Lecture 2: Vulnerabilities

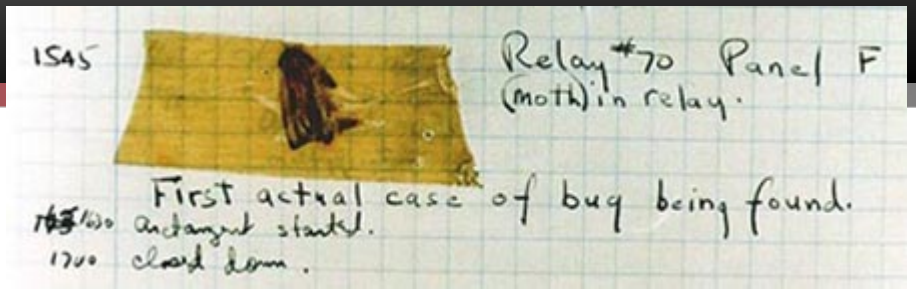
CS 181S

September 10, 2018

The Big Picture

Attacks
are perpetrated by
threats
that inflict
harm
by exploiting
vulnerabilities
which are controlled by
countermeasures.

Bugs



"bug": suggests something just wandered in

[IEEE 729]

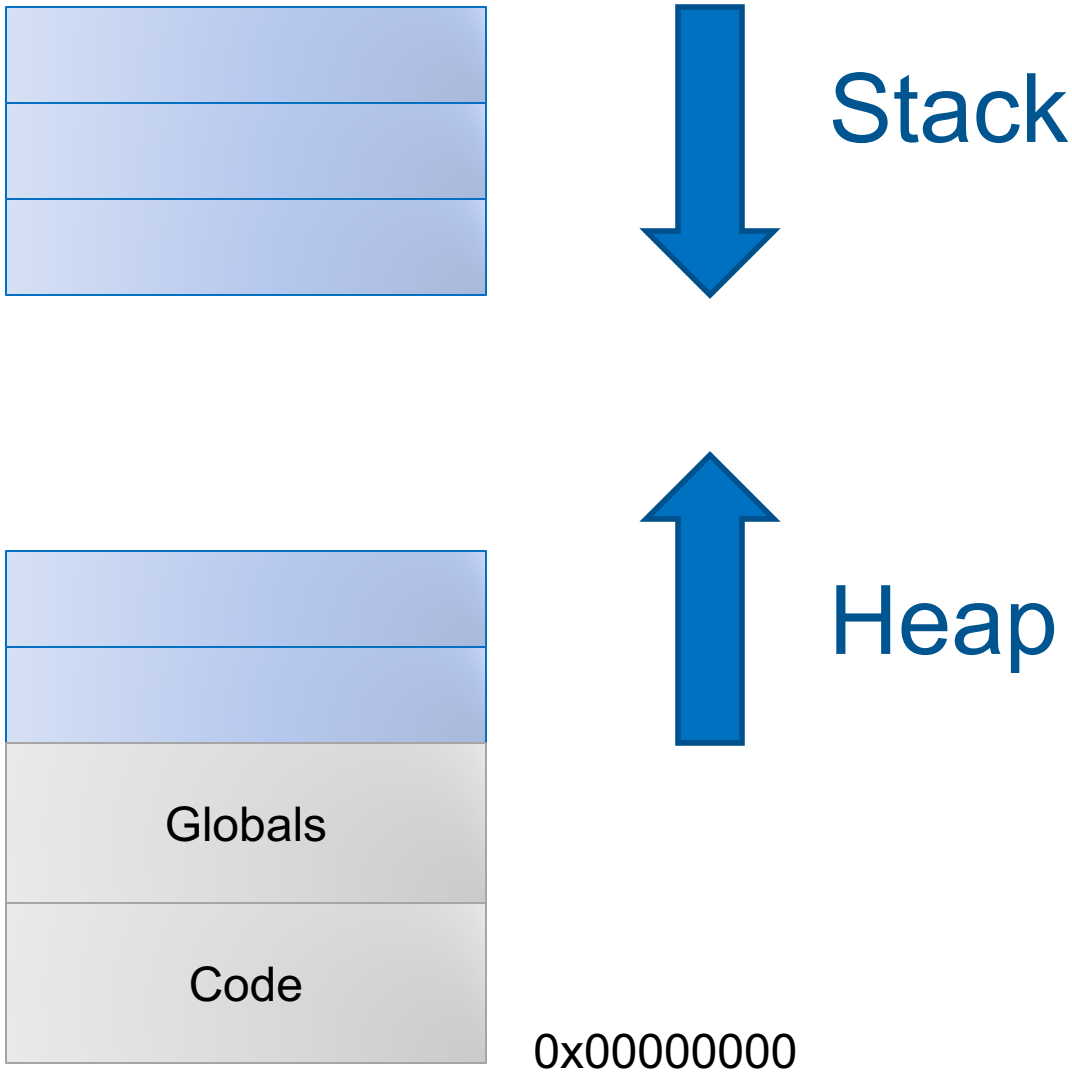
- Fault: result of human error in software system
  - E.g., implementation doesn't match design, or design doesn't match requirements
  - Might never appear to end user
- Failure: violation of requirement
  - Something goes wrong for end user

Vulnerability

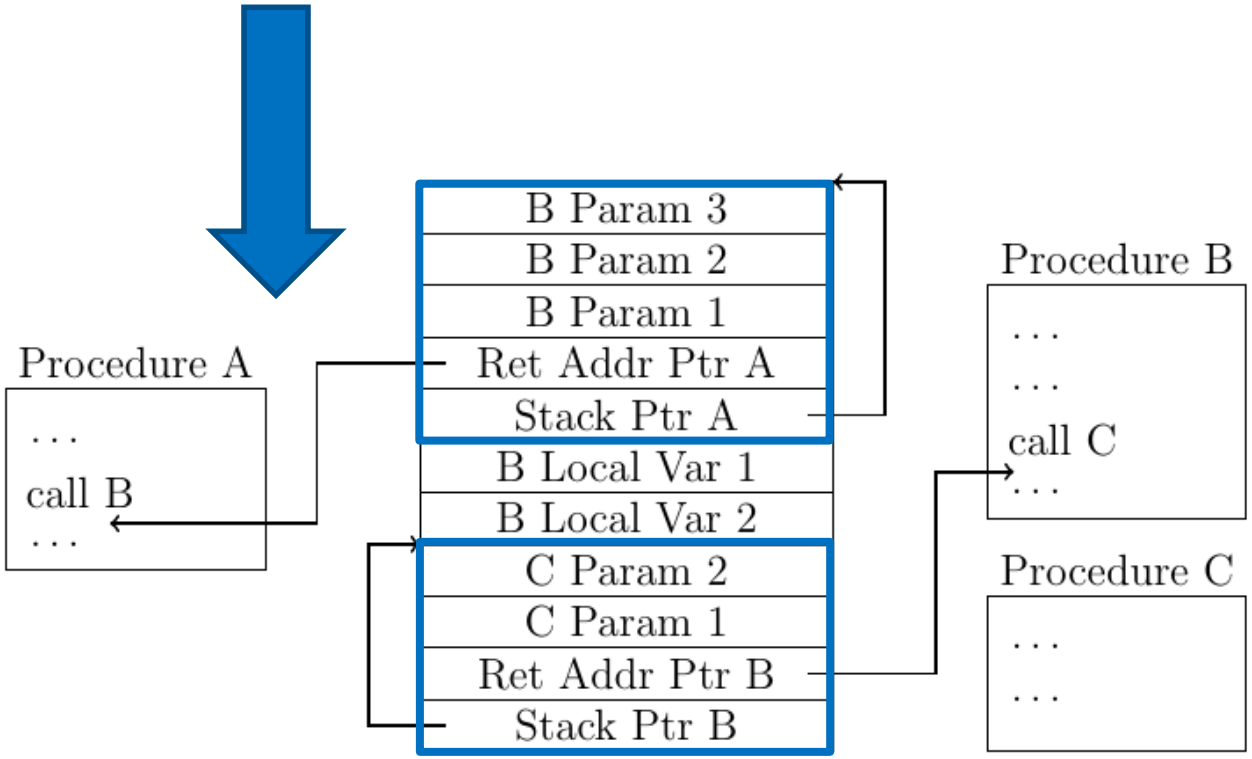
An unintended aspect of a system (design, implementation, or configuration) that can cause the system to do something it shouldn't, or fail to do something it should

- E.g., buffer overflows, code injection, cross-site scripting, missing authentication or access control, misconfiguration
- National databases: CVE, NVD
- Ignoring vulnerabilities is risky
  - Too often: "no one would/could ever exploit that"
  - *Weakest link* phenomenon
- **Assumptions are vulnerabilities**

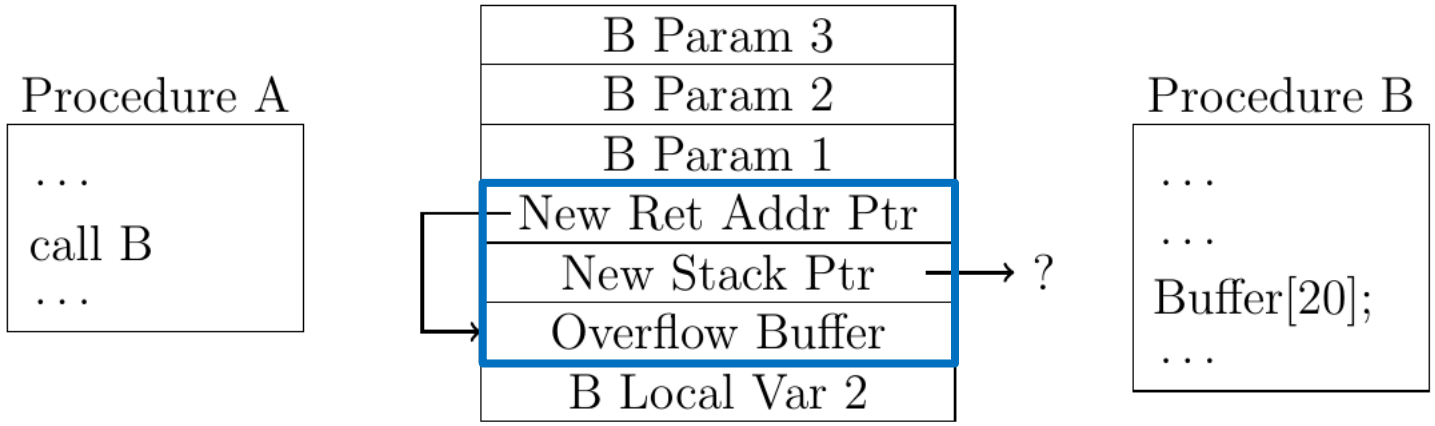
Memory: A Quick Review



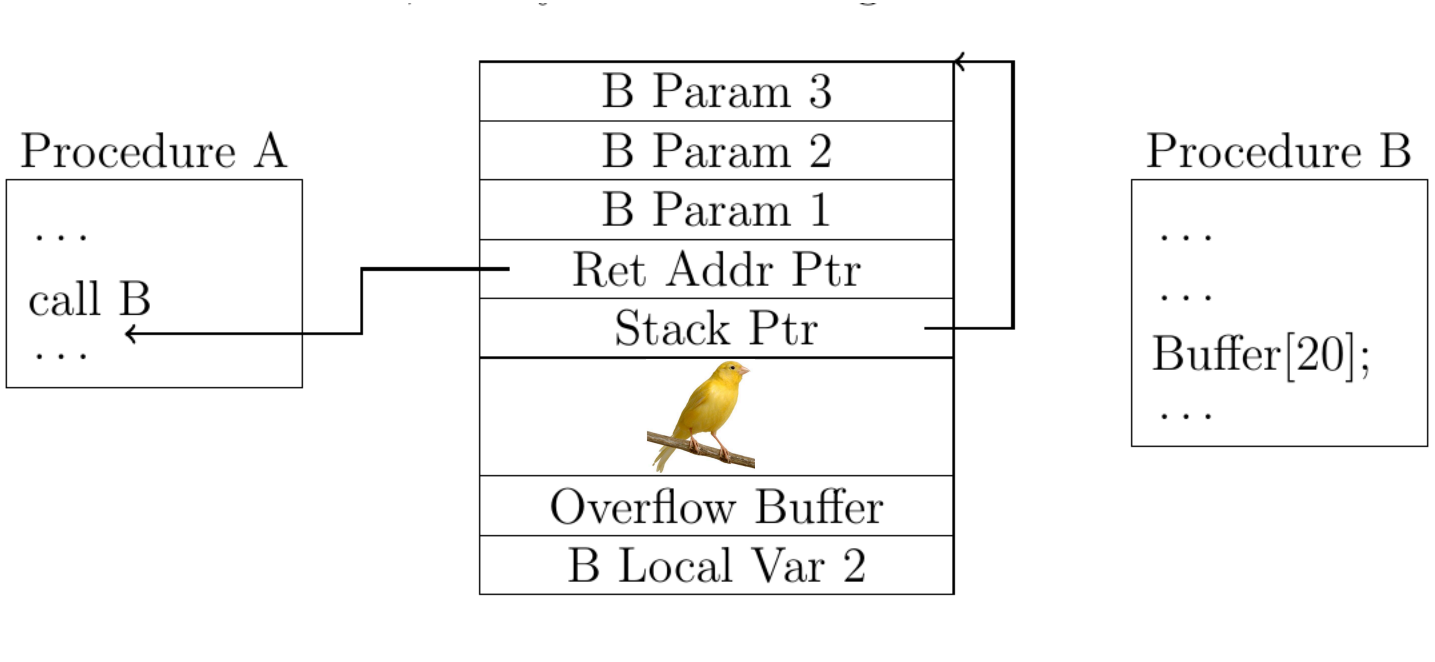
The Stack


B Param 3
B Param 2
B Param 1
Ret Addr Ptr A
Stack Ptr A
B Local Var 1
B Local Var 2
C Param 2
C Param 1
Ret Addr Ptr B
Stack Ptr B
Procedure A
. . .
call B
. . .
Procedure B
. . .
. . .
call C
. . .
Procedure C
. . .
. . .

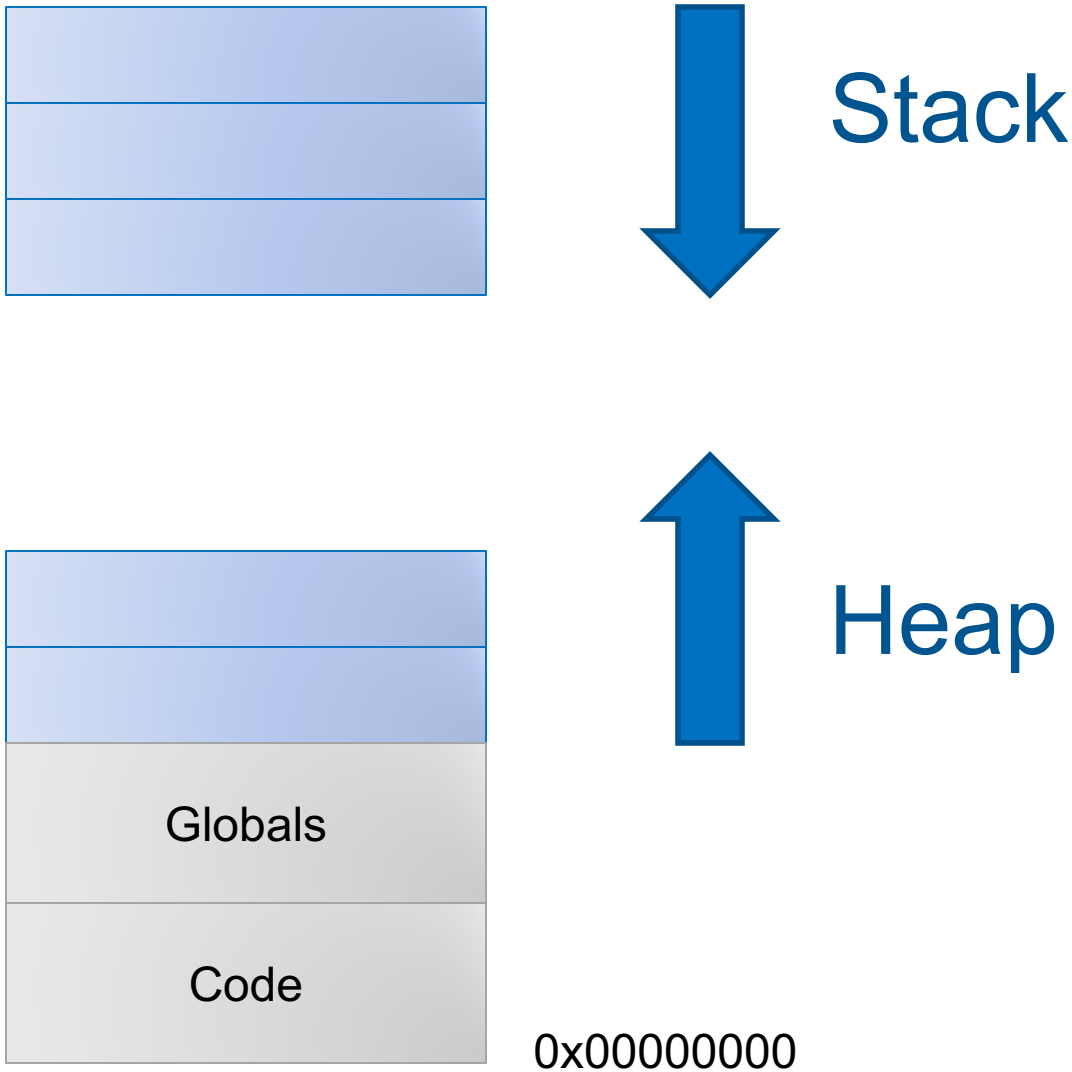
Stack Smashing



Canaries

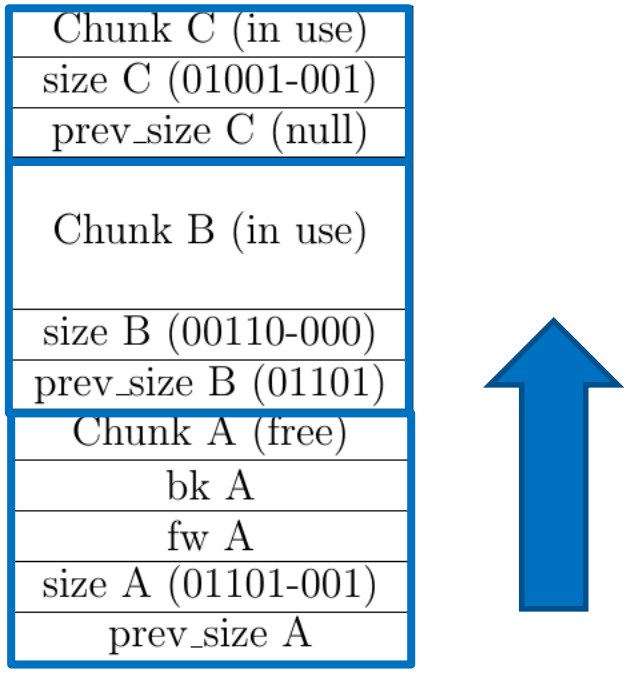

Procedure A
...
call B
...
B Param 3
B Param 2
B Param 1
Ret Addr Ptr
Stack Ptr
Overflow Buffer
B Local Var 2
Procedure B
...
...
Buffer[20];
...

Memory: A Quick Review



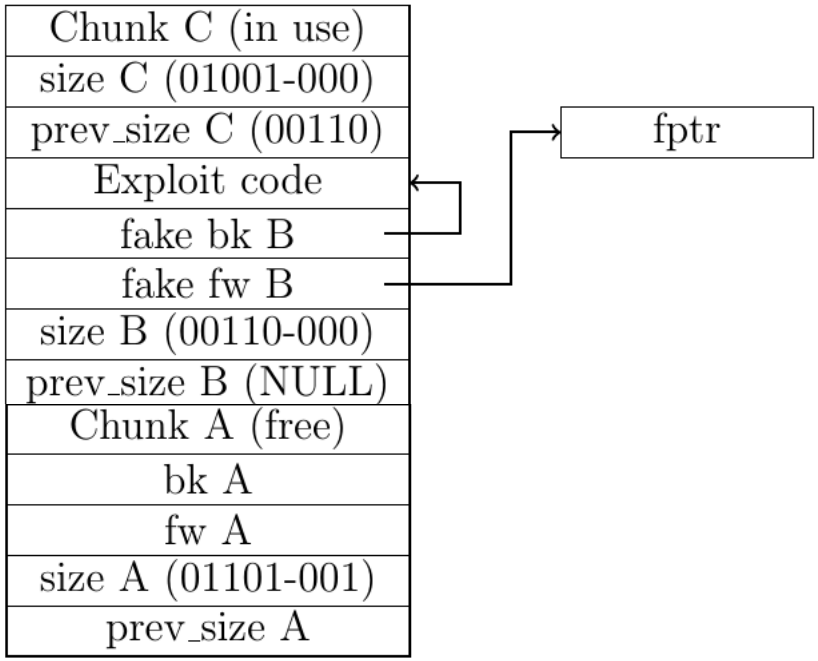
The Heap

```
INTERNAL_SIZE_T prev_size;     /* size of prev chunk (if free) */
INTERNAL_SIZE_T size;          /* size of chunk  */

struct chunk * fd;             /* double links -- used only if free */
struct chunk * bw;
```



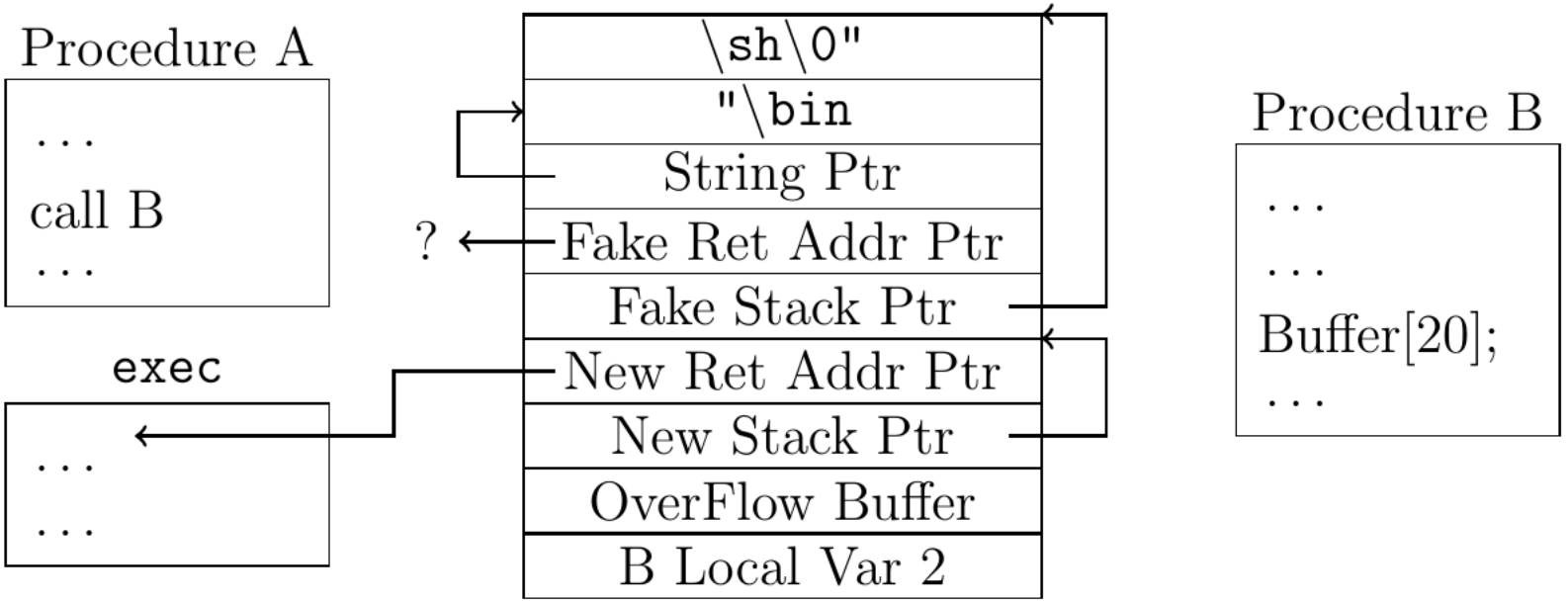
Heap Smashing

```
INTERNAL_SIZE_T prev_size;    /* size of prev chunk (if free) */
INTERNAL_SIZE_T size;         /* size of chunk  */

struct chunk * fd;            /* double links -- used only if free */
struct chunk * bw;
```



Memory Tagging

Return-into-libc



Address Space Layout Randomization

x86

- Intel Instruction Set Architecture (ISA)
- Introduced 1978, still supported
- As of 2018, most common architecture on servers, PCs, and laptops
- dense instruction set
- variable length instructions
- not word aligned

Return Oriented Programming

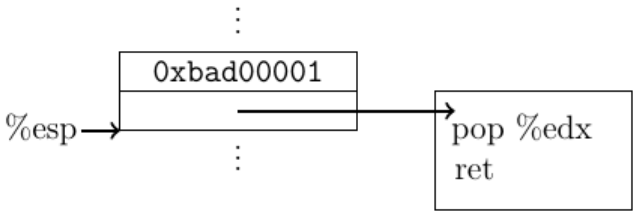
```
f7 c7 07 00 00 00
0f 95 45 c3
```

test $0x00000007, %edi
setnzb -61 (%ebp)

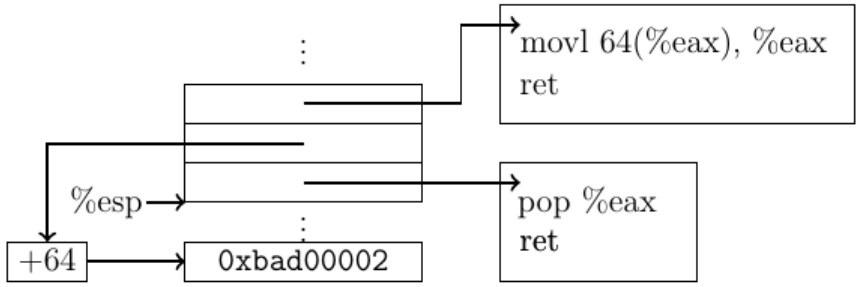
```
c7 07 00 00 00 0f
95
45
c3
```

movl $0x0f0000000, (%edi)
xchg %ebp, %eax
inc %ebp
ret

Example Gadgets



(a) Load constant gadget

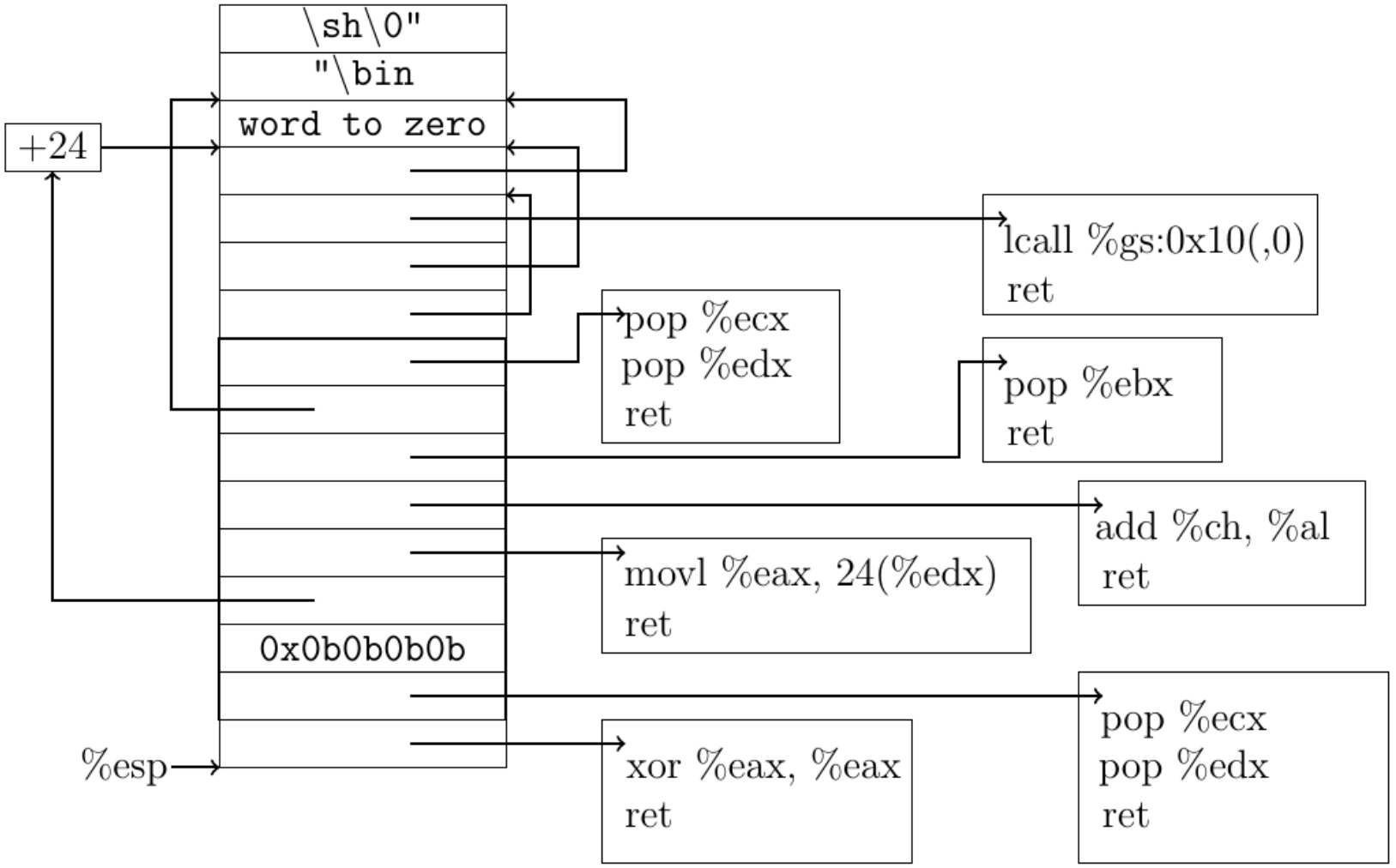


(b) Load from memory gadget

Return Oriented Programming

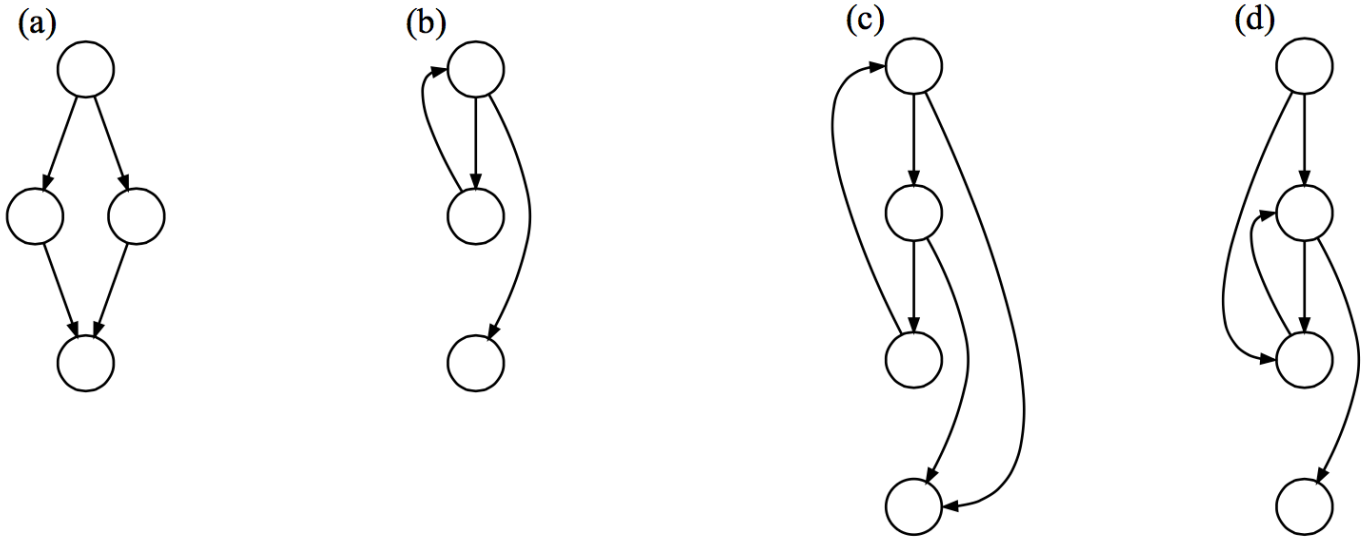
Image By: Dino Dai Zovi

Return-Oriented Shellcode

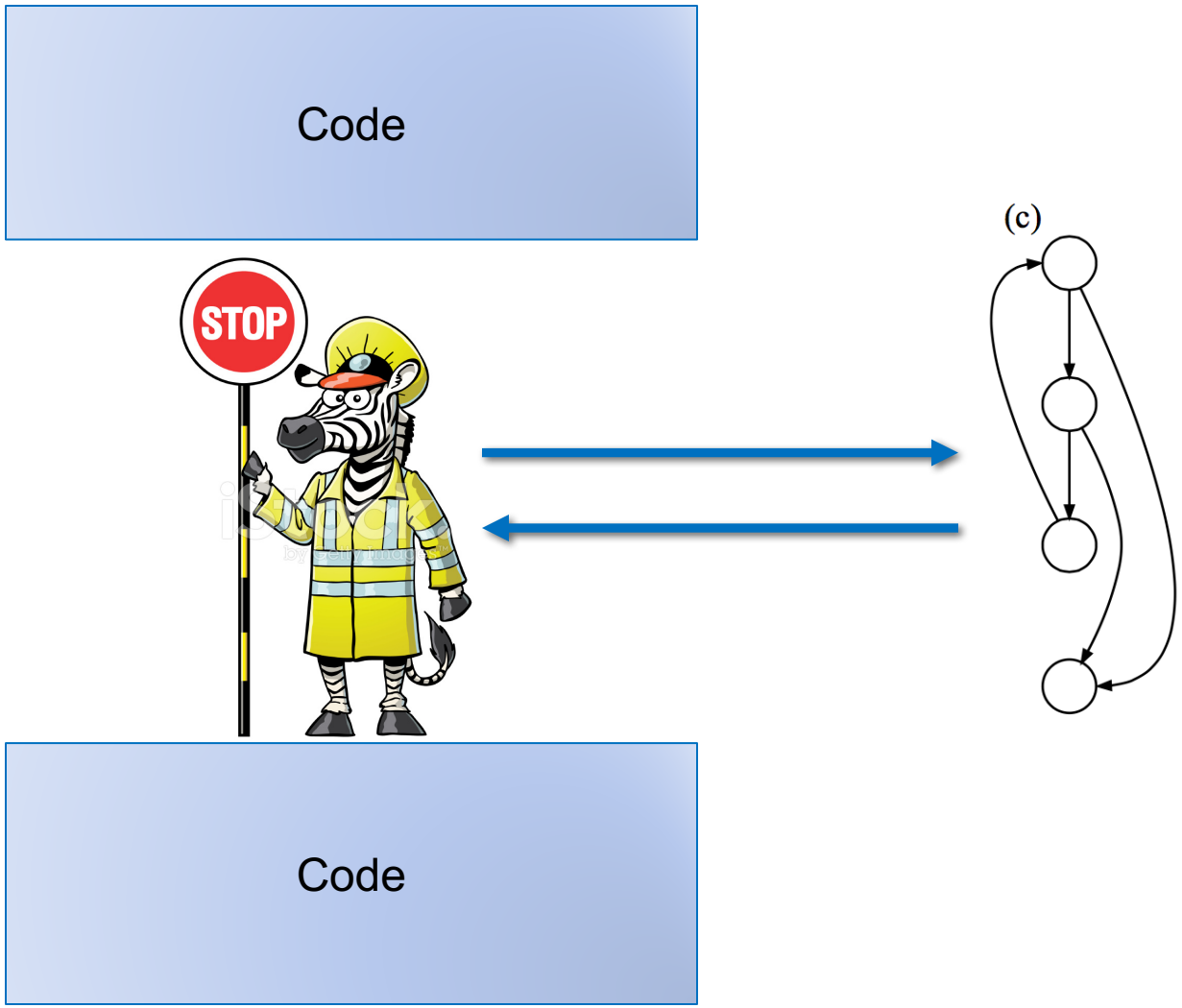


Gadget Elimination

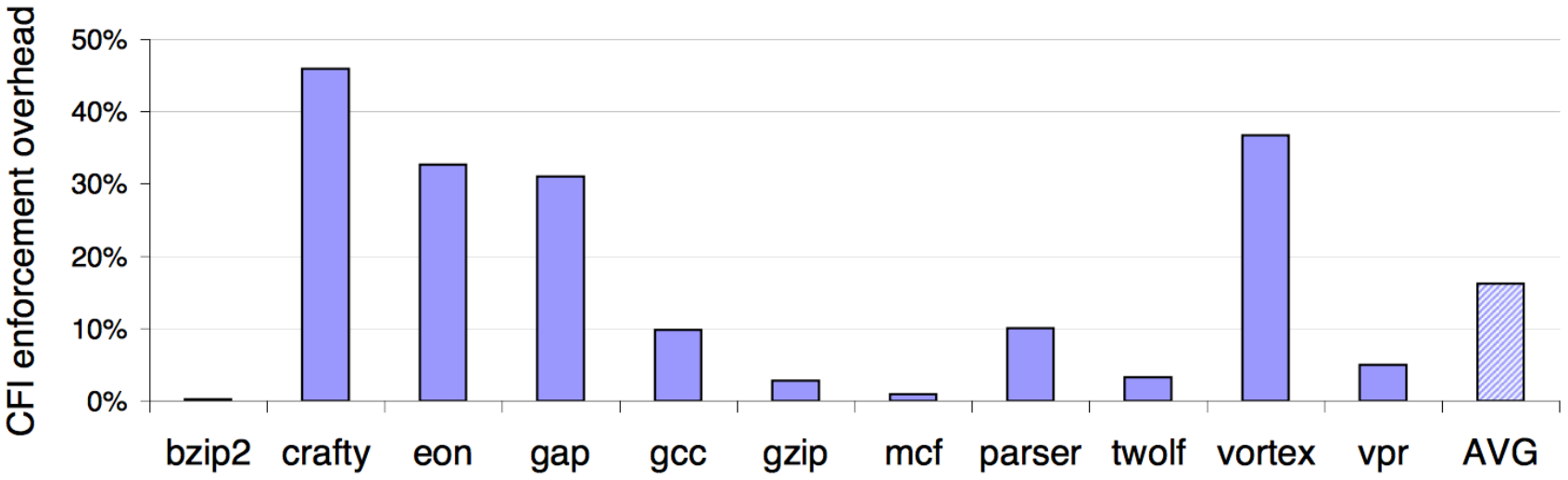
Control Flow Integrity



CFI = Insert Monitors

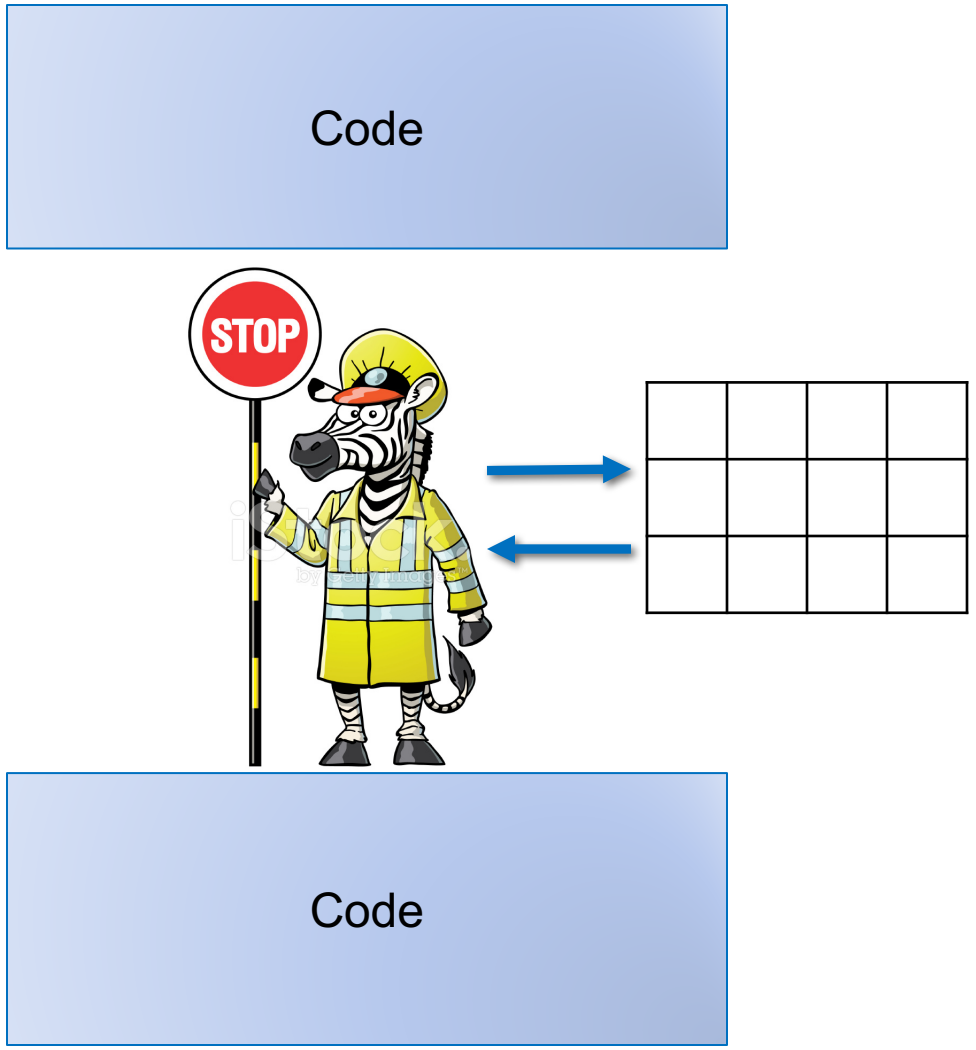


CFI Overhead

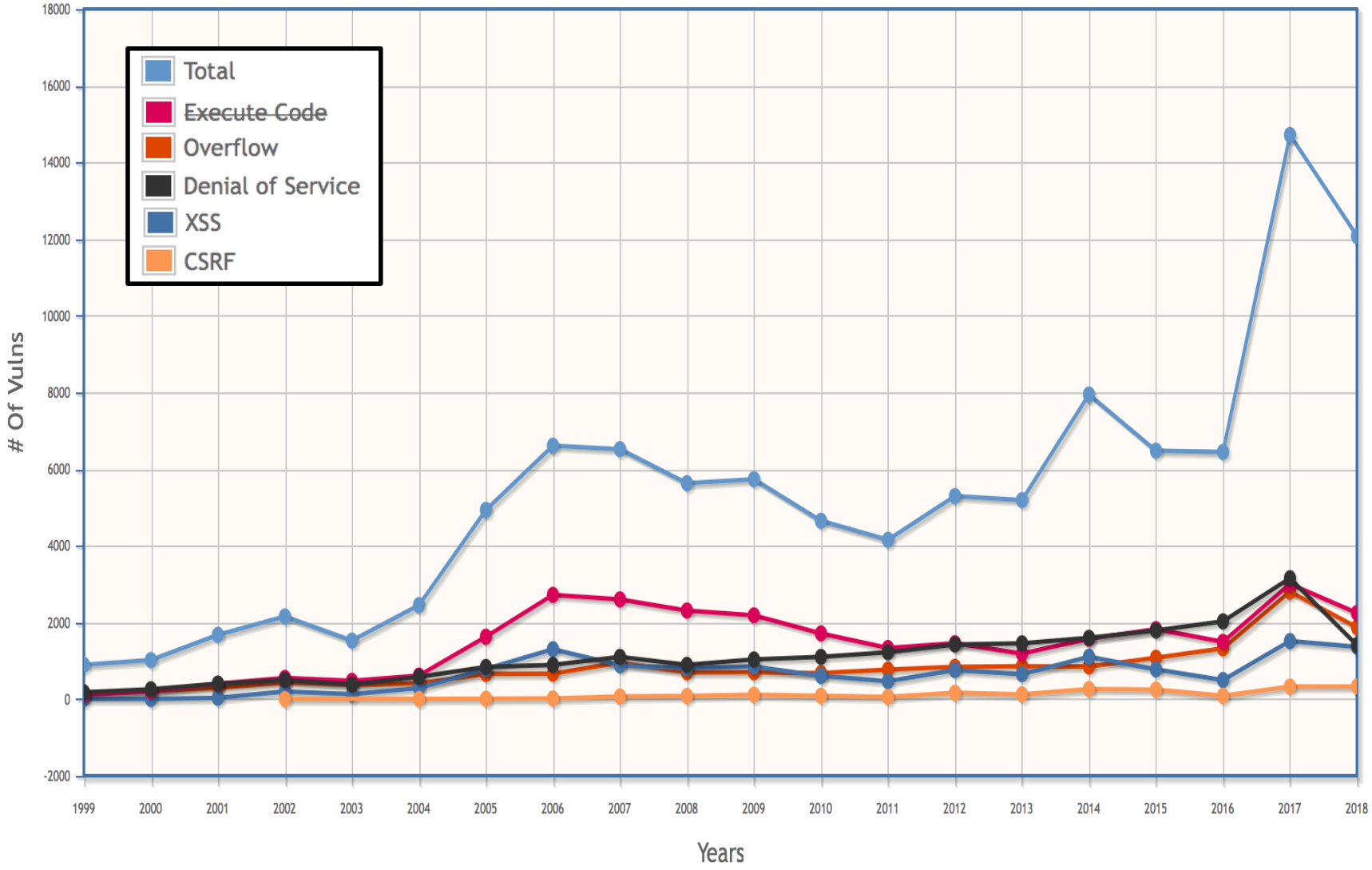


Control Flow Guard

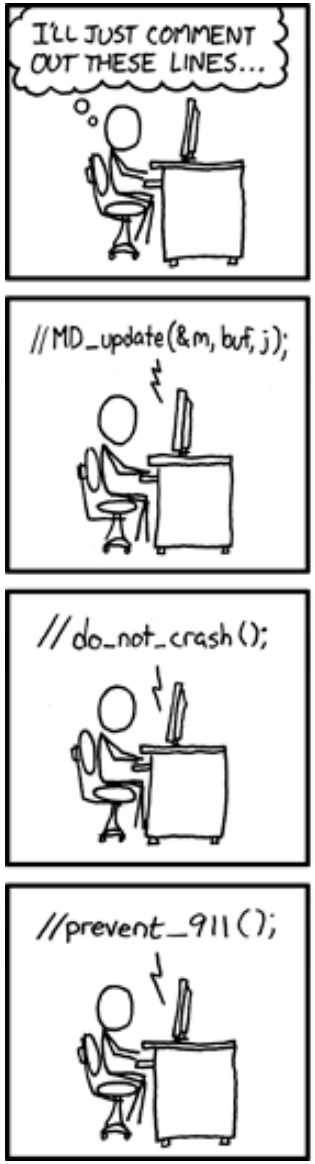
- Approximate CFI implementation in Windows 8.1, 10
- Jump is valid if it beginning of function
  - Granularity: 8 bytes
- Check implemented as bitmap



Vulnerabilities by Year



Vulnerabilities



IN THE RUSH TO CLEAN UP THE DEBIAN-OPENSSL FIASCO, A NUMBER OF OTHER MAJOR SECURITY HOLES HAVE BEEN UNCOVERED:

AFFECTED SYSTEM	SECURITY PROBLEM
FEDORA CORE	VULNERABLE TO CERTAIN DECODER RINGS
XANDROS (EEE PC)	GIVES ROOT ACCESS IF ASKED IN STERN VOICE
GENTOO	VULNERABLE TO FLATTERY
OLPC OS	VULNERABLE TO JEFF GOLDBLUM'S POWERBOOK
SLACKWARE	GIVES ROOT ACCESS IF USER SAYS ELVISH WORD FOR "FRIEND"
UBUNTU	TURNS OUT DISTRO IS ACTUALLY JUST WINDOWS VISTA WITH A FEW CUSTOM THEMES