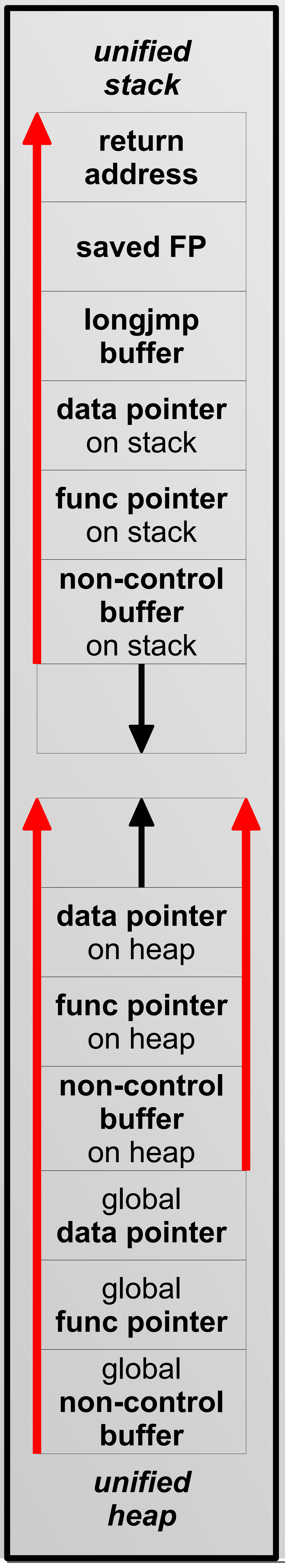
CERIAS

the center for education and research in information assurance and security

Implicit Buffer Overflow Protection Using Memory Segregation

Brent Roth (broth@purdue.edu)
Dr. Eugene Spafford (spaf@purdue.edu)

Modern Process



unified stack

return address

saved FP

longjmp buffer

data pointer on stack

func pointer on stack

non-control buffer on stack

data pointer on heap

func pointer on heap

non-control buffer on heap

global data pointer

global func pointer

global non-control buffer

unified heap

Motivation

The memory for a single process contains multiple forms of data.

- control data
  - return addresses, saved frame pointers, longjmp buffers, etc. that form the *call stack*
  - function and data pointers provide references to memory for calling functions and manipulating data
- non-control data
  - primitive datatypes (int, char, float, double, etc.) are used to store program-defined data

Modern processes store these different forms of data in the same unified stack and unified heap in the same memory segment. This allows a buffer overflow of non-control data to corrupt control data.

Modern defenses are still circumvented by modern attacks and do not prevent the corruption of control data. Instead they attempt to prevent it from hijacking control flow or detect it and terminate the process.

- Canary
- ASLR
- Non-executable memory

The corruption of control data can still be used for a denial-of-service attack

- Some defenses against buffer overflow result in denial-of-service
  - terminate process if detect corruption
  - force buffer overflow to result in a segmentation fault

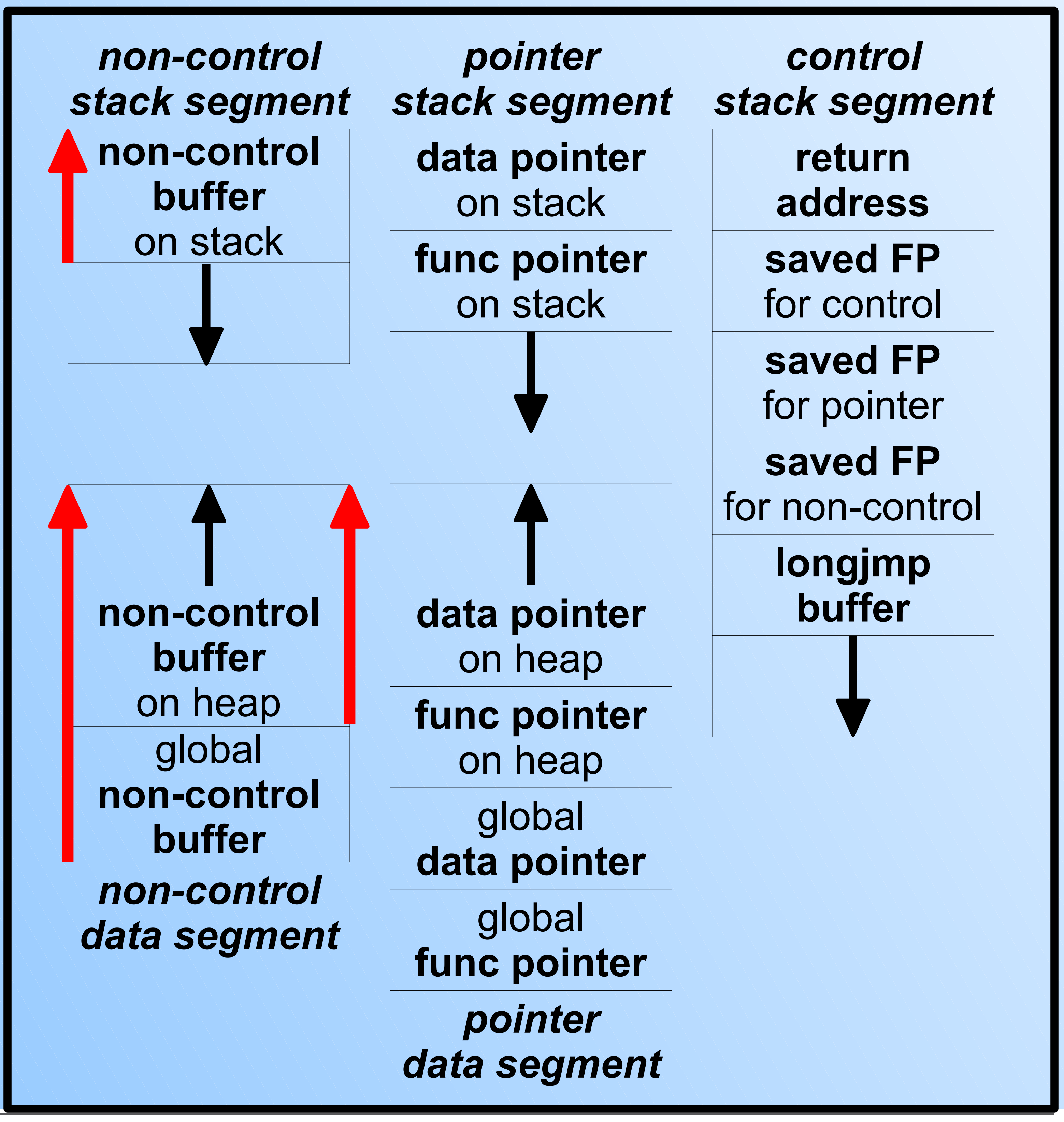
Goal

Segregate different forms of a data to their own stacks and heaps in their own memory segments within the same process. An instruction to read/write memory in one memory segment can not read/write memory in a separate memory segment. Thus, a buffer overflow of non-control data cannot corrupt control data. With control data uncorrupted, recovery is more likely, making denial-of-service harder to achieve with a buffer overflow.

Explore architecture modifications to further support memory segregation and corruption prevention

- Instruction Set Extensions
- Stack Growth Direction
- Secure Indirection

Process w/ Segregated Memory



non-control stack segment

non-control buffer on stack

non-control buffer on heap

global non-control buffer

non-control data segment

pointer stack segment

data pointer on stack

func pointer on stack

data pointer on heap

func pointer on heap

global data pointer

global func pointer

pointer data segment

control stack segment

return address

saved FP for control

saved FP for pointer

saved FP for non-control

longjmp buffer

PURDUE
UNIVERSITY

CERIAS

Discovery Park
e-Enterprise Center