Paxson
Spring 2017

CS 161
Computer Security

Discussion 1

# Week of January 23, 2017: GDB, x86, and Security Principles

Studying memory vulnerabilities requires being able to read assembly and step through it with a debugger. In this class, we'll be using 32-bit x86 and GDB.

## A few useful GDB commands

For OS X users: `lldb` uses different commands. You will be expected to know `gdb`.

- run (r)
- break (b) ⟨ `func` | **addr* | `line` ⟩: add a breakpoint at the specified spot
- step (s): continue to next line, next (n): next line, skip function calls
- stepi (si), nexti (ni): same, but at the instruction level
- continue (c): until next breakpoint
- ⟨enter⟩: repeat previous command
- print (p) [ /*f* ] ⟨ `var` | `$register` ⟩: print the specified value (in format *f*)
- list (l) [`line`]: show source code around the current line or `line`
- layout split: splits the GDB interface into source, assembly, and commands sections.
- disassemble (disas) [`func`]: show the assembly for the current context, or `func`
- x/*n*x[b|w] *addr*: print *n* bytes (b) or 4-byte words (w) of memory as hex (x)
  (If displaying bytes, keep in mind that x86 is *little-endian*!)

## Intro to x86 assembly

32-bit x86 prefixes its registers with *e-* (eax, ebp, esp...). x86–64 uses *r-* (rax, rbp, rsp...).

In AT&T syntax, the suffixes *-b*, *-i*, *-l*, and *-q* clarify if the instruction operates on bytes, 16-bit words, 32-bit words, or 64-bit words. Source is on the left, destination on the right.

There are 8 general-purpose registers: EAX, EBX, ECX, EDX, ESI, EDI, ESP, and EBP. The registers EBP (base pointer) and ESP (stack pointer) are usually used to delimit the current function's *stack frame*.

The stack grows down (towards lower addresses), by decrementing ESP (`subl $0x18, %esp`) or using the shortcut `push`: `pushl %ebp` (decrement ESP by 4 and copy EBP there).

Correspondingly, `popl %ebp` puts the memory (ESP,ESP+4) into EBP and increments ESP.

The usual *function prologue* is

```
push %ebp       // save the top of the previous frame
mov %esp %ebp   // start new frame by moving EBP down to ESP
sub X %esp      // X = size of local variables
```

And the corresponding exit is

```
add X %esp      // * (sometimes ‘mov %ebp %esp‘)
pop %ebp        // *
ret             // pops return address from stack, goes there
```

* sometimes these two lines are replaced with just `leave`.

Conversely to `ret`, `call` ***addr*** pushes EIP (the instruction pointer, that is, the address of the *next* instruction) onto the stack as a saved return address before jumping to *addr*.

A more thorough overview of 32-bit x86 can be found at https://www.cs.virginia.edu/~evans/cs216/guides/x86.html

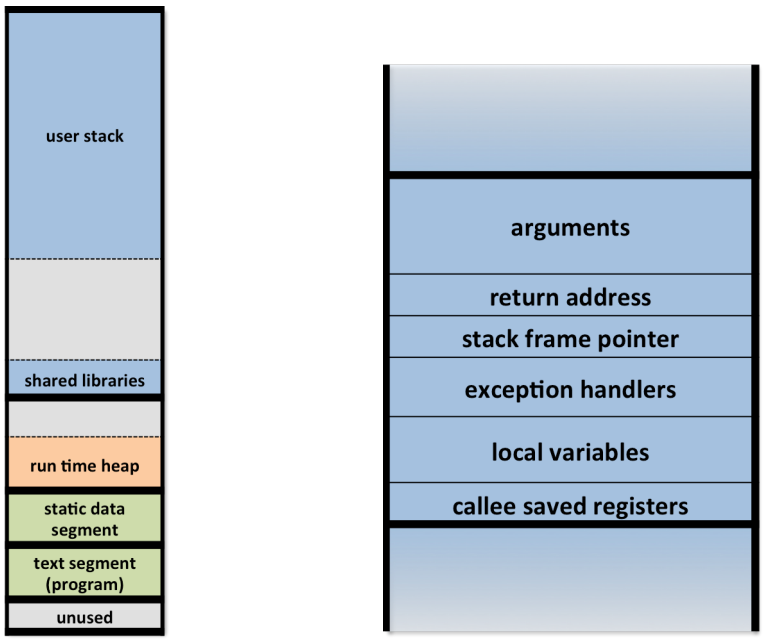


Figure 1: Left: memory layout for 32-bit Linux. The stack (left, at top) grows downward. Right: the contents of one frame on the stack (exercise: match the entries up with the instructions in the function prologue and exit).

Note: **Feel free to come by office hours held by any of the staff.** Don't hesitate to ask for help! Our office hours exist to help you. Please visit us if you have any questions or doubts about the material.

---

**Question 1** ***Security Principles*** **(10 min)**

We discussed the following security principles in lecture (*or in the lecture notes*, which you are responsible for reading):

A. Security is economics
B. Least privilege
C. Failsafe defaults
D. Separation of responsibility
E. Defense in depth
F. Psychological acceptability
G. Human factors
H. Complete mediation
I. Know your threat model
J. Detect if you can't prevent
K. Don't rely on security through obscurity
L. Design security in from the start

Identify the principle(s) relevant to each of the following scenarios:

1. New cars often come with a valet key. This key is intended to be used by valet drivers who park your car for you. The key opens the door and turns on the ignition, but it does not open the trunk or the glove compartment.
2. Many home owners leave a house key under the floor mat in front of their door.
3. Convertible owners often leave the roof down when parking their car, allowing for easy access to whatever is inside.
4. Warranties on cell phones do not cover accidental damage, which includes liquid damage. Unfortunately for cell phone companies, many consumers who accidentally damage their phones with liquid will wait for it to dry, then take it in to the store, claiming that it doesn't work, but they don't know why. To combat this threat, many companies have begun to include on the product a small sticker that turns red (and stays red) when it gets wet.
5. Social security numbers, which we all know we are supposed to keep secret, are often easily obtainable or easily guessable.
6. The TSA hires a lot of employees and purchases a lot of equipment in order to stop people from bringing explosives onto airplanes.

**Solution:** (Note that there may be principles that apply other than those listed below.)

1. Principle of least privilege. They do not need to access your trunk or your glove box, so you don't give them the access to do so.
2. Unfortunately we often do rely on security through obscurity. The security of your home depends on the belief that most criminals don't know where your key is. With a modicum of effort, criminals could find your key and open the lock.
3. Security is economics. Even if they left the top up, it would be easy for a criminal to cut through it. If the criminals did that, it would cost the owner the cost of the items in the car and the cost of a new roof!
4. Detect if you can't prevent. People will try to scam cell phone manufacturers, and there is nothing the companies can do to stop this. But they can (and do) detect when people have voided their warranty via liquid damage.
5. Design security in from the start. Social security numbers were not designed to be authenticators, so security was not designed in from the start. The number is based on geographic region, a sequential group number, and a sequential serial number. They have since been repurposed as authenticators.
6. Security is economics. They spend a lot of money to protect airplanes, lives, and the warm/safe/fuzzy feeling that people want to have when they fly.