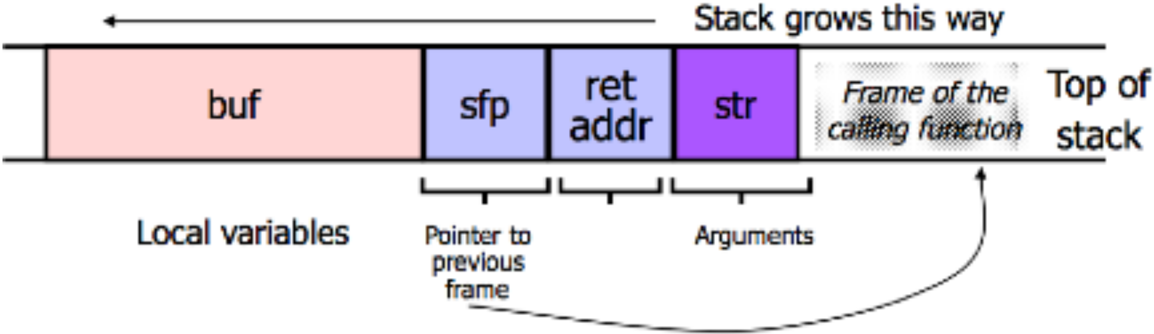
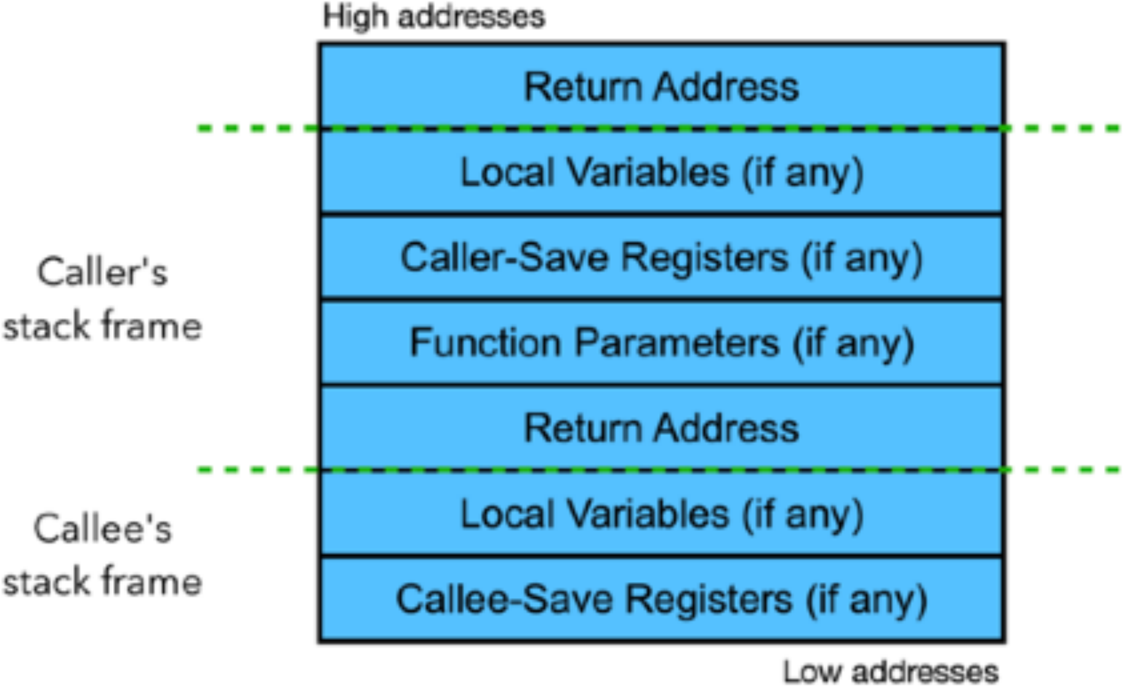
Stack execution

Allocate local buffer (126 bytes in the stack)

Copy argument into local buffer


Stack grows this way
buf
sfp
ret
addr
str
Frame of the
calling function
Top of
stack
Local variables
Pointer to
previous
frame
Arguments

```
void func(char *str){
  char buf[126];
  strcpy(buf,str);
}
```



High addresses

Return Address

Local Variables (if any)

Caller-Save Registers (if any)

Function Parameters (if any)

Return Address

Local Variables (if any)

Callee-Save Registers (if any)

Caller's stack frame

Callee's stack frame

Low addresses

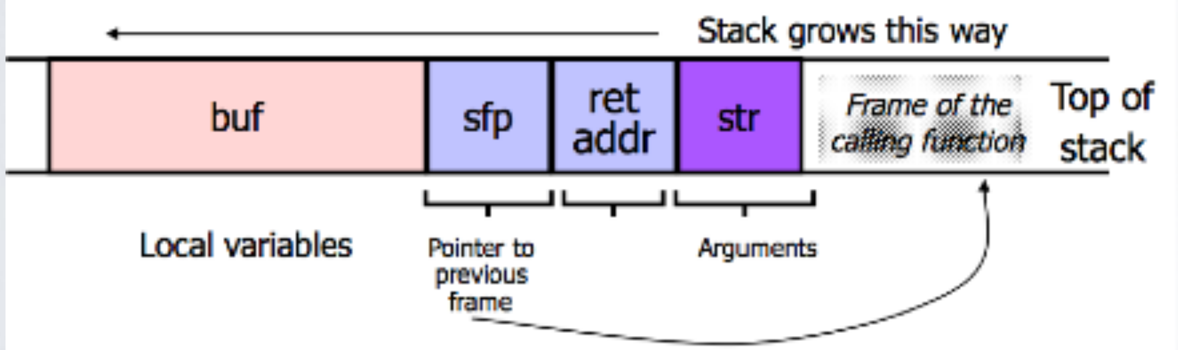
Purposely left out things that differ between compilers, ABIs, or architectures

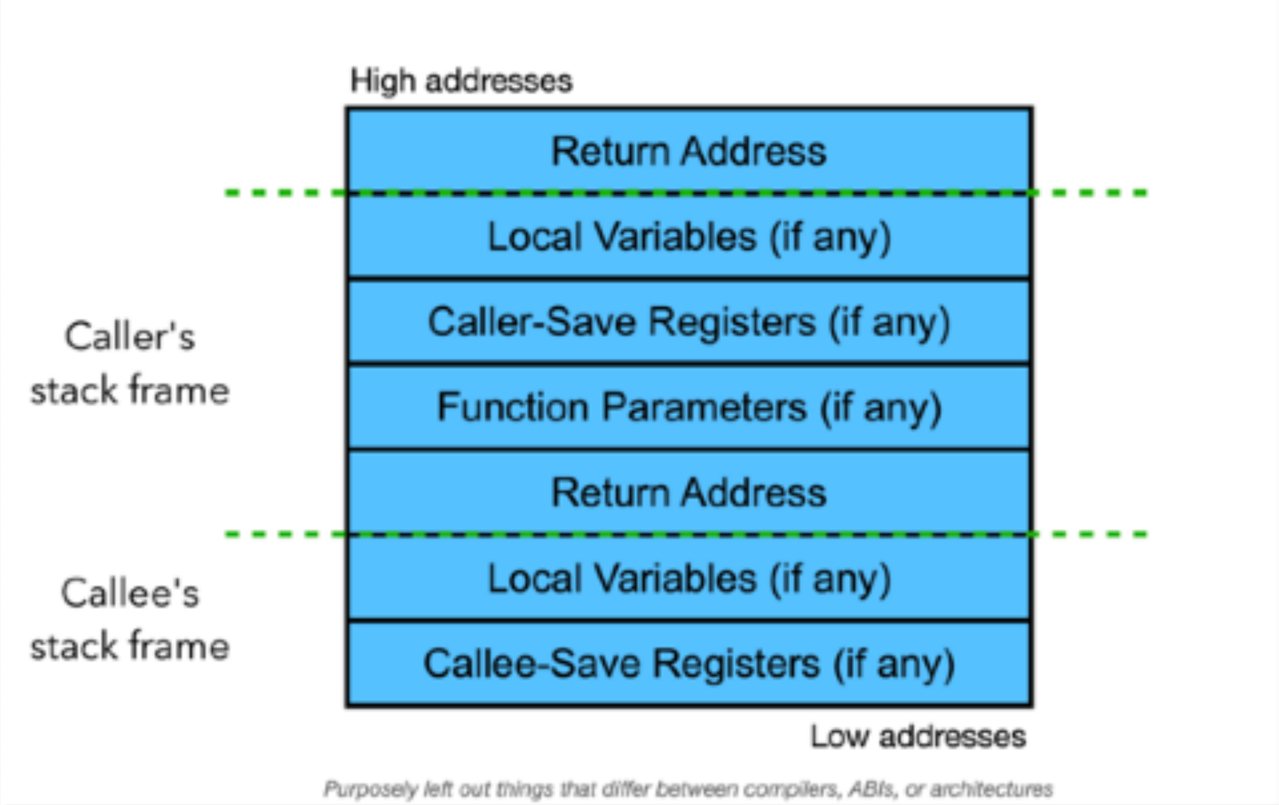
Stack Memory Diagram Conventions, Vuln1001, ost2.fyi, 2022

Stack execution

```
void func(char *str){
  char buf[126];
  strcpy(buf,str);
}
```

Copy argument into local buffer

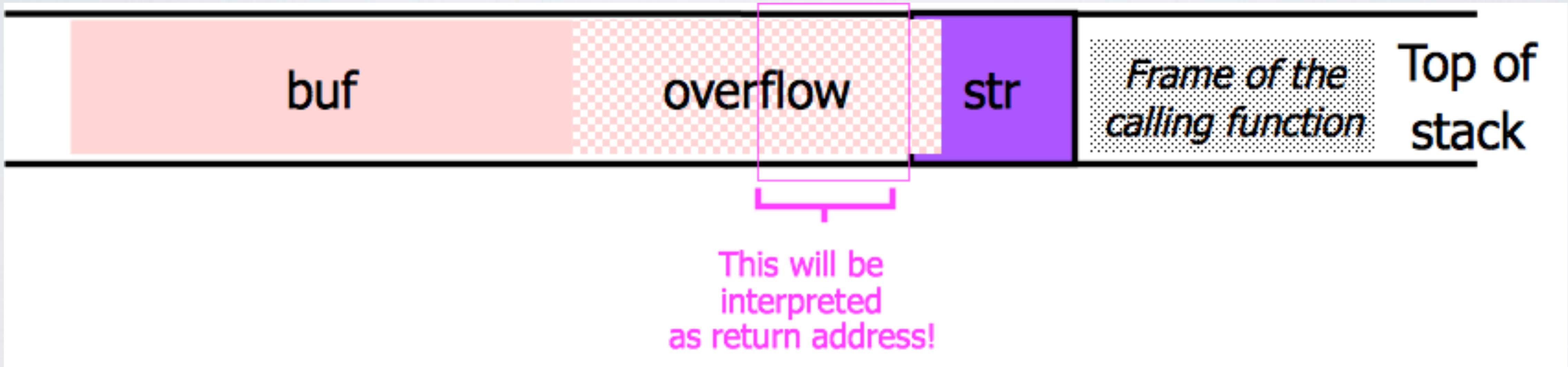



High addresses
Return Address
Local Variables (if any)
Caller-Save Registers (if any)
Function Parameters (if any)
Return Address
Local Variables (if any)
Callee-Save Registers (if any)
Caller's stack frame
Callee's stack frame
Low addresses
Purposely left out things that differ between compilers, ABIs, or architectures

Stack Memory Diagram Conventions, Vuln1001, ost2.fyi, 2022

What if the buffer is overstuffed?

strcpy **does not check** whether the string at *str contains fewer than 126 characters ...


buf
overflow
str
Frame of the calling function
Top of stack
This will be interpreted as return address!