# What Programmers Should Know About Security

University of Rochester

Matthew Dalton, CISSP

## Goals

- Brief Overview of Programming Practices
- Major Pitfalls in Secure Programming
- Examples (if there is time)
- Q & A
- Additional Resources

## Credits

- Many of the areas and concepts of this presentation are taken from “Building Secure Software” by John Viega & Gary McGraw.  I would suggest buying it if you are really interested in learning more on the subject.

## Guiding Principles for Software Security

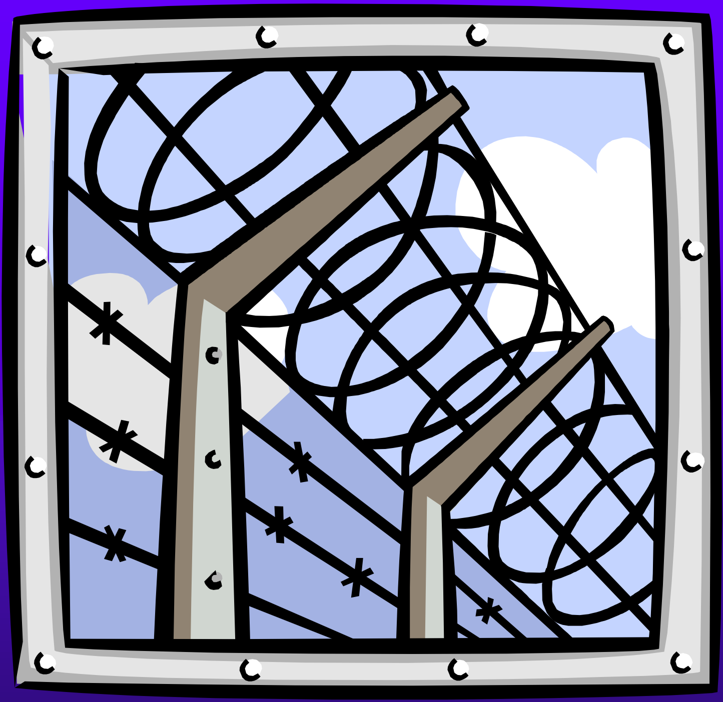
- Secure the weakest link
- Practice defense in depth
- Fail securely
- Follow the principle of least privilege
- Compartmentalize
- Keep it simple
- Promote privacy
- Remember that hiding secrets is hard
- Be reluctant to trust
- Use your community resources

## Secure the weakest link

- Your system is only as secure as the least secure component. The same is true for software.

## Practice Defense in Depth

- Just as in designing a good system, good software doesn't rely on just one mechanism to say that it is safe.
- You may have authenticated, but can you trust the user?
- The information the client sent may not always be the information that the server receives.

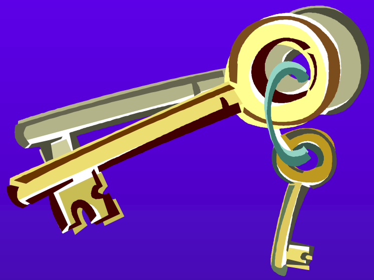


## Fail Securely

- When your system fails, make sure that it fails in a secure manner.
- For authentication, it’s guilty until proven innocent.

## Follow the Principle of Least Privilege

- Only give the privileges that you need to get the job done
- Only keep privileges for as long as you need them



## Compartmentalize

- When you are writing your code, section it off when logically possible
- This is harder than it sounds

# Keep it Simple

- If software is simple, it is much easier to check for security
- Security should be an Opt out, not Opt in solution

## Promote Privacy

- Write your software with the user’s privacy in mind
- If you don’t have to use private data – DON’T
- If possible, only store private data in one location.  This makes it easier to verify that it is only being used for legitimate purposes

## Remember that Hiding Secrets is Hard

- Obfuscation is rarely the best form of security
- When relying on secrets, the crackers almost always win

## Be Reluctant to Trust

- Snake-Oil FAQ – http://www.interhack.net/people/cmcurtin/snake-oil-faq.html
- Don't even trust yourself.  Make sure to have your code reviewed by another person

## Use your Community Resources

- When it comes to most aspects of security, don’t think you know more than the rest of the world
- Rely on proven methods and algorithms for known problems

## Major Pitfalls in Secure Programming

- Buffer Overflows
- Access Control
- Race Conditions
- Randomness & Determinism
- Applying Cryptography
- Trust Management & Input Validation
- Password Authentication
- Database Security

## Buffer Overflows

- Aleph One, “Smashing the Stack for Fun and Profit” http://www.insecure.org/stf/smashstack.txt
- Most common source of vulnerability according to CERT

## Access Control

- Use umask appropriately
  - Values such as 022 (world can read) or 066 (only owner can read)
- Use chroot
  - Better than nothing, but be sure and drop privileges
- Windows is similar but often more granular

## Race Conditions

- These occur when something must hold true for a certain period of time
- Temp file redirections, moving links are examples

# Randomness & Determinism

- Computers are deterministic machines and as such have no randomness of their own
- Make sure that your source of entropy is not less random than you need

## Applying Cryptography

- Use it – use it correctly
- Don’t reinvent the wheel

# Trust Management & Input Validation

- Never trust the user to give you the data you want
- Beware of metacharacters such as ; | ../ and many others
- Beware of hex encoding, unicode, or others
- If you have taint checking, use it. If you don’t, use it’s principles

## Password Authentication

- Passwords are only as secure as a user makes them
- An 8 character password can be:
  - Only lower case letters: 2.08 * 10^11
  - Upper and lower case letters: 5.34 * 10^13
  - Alphanumeric: 2.18 * 10^14
  - All characters available (95): 6.63 * 10^15

## Database Security

- Most databases don’t have encrypted channels
- All former rules apply
- Statistical attacks may threaten privacy
- SQL injection attacks can get by many defenses.

## Additional Resources

- http://del.icio.us/dalton42/programming is continuously updated.