The Friday TA Crew Presents…
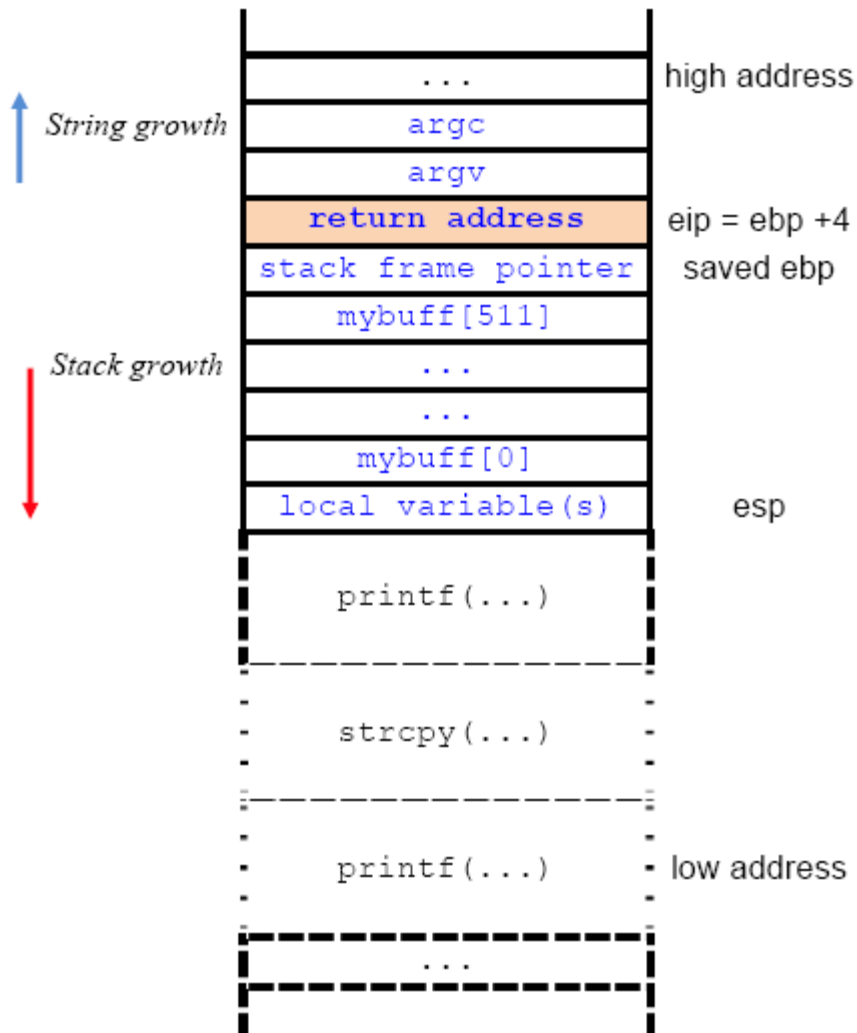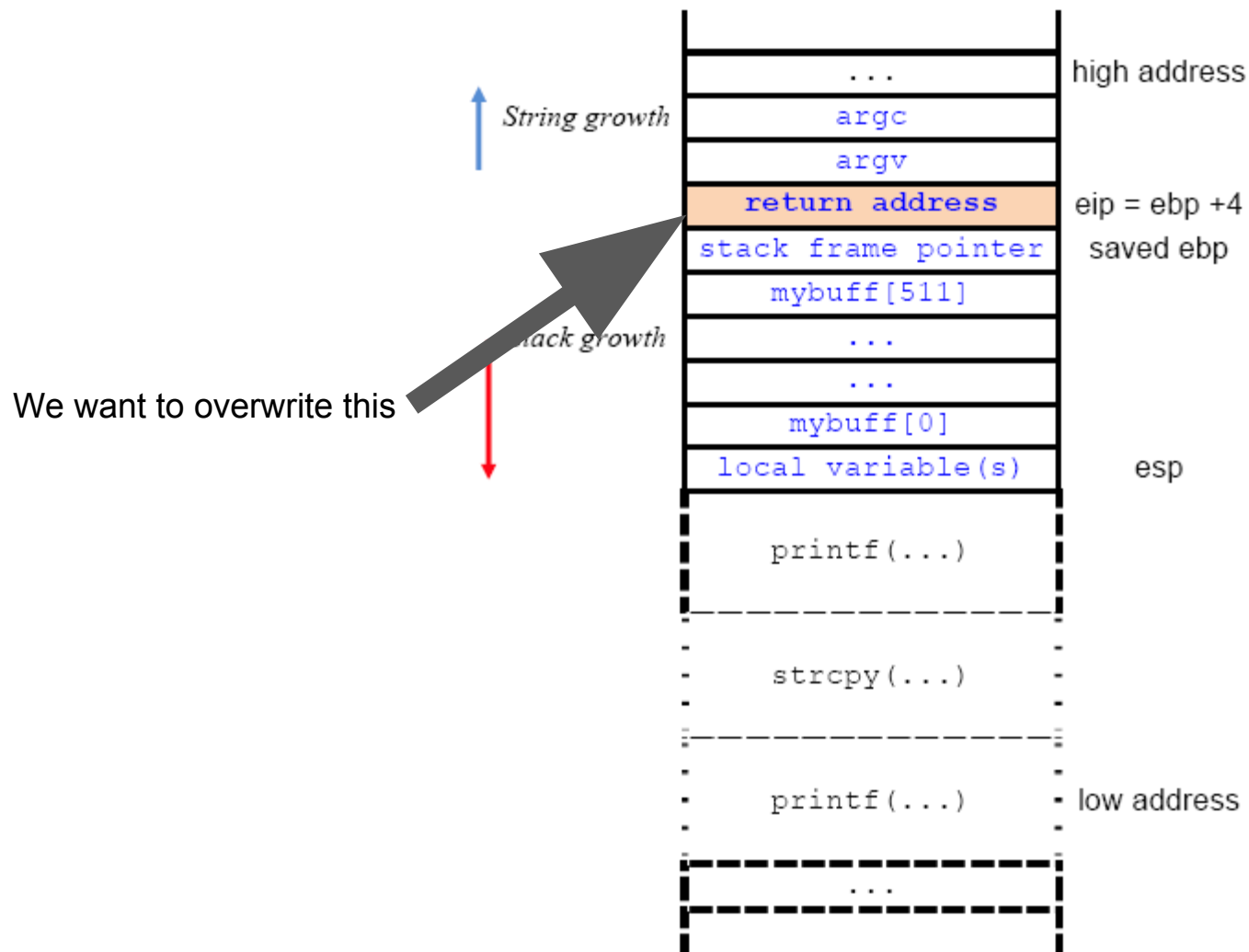
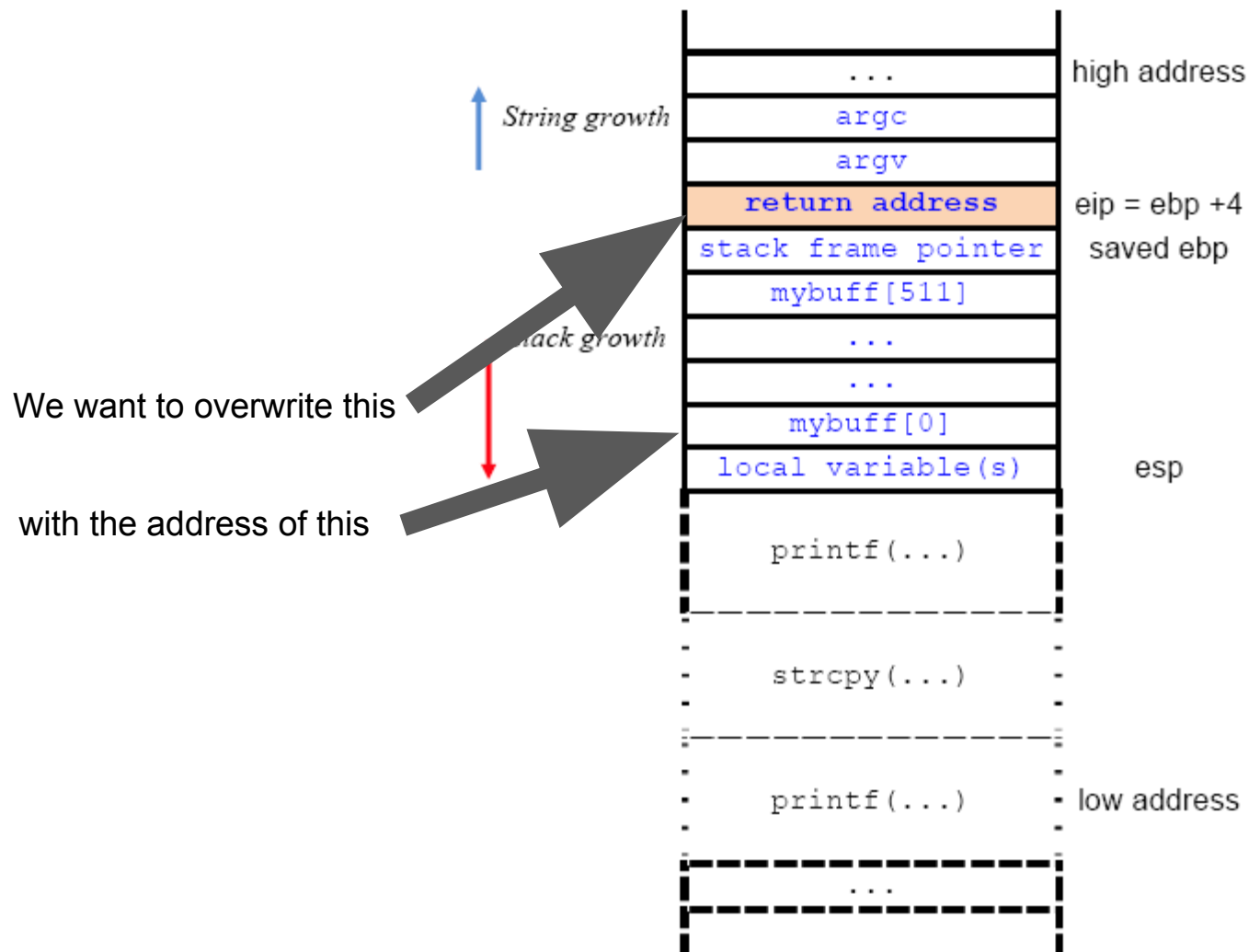# Buffer Part III

What are we trying to achieve?

What are we trying to achieve?

...	high address
argc	
argv	
return address	eip = ebp +4
stack frame pointer	saved ebp
mybuff[511]	
...	
...	
mybuff[0]	
local variable(s)	esp

String growth ↑

Stack growth ↓

We want to overwrite this

```
printf(...)

strcpy(...)

printf(...)          low address

...
```

...	high address
argc	
argv	
return address	eip = ebp +4
stack frame pointer	saved ebp
mybuff[511]	
...	
...	
mybuff[0]	
local variable(s)	esp

String growth

stack growth

We want to overwrite this

with the address of this

printf(...)

strcpy(...)

printf(...) — low address

...

String growth

high address

...

argc

argv

return address — eip = ebp +4

stack frame pointer — saved ebp

mybuff[511]

Stack growth

...

...

mybuff[0]

local variable(s) — esp

printf(...)

strcpy(...)

printf(...)

...

We want to overwrite this

with the address of this

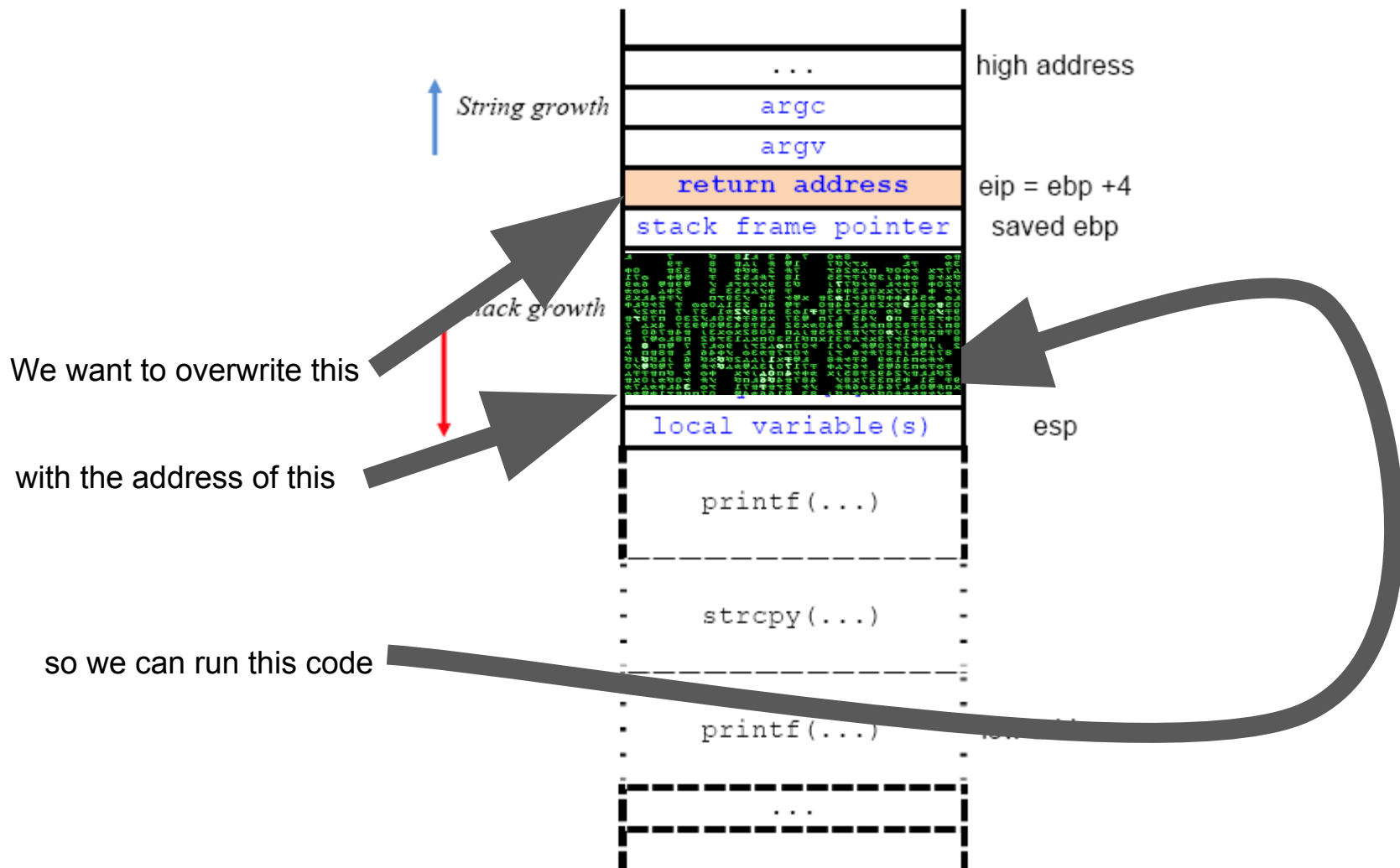so we can run this code

So what tools do we have to achieve this?

GDB

Useful GDB Tools:

- x [address]
  - Try running `x $rsp` to find the address of rsp!
  - x/[number] $rsp (i.e. `x/64 $rsp`).
- info reg
  - get the information of the registers
- disass
  - disassembles
- si
  - step instruction
- info proc mapping
  - gives you a mapping of the memory