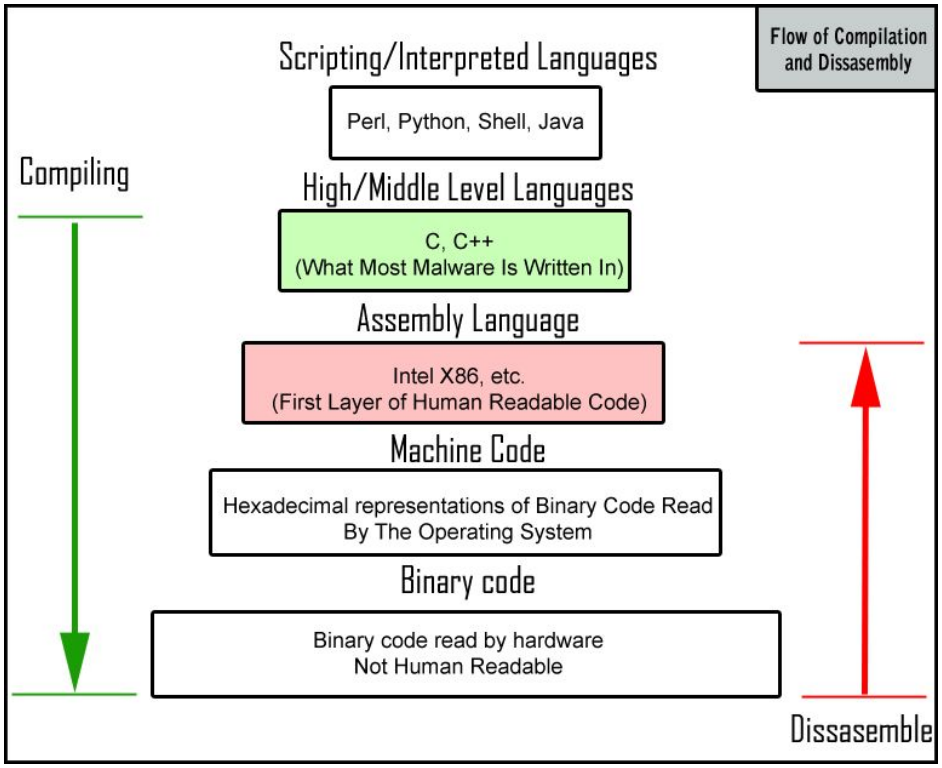
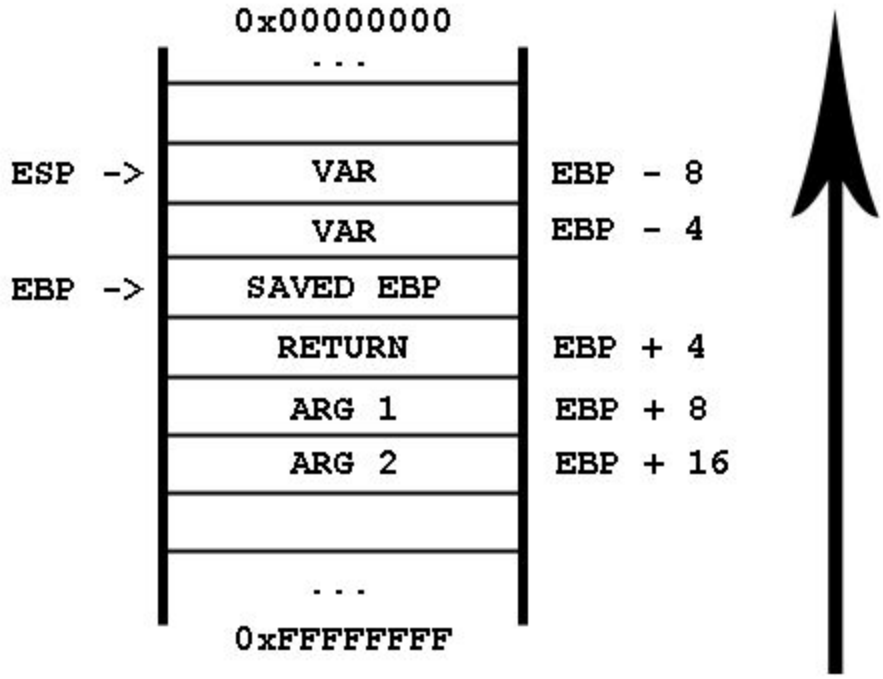
Chapter 7

What does a program actually look like in memory?



What is “the Stack”?



What is the first step?
-Test if vulnerable by sending bigger and bigger buffers until it crashes
Next?
-Pattern_create - determine location of IEP
-Pattern_offset
What is shellcode?

- Do we need to use \x90?
- Why put the 16* (\x90) before the shellcode?

How do you “find space” for your shellcode?

- Will there always be enough space?
    - Solutions? (staged payload, trampoline)

What are bad characters?

- How many times do you need to test the bad character (what is signal to stop?)

Why JMP ESP?

- ESP points directly to the start of your payload (after execution of the `ret` in the function you're attacking) because you put the payload right after the 4 bytes that overwrite the return address on the stack. `ret` pops 4 (or 8) bytes into EIP, leaving ESP pointing to the payload that directly follows.

Overall Steps:

1. Fuzz (Crash a Program)
2. Control EIP
    a. Pattern_create / Pattern_offset
    b. In mona.py or in msf (.rb)
3. Locate Space for your Shellcode
    a. Standard Shellcode requires 350-400 Bytes
    b. Stagers requires
4. Determine Bad Characters
    a. Need to do multiple times (breaks on bad characters)
    b. Stop when see FE, FF (or end non-bad chars)
5. Redirect Program Execution
    a. Locate modules without protections
        i. !mona modules
    b. Find a return address (JMP ESP)
        i. Does not have to be a single instruction
6. Generate appropriate shellcode
    a. Must match OS version (32/64)
    b. Different prompts based on shellcode type

    c. EXITFUNC = Thread
    d. -e (specify encoder)
7. Get a Shell
    a. Different prompts
        i. Sometimes blank (no leading symbol)

- Prompts look different sometimes (look at kosta's homework - compare 7.7.1 to 8.8.1)
- What does it mean if not resolving with nc
- Why need to you a two stage exploit?
- What does the EXITFUNC = Thread do?
- Talk through msfvenom inputs

-Tell student looking through code. Look specifically at Matts (like exercise 7.8.1)
-Show pwntools
Emphasize turning in assignments on time
-Remind going to combine lesson 11

-How to troubleshoot
 --Exploit not working on
    --verify network connectivity
    --check inputs to shellcode