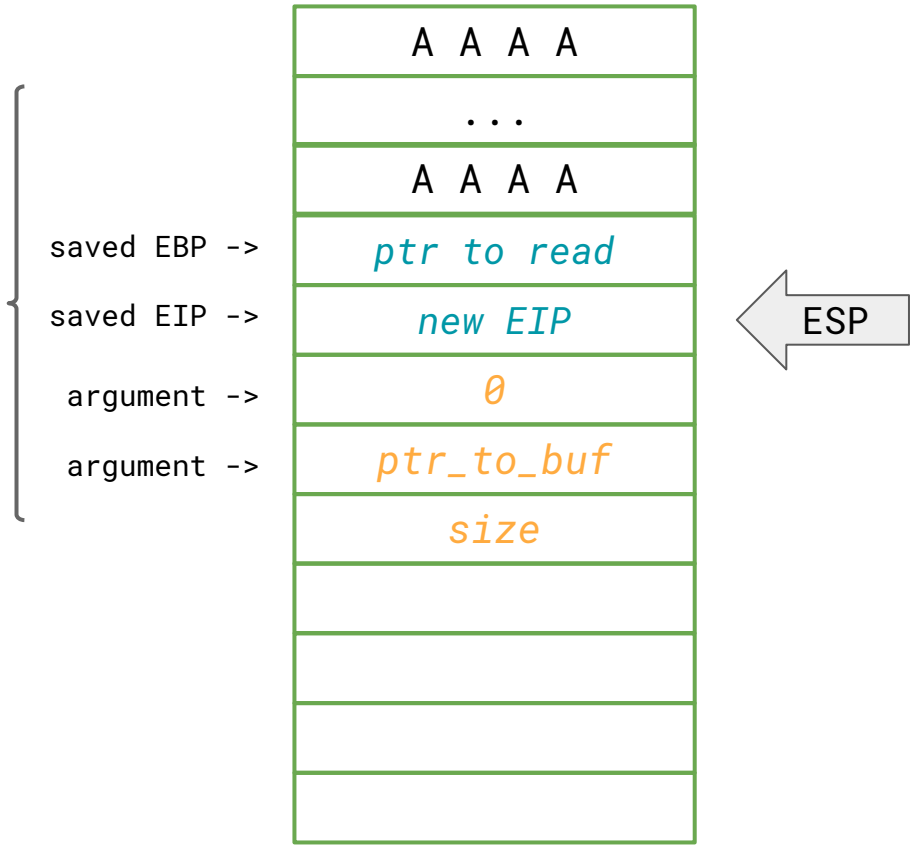
Return to Function (LibC)

- Overwrite EIP and jump to a **function**
- Setup the **arguments**



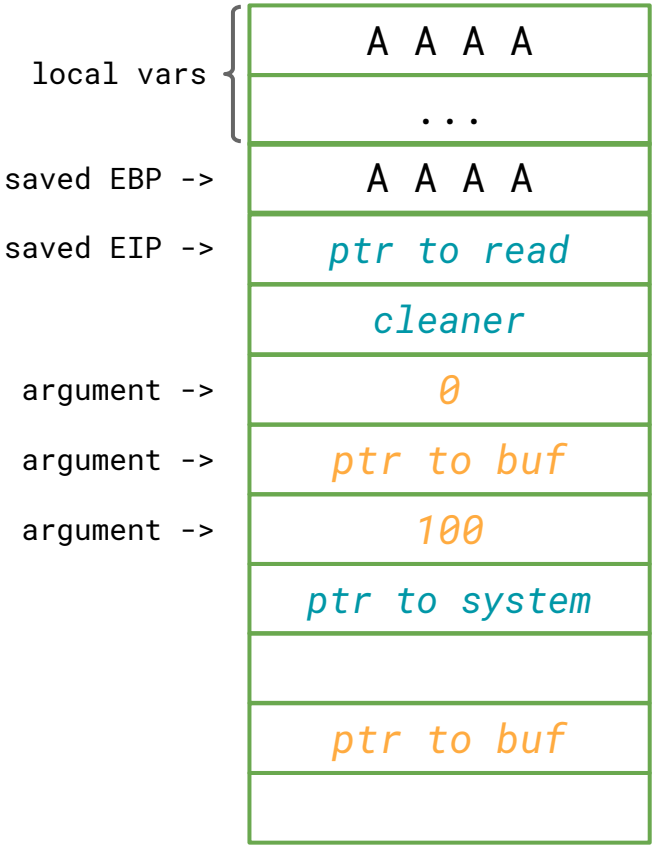
Chain

- Overwrite EIP and jump to a **function**
- Setup the **arguments**
- **Multiple times**



Chain Functions (aka ROP)

- Overwrite EIP and jump to a **function**
- Setup the **arguments**
- **Multiple times**



Gadget

- Instructions followed by a **ret**
- or a **int 0x80**
- or a **syscall**

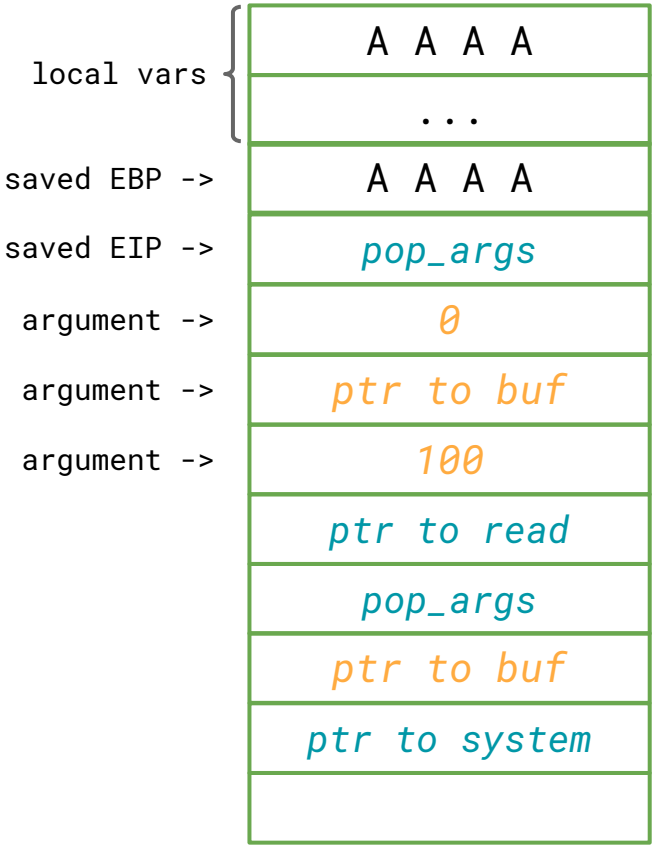
```
pop eax ; ret
```

```
pop ebp ; ret
```

```
pop edx ; pop eax ; ret
```

ROP 64-bit

- Arguments on regs
- Setup the **arguments**
- Cleaner Gadget become Setter



Weird Machines

- Vulnerabilities and **abstractions** create weird machines
  - ROP, etc.
- Writing an Exploit is **Programming** a **Weird Machines**. ~Sergey Bratus