Software Vulnerabilities

- Daniel Ready
- Prof. Schauma
- CS615A
- 21 April 2010
Software Security
Who cares?

- Recall lecture:
- Possible harm or loss of
  - Access to data
  - Integrity of data
  - Availability of services
  - Reputation
  - Monetary loss due to any of the above
  - Monetary loss due to physical items of actual value
Software Security

The Frontlines: Passwords

- Passwords are used to protect data from any of the previous harms
- Problems associated with passwords
  - Simplicity
  - Reuse
  - Easy Duplication
  - Problems with uniqueness
Software Security
Passwords: Vulnerable

- Simplicity
  - Dictionary Attacks
  - Rainbow tables
- Reuse
  - Phishing
- Easy duplication
  - Two way authentication
- Uniqueness
  - Keepass
Software Security
Injection Attacks

- Demands sanitation of user-input
- Allows user to run commands they typically wouldn't normally be able to
  - Considerations for +s bit
- PHP/MySQL Example:
  - `mysql_query("SELECT * FROM users WHERE name = '".$user."");`
  - What happens if $user = "name'; DROP TABLE 'users"?`
  - What happens if $user = "name' OR 1 = 1"?
Software Security
Injection Attacks, The Examples

- $user = "name'; DROP TABLE 'users";
  - Query becomes:
    - `mysql_query("SELECT * FROM users WHERE name = 'name'; DROP TABLE 'users'" );`

- $user = "name' OR 1 = 1";
  - Query becomes:
    - `mysql_query("SELECT * FROM users WHERE name = 'name' OR 1=1" );`
Trick a program which reads files, but respects permissions, into thinking that you have permission to read a file you don't actually have permission to read.

Vulnerable programs:

- if(access(file, R_OK) != 0) {
  return;
}
- int fd = open(file, RD_ONLY);

Pro-tip: Never use access to determine access to a file.
Software Vulnerabilities
Time-of-check-time-of-use

- How it works:
  - Program has higher permissions than user (+s)
  - `./myprog` file will print the file
  - User creates a file that he or she is able to read
  - `./myprog` readablefile
  - Between the access(3) call and the open(2) call, user switches readiblefile to be a symlink to a file he or she doesn't have access to, but myprog does
  - myprog opens the file and prints it
- How does the user switch it at just the right time?
Software Vulnerabilities

Buffer Overflows

- Leads to arbitrary code execution
- Very dangerous, relatively easy to exploit
- This is why you always check your buffer lengths!

How does it work?
- Let's review the stack frame
- What if you don't check your bounds?
- Let's see it in action

A note about randomize_va_space

See ”Smashing the Stack for Fun and Profit” by ”Aleph One”