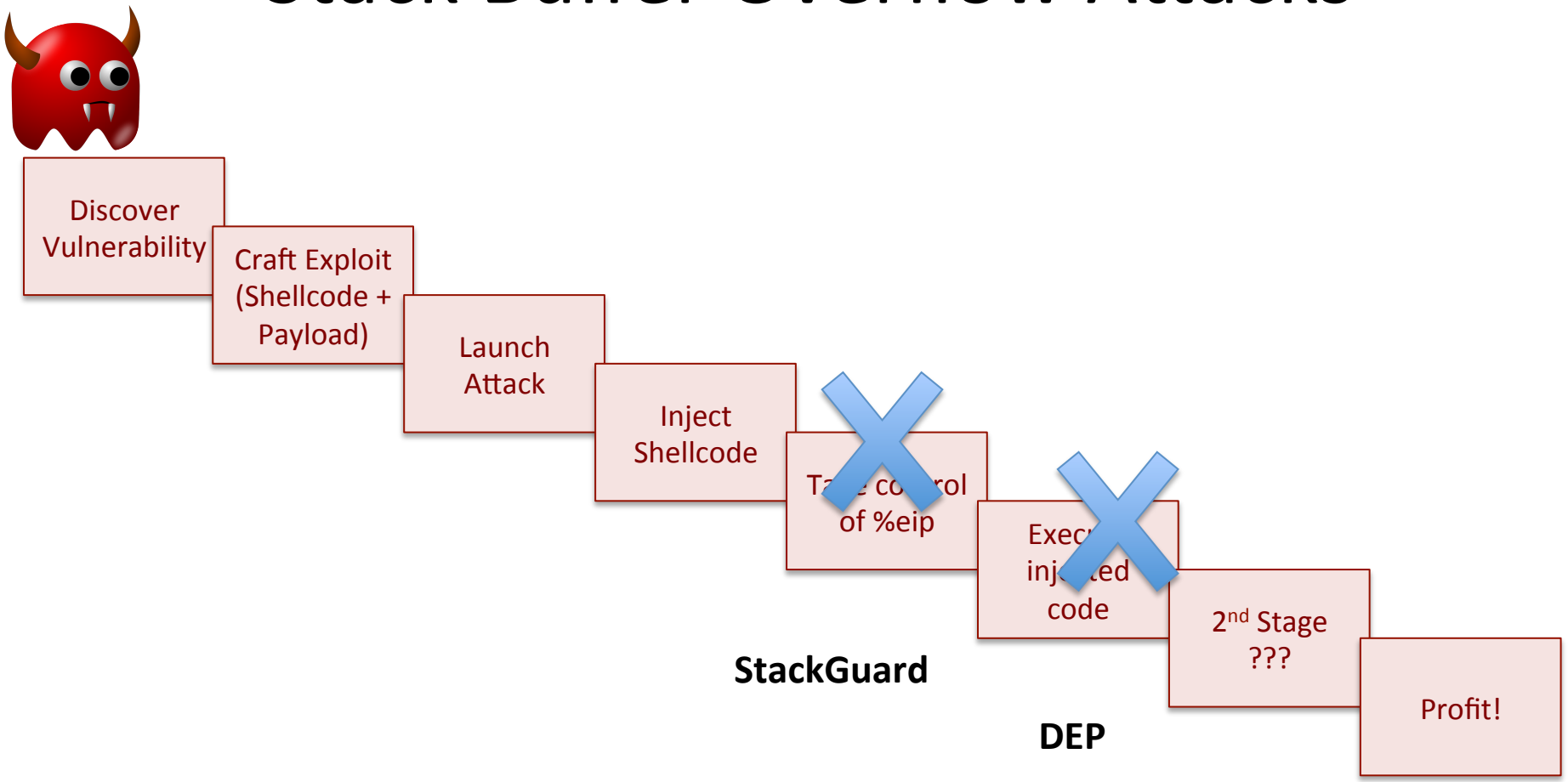
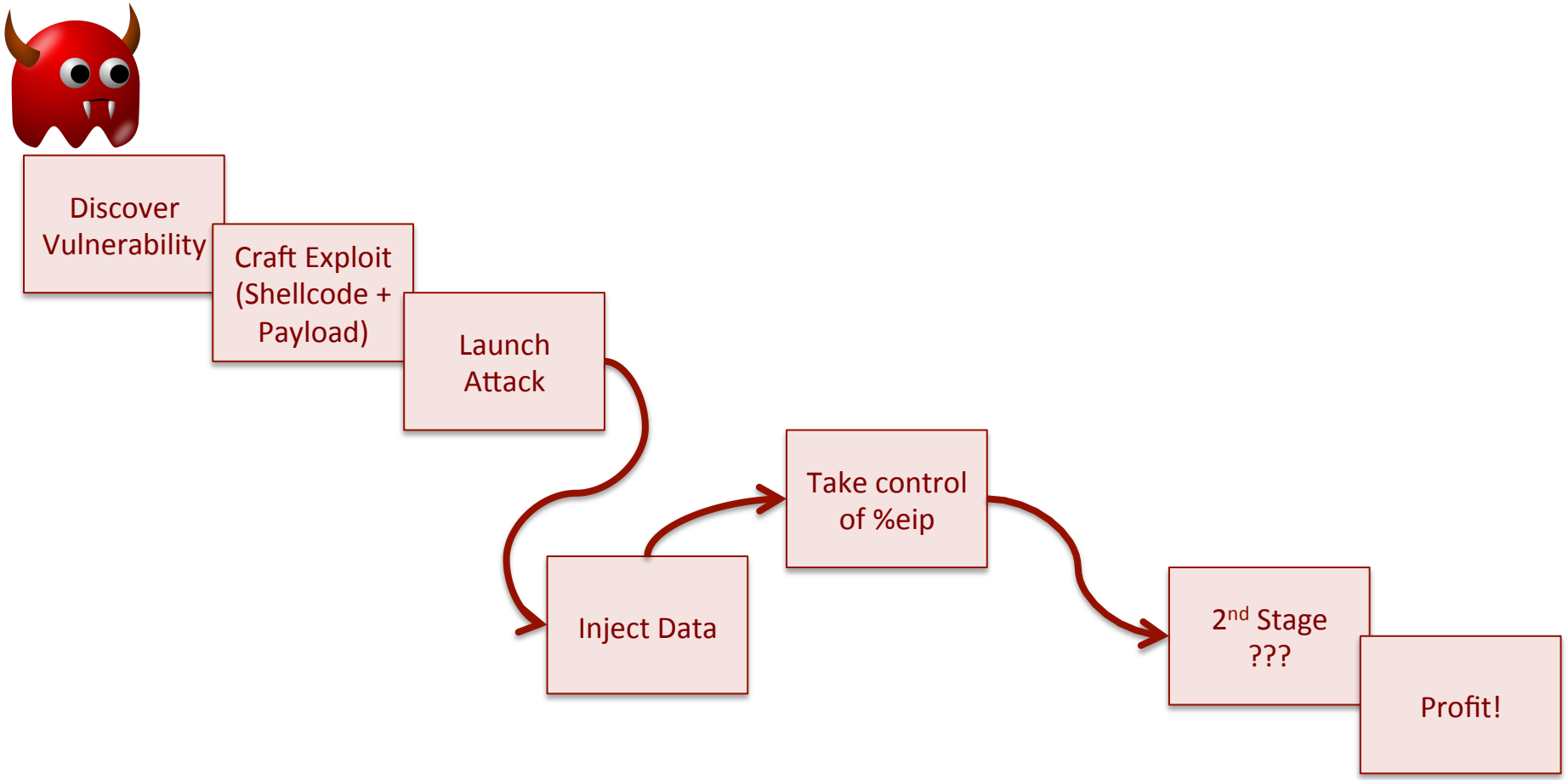
# RETURN-ORIENTED PROGRAMMING

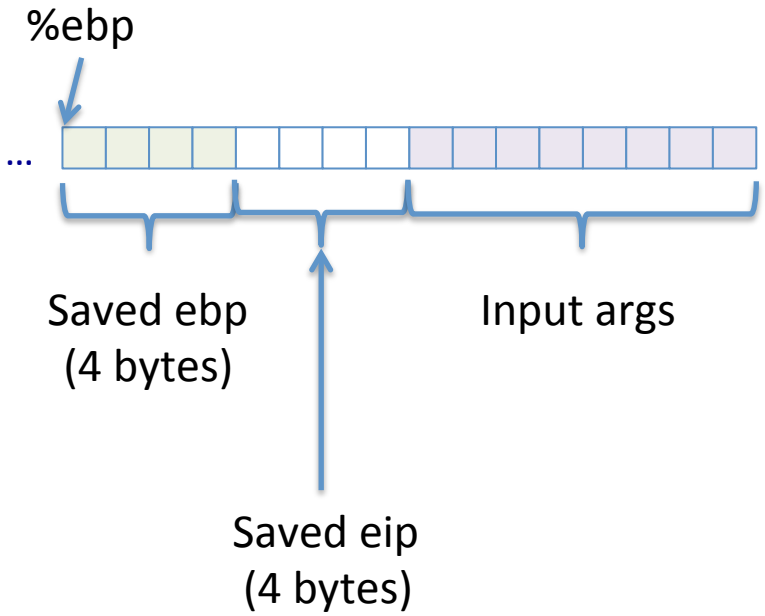
# The “Kill Chain” for Stack Buffer Overflow Attacks



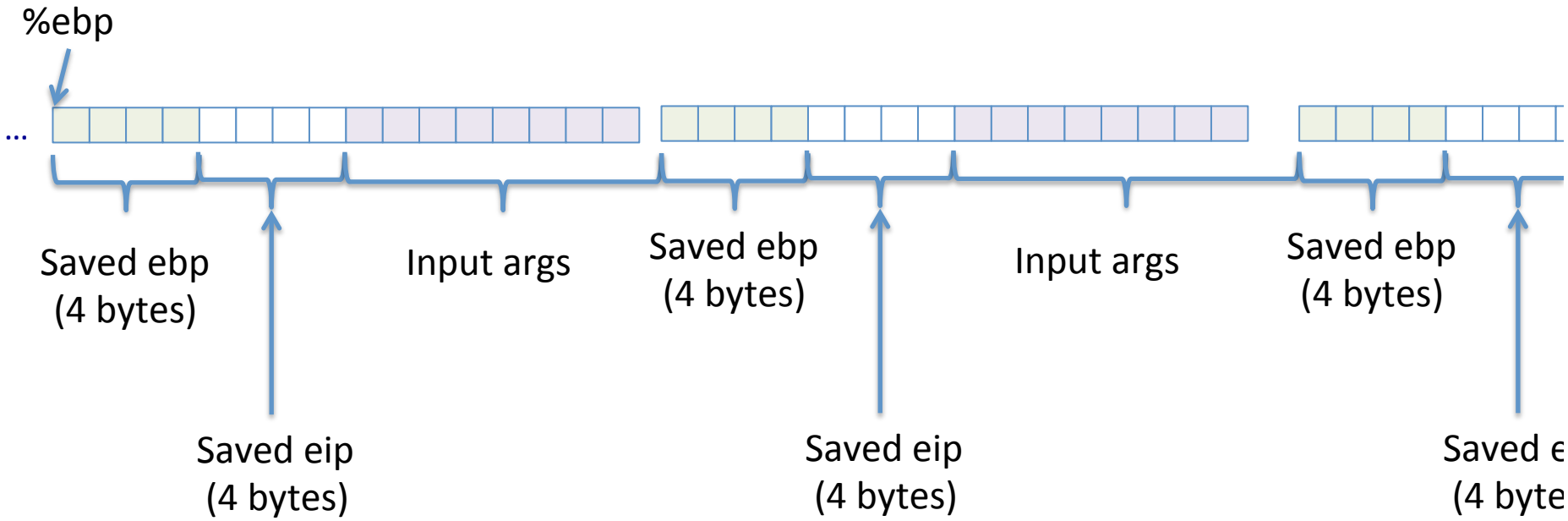
# The “Kill Chain”: ROP



# What's after our stack frame?

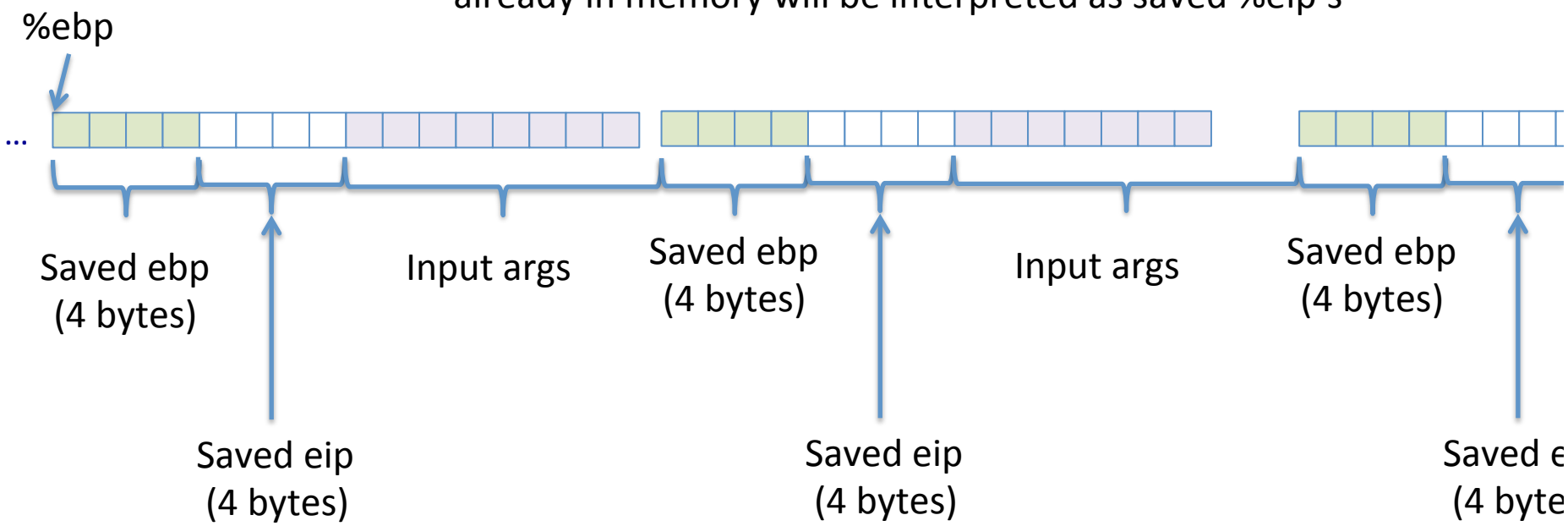


# Lots of other stack frames!



# Return-Oriented Programming

Inject data onto the stack so that addresses of instructions already in memory will be interpreted as saved %eip's



# Return-Oriented Programming

- Find short snippets of code that do something useful, then return
  - Call these “gadgets”
  - Figure out the right sequence of gadgets to implement your payload
  - Inject data to write the addresses of your gadgets into the return addresses of successive stack frames
- When the program runs, as it pops the stack, your gadgets run execute the payload

# Return-Oriented Programming

- Doesn't require code injection!
- Much more difficult than simple stack smashing
  - Essentially requires a "compiler"
- Depends on the presence of enough useful "gadgets" at predictable locations in memory