CS390S, Week 1: Introduction to Secure Programming

Pascal Meunier, Ph.D., M.Sc., CISSP
January 10, 2007
Developed thanks to support and contributions from Symantec Corporation, support from the NSF SFS Capacity Building Program (Award Number 0113725) and the Purdue e-Enterprise Center

Copyright (2004) Purdue Research Foundation. All rights reserved.
Vulnerability

- A flaw in a system that allows a policy to be violated
- Example policy: the content on a web site is restricted to authenticated users.
- Vulnerability: the web site relies on JavaScript to be executed on the client browser for access control
- Exploit: Disable JavaScript
- An abundance of vulnerable sites exist
Exploit

- An exploit is the act of exercising a vulnerability.
- Also used to refer to an actual program, binary or script that automates an attack.
Latent Vulnerabilities

- Sometimes, a vulnerability can be protected by a change that leaves the vulnerable code in place:
  - some change external to the application (firewall)
  - a configuration change (disabling an option)
  - a code wrapper that blocks exploit attempts

- A vulnerability that is not exploitable at the moment is a latent vulnerability
Potential Vulnerabilities

- Bad practices, quality defects and other flaws that could result in vulnerabilities in a different code context are potential vulnerabilities.
Exploitability

- Difficult to establish whether a vulnerability is \textit{exploitable}, \textit{latent} or \textit{potential} in complex systems

- A latent or potential vulnerability can become exploitable when
  - The software is used in a different context sometime after its design
  - The configuration is changed
  - The code is changed
  - Someone thinks of something you didn't

- Is a memory leak exploitable?
  - It depends!
Exposure

- An exposure is an information leak that may assist an attacker.
- Examples:
  - Software identification and version number released when connecting to a service, which may be used to select the most effective attack.
  - When web pages display SQL error messages
  - When an IT person is having trouble (e.g., with their firewall) and posts questions to public mailing lists with their company's email address
  - Sun's "tar" utility disclosed part of the password file
Exercise

- Identify other examples of information leaks that may assist an attacker
Exercise Sample Answers

- Identify other examples of information leaks that may assist an attacker
  - Finger may release information about who is online, e.g., administrators
  - Source code leaks (especially if the code contains vulnerabilities)
  - Directory listings
  - Wireless networks that broadcast their existence
A security objective is a high-level description of what the program or system must accomplish.

- Federal regulations drive many of these objectives
  - HIPAA (Health Insurance Portability and Accountability Act)
  - etc...
- Examples:
  - all money transfers must be legal
  - the system must pass EAL4 Common Criteria certification
Security Requirements

- **Functional**
  - Logs
  - Access control
  - Intrusion detection

- **Assurance**
  - Methodology used to create the software
  - Programmer training (secure programming)
  - Source code audit for vulnerabilities
  - Tools used to find vulnerabilities
  - Architecture providing security advantages
Policies specify which activities, states and processes are allowed.

- Examples:
  - All users must be authenticated
  - Money transfers can only be requested by the account owner

- Also refers to security models that specify rules

- Famous policies (e.g., see Bishop 2002):
  - The Bell-LaPadula confidentiality model
  - The Biba integrity model
  - The Clark-Wilson integrity model
Risks to Confidentiality, Integrity and Availability

- Confidentiality is threatened when information can be revealed in violation of a policy
  - Examples: eavesdropping and inadequate access control
- Integrity is threatened when information can be manipulated by an attacker.
  - Example: "man-in-the-middle" attack
- Availability is threatened when a resource can be disabled or made unavailable.
  - Assets can be classified according to these
Example

- An FTP server is read-only. If passwords are sent in clear text, what is threatened if transmissions are captured?
  - Confidentiality of the passwords
    - Confidentiality of the documents on the FTP server
    - Confidentiality, Integrity and Availability of other resources that use the same password!
Question

- Privacy fits best into which category?
  a) Confidentiality
  b) Integrity
  c) Availability
Question

- Privacy fits best into which category?
  a) **Confidentiality** (not a perfect fit)
  b) Integrity
  c) Availability

- Note that some organizations prefer to put emphasis on privacy separately from CIA (e.g., Purdue University's Security and Privacy office). Also, privacy advocates consider it important to be able to verify the integrity of personal information, especially when that information can be used against them (e.g., credit reports).
People Want Software That:

- Is produced with security assurance
  - Analogy: some applications exposed to the internet are like disguised cartons of eggs on the sidewalk

- Lowers security risks
  - To comply with laws mandating low security risks
    - HIPAA
    - GLBA (Gramm-Leach Bliley)
    - FERPA
  - To protect trade secrets and other valuable company information

- Has fewer maintenance headaches (patching) and costs
- Protects their reputation
Exercise

- Identify risks that would cause you to stop using a product. Be specific.
Exercise Example Answers

- Frequencies of vulnerabilities and patches
- Absence of patches (or slow turnover) for known issues
- Severity of vulnerabilities
- Criticality of the application
- Unreliability of patches
  - Patches that break previous fixes
  - Patches that are incompatible with other software
  - Downtime while applying patches
- Unreliable file systems (non-journaled)
- Which matter most (for whom)?
Your motivation as a participant in software development

- How important is quality?
  - Quality assurance is inclusive of secure programming techniques

- How much design?
  - Information assurance happens by design

- How risk-averse?
  - Security problems in your projects and code can hurt your reputation as well as your employer's
Cost of Patching

- Cost of evaluating vulnerability claims
- Cost of patch development and testing
- Cost of patch notification and download system
- NIST recommendation on applying patches (s.p. 800-40)
  - Patch and Vulnerability Group (customer's cost)
    - test patches
    - notify administrators
    - monitor application of patches by system administrators
- Vulnerability scanning to verify or enforce
**Question**

- Patches incur costs to?
  a) the vendor
  b) the customer
  c) both
Question

- Patches incur costs to?
  a) the vendor
  b) the customer
  c) both
Cost of Patching vs Preventing Flaws

- Security bugs are introduced at 3 different stages:
  - Design and architecture
  - Implementation
  - Operations
- Fixing security bugs with a patch costs 60 times more than catching them at design time*

Question

- If it costs $100,000 to issue each security patch, approximately how much could have been saved by correcting the problem at design time?
  a) $9,800
  b) $98,000
  c) $980
If it costs $100,000 to issue each security patch, approximately how much could have been saved by correcting the problem at design time?

a) $9,800  
b) $98,000  
c) $980

Note: It isn’t possible to catch all security flaws at design time.
Results from Current Software Engineering Methods

- Several thousand vulnerabilities reported each year
- About 50% of vulnerabilities are commonly repeated mistakes
- About 25% of vulnerabilities could be avoided by applying secure design principles at *design* time
- Need new methods
- Patches created using the same development methods that created the buggy software, are likely buggy themselves!
  - “We can't solve problems by using the same kind of thinking we used when we created them.” (Albert Einstein)
Question

- Approximately what percentage of documented vulnerabilities are common repeated mistakes?
  a) 25%
  b) 50%
  c) 75%
Question

- Approximately what percentage of documented vulnerabilities are common repeated mistakes?
  a) 25%
  b) 50%
  c) 75%
Question (guess)

- How much money does a developer for a large software project typically save a company when catching and fixing a vulnerability during development instead of patching?
  a) $1,000
  b) $10,000
  c) $100,000

- A vulnerability fixed before release may take one hour, compared to weeks of several people's time to fix it after release.
Answer

- c) $100,000

- Without counting
  - 1) costs to customers
    - Especially if revenue-generating activities are interrupted!
  - 2) intangible costs
Patches Don't Work Well

- Vendors can take a long time to issue patches
  - Risk: Malicious researchers can find the vulnerability independently
    - More and more likely to happen due to the "birthday effect"
- People are late applying patches
  - Afraid of breaking working mission critical things
  - Requires time, testing
  - Difficult to verify compliance in a large federated enterprise
- Patches give excellent information to attackers
- Patches don't repair exploited systems
Today's Challenge

- Vendors, Academia and Government need as many people trained in secure programming as possible
  - That means you ;)
- Race between "good" and "bad" guys in finding vulnerabilities first isn't good
- Need to produce software with as few security issues as possible right from the start