Abraham Adam, ID: 106302961

Stack Smashing

- Last time:
  - Isolation
    - Communicate through the OS with system calls
    - Ex: mov 137 $r0
    - Ex: mov $buf $r1
    - Syscall
  - Watch out for when writing OS
    - Syswrite (char *buf)
    - Char* tmp;
    - Int len;
    - Len = strlen(buf);
    - tmp = malloc (len+1);
    - strcpy (tmp,buf);
- Problem buf lies in the user space
- How to fix the problem
  - read only memory in OS space
  - Copy content to the OS variable
  - Make source read only
  - Copy on write
  - Common: update TLB to make process less access to buffer when the process makes system call.
- 2 important points:
  - Time of check to time of use (Tattou)
  - Use pass by value between trust domains

Virtual Memory Layout
• Wuftp 2.6.0
  o Void loguser (char * user){
  o Char buf[512]
  o Sprintf(buf,user)}
  o buf = 513 bits, because the stack is written bottom up we over write the return address
  o need to include 2 things for this attack:
    ▪ pointer for the code to run
    ▪ the code itself that will run
  o How to stop it:
    ▪ NX-bit: Make pages of the stack as non-executable
    ▪ Tare injection aka “intern-to-lib” still works
      • System( char * cmd ), expecting a pointer
• Why bad guys attack programs? get privalged that they are not supposed to have