THE UNIVERSITY of NEW ORLEANS

CSCI 4621/5621 Intro to CyberSecurity

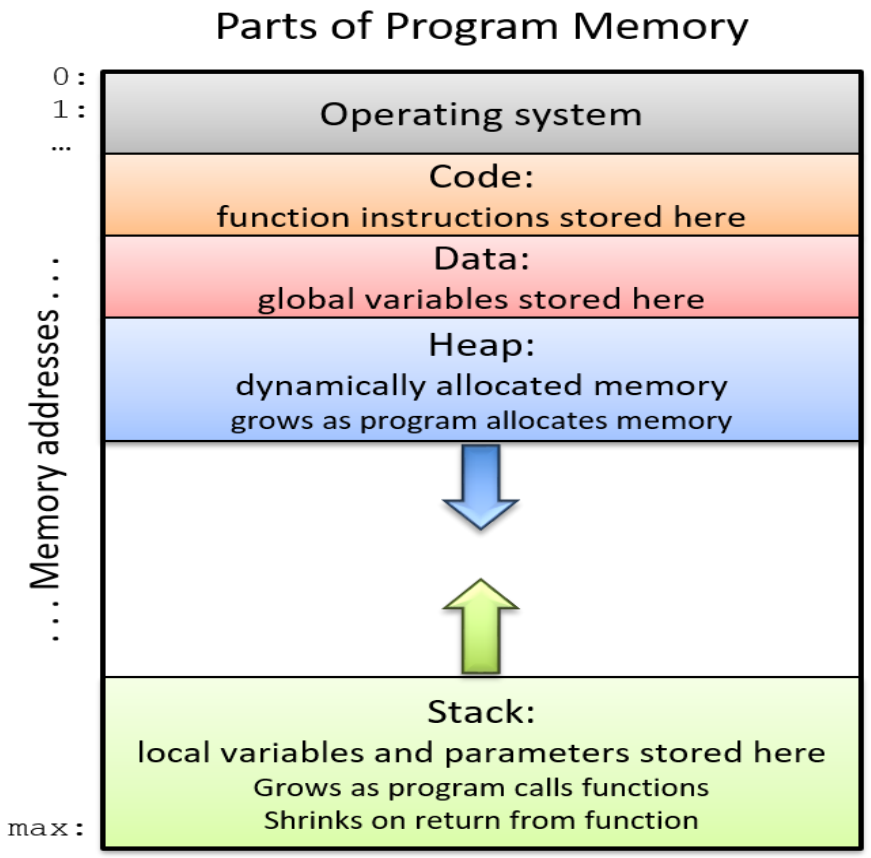
05: SMASHING THE STACK

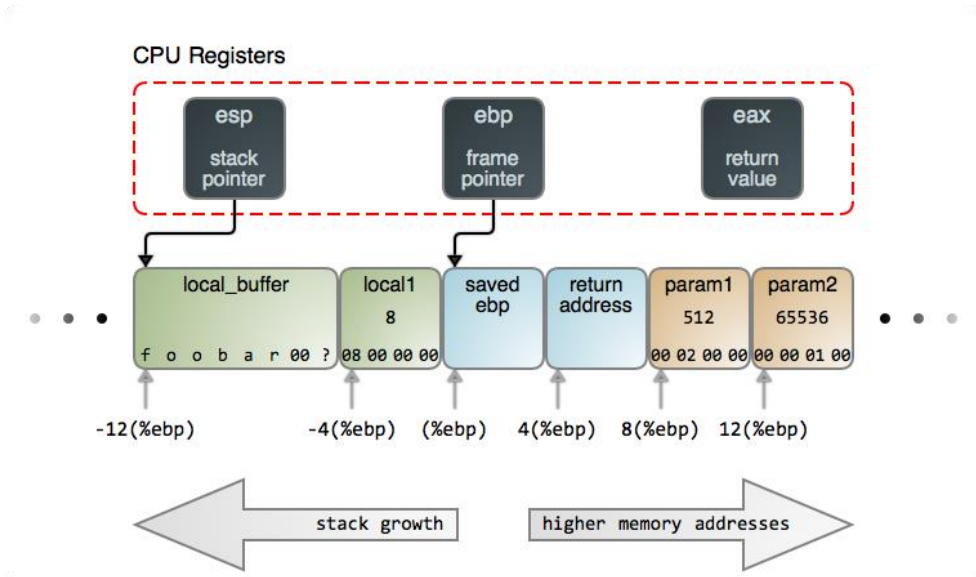
Vassil Roussev

vassil@cs.uno.edu

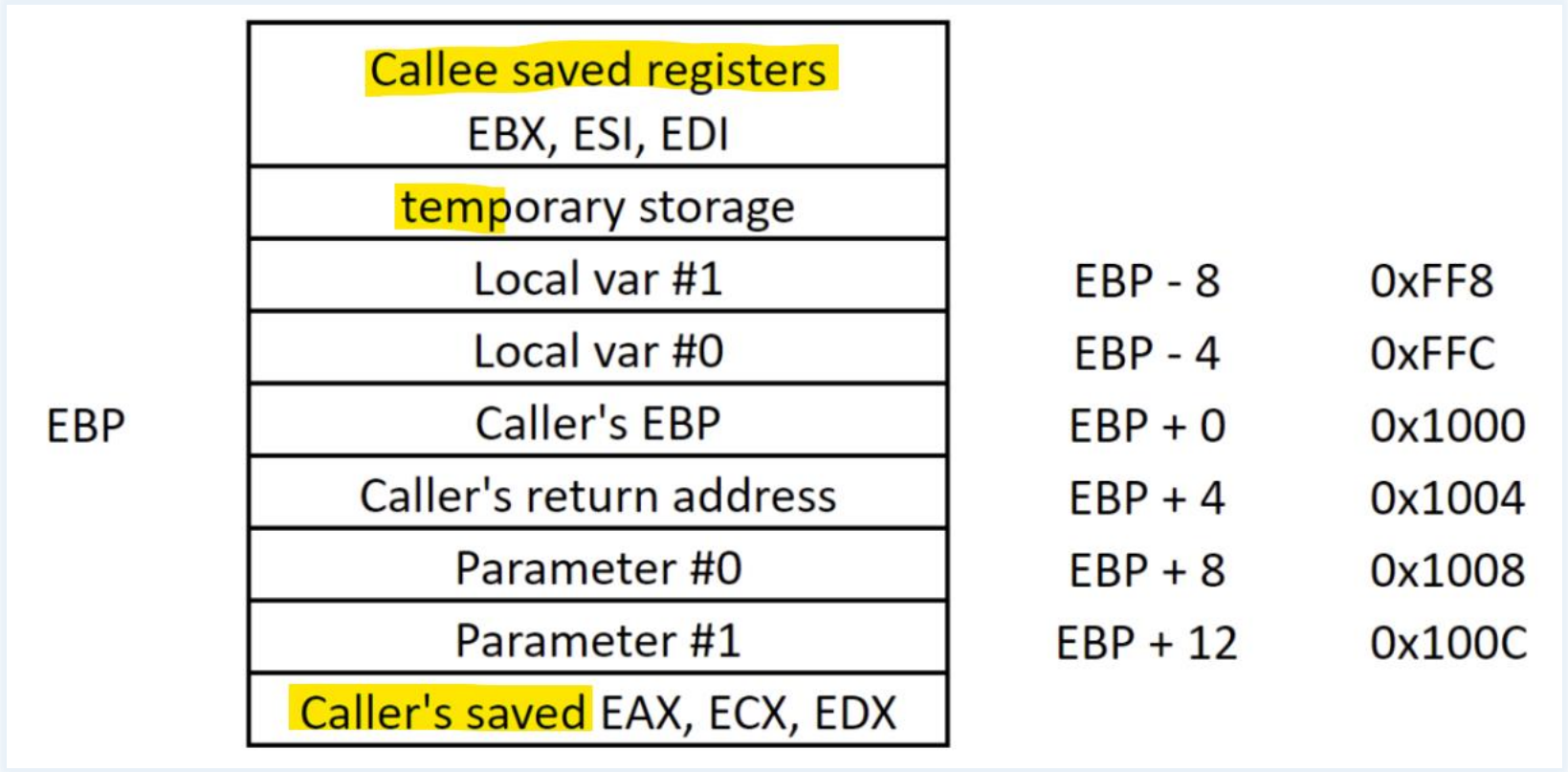
READING: Phrack 49 (1996)

PROCESS & MEMORY LAYOUT





Stacks in C (extended version)



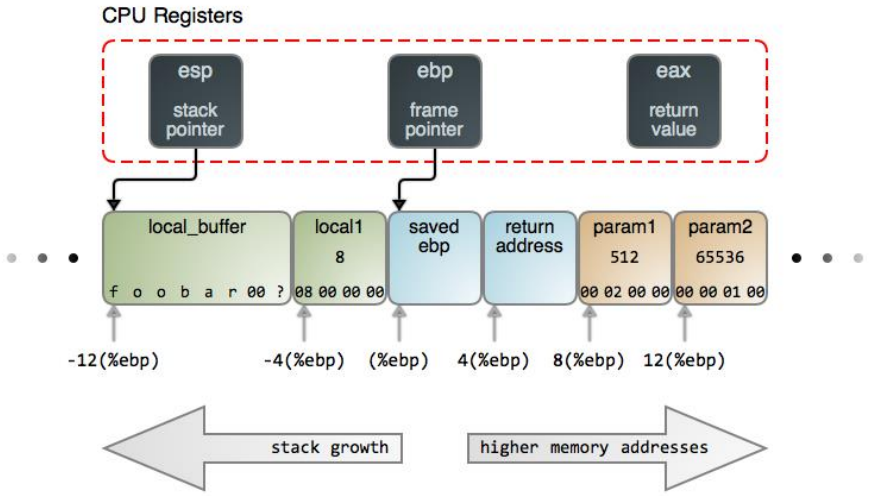
[CREDIT: Menon]

The code

- template:
  - `~/4621/overflow`
- base installation
  - *ubuntu*
    - `apt install build-essentials gcc-multilib nasm`
    - `apt install python2`
- quick look at the
  - `Makefile`
- build
  - `make <target>` → *executable*
  - `make <target>s` → *assembly*
- ref
  - [Smashing the stack for fun and profit] (phrack)
  - [Buffer overflow attacks explained] (medium)
  - [Buffer overflow works in gdb but not without it] (stack overflow)

STEP 1 – A C VIEW OF THE STACK

- overflow1.c
- Goals
  - » *play w/ pointer arithmetic*
  - » *observe the lack of boundaries*
- Experiments
  - » *print parameters* **a**, **b**, **c**, *local variables* **b1**, **b2**, **b3**
  - » *push index boundaries*
  - » *observe* **(&a)[-1]**
  - » *observe* **((unsigned int *)b1)[5]**



Step 2

- overflow2.c
- goal
  - » *redirect control flow*
  - » *change program logic*

Step 3 – shellcode

- execve()
  a. Have the null terminated string "/bin/sh" somewhere in memory.
  b. Have the address of the string "/bin/sh" somewhere in memory followed by a null long word.
  c. Copy **0xb** into the EAX register.
  d. Copy the address of the address of the string "/bin/sh" into the EBX register.
  e. Copy the address of the string "/bin/sh" into the ECX register.
  f. Copy the address of the null long word into the EDX register.
  g. Execute the **int $0x80** instruction.

Step 4 – adding the NOP sled

- make nop_shell
- work in gdb to determine correct addresses

Step 5 – vuln + shellcode

- disable ASLR
  - » `sudo sysctl kernel.randomize_va_space=0`
- `envexec`
  - » *ensures that code runs in the same environment*
    - addresses found in `gdb`
  - » *run in `gdb` to get address*
    - `make run_gdb`
  - » *run in shell*
    - `make run_shell`
- profit!