CSCI 1800 Cybersecurity and International Relations

Security Through Software Engineering

John E. Savage

Brown University
Outline

• Security modeling including access control
• Federal security regulations and standards
• Software vulnerability assessments
• Microsoft’s Security Development Lifecycle
• Introduction to Threat Analysis
Policy, Models and Trust

• To have secure systems, engineers need
  – Clear security goals
  – Effective implementation strategy

• A security policy puts constraints on actions that can taken by actors on objects in the system in order to achieve security goals.
Security Policy Components

• **Actors**
  – Individual or group agents interacting with a system.

• **Objects**
  – Informational/computational resources affected by policy.

• **Actions**
  – Possible modifications to objects, e.g. read, edit, copy, remove

• **Permissions**
  – Rules constraining actions that actors may take on objects.

• **Protections**
  – Features of policy, e.g. confidentiality, availability, anonymity.
Security Model

• A **security model** is an **abstraction** providing **conceptual language** to specify **security policies**.
  – E.g. Unclassified (U), Confidential (C), Secret (S), Top Secret (TS)
  – **Special compartments** for special compartmentalized information (SCI), such as human intelligence (HUMINT), satellite observations (GEOINT)
Two Models of Access Control

• **Discretionary access control**
  – Owner may specify permissions on files
  – A more relaxed form of control

• **Mandatory access control**
  – Administrator fixes permissions in advance.
  – More strict control

• **Rules have subjects** (parties requesting access) and **objects** (those things being accessed).
Bell-LaPadula (BLP) Access Control Model

• Applies to confidentiality – dates from the 1970s
• Object x and user u have security levels L(x) & L(u)
  – Examples of security levels: U, C, S, TS
• For users u and v, v has higher clearance than u if L(u) ≤ L(v). u can pass info to v but not vice versa.

• No “read up” (user can’t see more secure data)
  – User u can read x only if L(x) ≤ L(u)
• No “write down” (user can’t use more secure data)
  – User u can write to object x only if L(u) ≤ L(x).
BLP Access Control Model

L(u) ≤ L(v)

BLP model weakness: only handles confidentiality
Ken Biba ('77) Access Control Model

• Goal of Biba’s model is to maintain data integrity:
  – the accuracy and consistency of data over its life-cycle.
• Let $I(x)$ and $I(u)$ be the integrity of user $u$ & object $x$
  – The higher is $I(u)$ or $I(x)$, the more trustworthy or accurate the user $u$ or object $x$ is.
• Don’t read from lower integrity level
• Don’t write to higher integrity level.
  • User $u$ can read object $x$ only if $I(u) \leq I(x)$.
  • User $u$ can write to object $x$ only if $I(x) \leq I(u)$. 
Role-Based Access Control

- **Components:** users, roles, permissions, sessions
  - A role is a collection of users.
  - A session is an interaction for a period of time.

- **Role hierarchy** is defined, as in a corporation.
  - President IsA manager IsA employee
  - Higher role user inherits permissions of lower one
  - When is this **not** a good idea?

- **Role constraints** may be imposed
  - Example: avoid conflicts of interest.
USG Security Standards

  – Division A: system has a formal process for verification of security
  – Division B: mandatory access control
  – Division C: discretionary access control
  – Division D: minimal protection criteria
USG Security Standards

  - It subsumes the Orange Book
  - Defines key concepts related to security evaluations
  - Framework for documenting security goals
  - Not a certification vouching for product security.
Government Regulations & Standards

• Federal Information Processing Standards (FIPS 140-2)
  – Standards for designing/handling of cryptographic modules by US government organizations
  – Last updated in 2002

• Security levels:
  1. No physical security, can run modules on open machine
  2. Some physical security, e.g. tamper-evident coatings, some role-based authentication, trusted OS
  3. Prevent physical tampering, identity-based authentication instead of role-based authentication
  4. Tighter physical security, all keys and messages destroyed when unauthorized attempts to break security are made.
Government Regulations & Standards

• **HIPAA (1996)**
  – Sets privacy standards on patient records for healthcare providers and employers.

• **Family Educational Rights and Privacy Act (FERPA) (‘74)**
  – Requires protection of privacy of educational records in US

• **Federal Information Security Management Act (FISMA)**
  – Revised in 2014 – provides government information security.
  – It requires federal agencies to implement processes and controls designed to ensure the confidentiality, integrity, and availability of system-related information.
  – Must follow FISMA and NIST standards, and legislative requirements, such as the Privacy Act of 1974.
Software Vulnerability Assessment

• The problem: software can be enormous
  – Mac OS X 10.4 has > 86 million lines of code!
  – Code can have both performance & security bugs

• “A vulnerability is a security exposure that results from a product weakness ... the product developer did not intend to introduce and should fix once it is discovered.”*

• How many errors per 1,000 lines of code (KLOC)?
  – Estimates vary from 5-50 defects per KLOC
  – Some open-source have bug density of .4 bugs/KLOC!†

* Microsoft definition
Lect 20 4/11/18
Approaches to Vulnerability Assessment

• **Black-box analysis**
  – Penetration test (pentest) done without knowledge of innards.
  – Pentests look for security vulnerabilities

• **White-box analysis**
  – Same but with full knowledge of hardware/software, network environment, etc.
Code Analysis for Privacy/Security

• Problem: Cybercrime is here to stay.

• Goal: Find and remove privacy/security hazards.
  – Static code analysis studies non-running programs.
    • This is white-box testing of non-running code. E.g.
  – Dynamic analysis examines running programs.

• Good analysis requires training and investment
  – Software engineers generally not educated on this.
  – Microsoft’s Security Development Lifecycle (SDL) represents a big step forward.

• Benefits: Improved security, privacy and reliability.
Components of Static Code Analysis*

• Data Flow Analysis
  – Based on analysis of basic blocks, sections of code in which control stays within a block during execution

• Control flow graph (CFG)
  – Shows all possible control paths

• Taint analysis
  – It identifies variables touched by users or “tainted”
  – Influence of tainted variables on the CFG and actions

* https://www.owasp.org/index.php/Static_Code_Analysis
Example of a Basic Block

• 1. $a = 0;
• 2. $b = 1;
• 3.
• 4. if ($a == $b)
• 5. { # start of block
• 6. echo “a and b are the same”;
• 7. } # end of block
• 8. else
• 9. { # start of block
• 10. echo “a and b are different”;
• 11. } # end of block

Note: # starts a comment
Example of Data Flow Graph

node 1

node 2
node 3
node 5
node 6

node 4

node 7
node 8
node 9
Dynamic Code Analysis

• Good analysis explores all important paths
  – Requires good choice of test data
  – Incomplete testing can result in catastrophic failure
• “Fuzzing” can reveal hidden errors
  – User tries inputs of length 1, 2, 3, ... until some limit
• Note: malware may detect it is being run in a virtual environment and not activate
• Many tools exist for dynamic software analysis
Top Incidents in 2016 Verizon Report*

Incidents representing more than 90% of data breaches:

1. Insider and privilege misuse
2. Cyber-espionage – attacks by external actors hunting for data & trade secrets
3. Web application attacks
4. Crimeware – malware incidents, typically opportunistic and financially motivated in nature (e.g., banking Trojans, ransomware).
5. Point-of-sale (POS) Intrusions
6. Denial of service (DoS) Attacks
7. Payment card skimmers – tampering of ATMs and fuel terminals.
8. Physical theft and loss of data or IT-related assets.

Information Security Attributes

The CIA Triad

- **Confidentiality**: Access to information is limited to those with proper authorization.

- **Integrity**: Maintaining the consistency, accuracy and trustworthiness of data during its life cycle.

- **Availability**: Reliable access is maintained to resources by authorized parties.
Modeling System Threats

• *Data flow analysis* preferable to *exploring assets or motivations of attackers*.

• Group components by *trust boundaries*
Microsoft’s Security Development Lifecycle (SDL)

• Bill Gates inaugurated Microsoft’s Trustworthy Computing Initiative in 2002.
  – Success with major new corporate initiatives often requires support from top management.
• Every product that impacts privacy or may be used by children needs security analysis.
  – This means almost all hardware/software products
• Average cost of fixing a security bug was $300K!
• Microsoft code is now among the most secure!
### Microsoft’s SDL

<table>
<thead>
<tr>
<th>Training</th>
<th>Requirements</th>
<th>Design</th>
<th>Implementation</th>
<th>Release</th>
<th>Verification</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core Security Training</td>
<td>Establish Security Requirements</td>
<td>Establish Design Requirements</td>
<td>Use Approved Tools</td>
<td>Incident Response Plan</td>
<td>Dynamic Analysis</td>
<td>Execute Incident Response Plan</td>
</tr>
<tr>
<td></td>
<td>Create Quality Gates / Bug Bars</td>
<td>Analyze Attack Surface</td>
<td>Deprecate Unsafe Functions</td>
<td>Final Security Review</td>
<td>Fuzz Testing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Security &amp; Privacy Risk Assessment</td>
<td>Threat Modeling</td>
<td>Static Analysis</td>
<td>Release Archive</td>
<td>Attack Surface Review</td>
<td></td>
</tr>
</tbody>
</table>

- Personnel must be trained.
- Security requirements, risk assessment needed
- Must do threat modeling (STRIDE) and reduce attack surface.
  - **Spoofing**, **Tampering**, **Repudiation**, **Information Disclosure**, **Denial of service**, **Elevation of privilege**.
- Implementation requires good tools to protect against attacks
- Must plan for post-release handling of errors
- Verification needed via dynamic analysis including fuzzing.
STRIDE Threats*

- **S** – Spoofing
- **T** – Tampering
- **R** – Repudiation
- **I** – Information Disclosure
- **D** – Denial of Access or Service
- **E** – Elevation of Privilege

* Microsoft’s mnemonic for types of software threats
STRIDE Explained

• S – pretending to be another person or thing
• T – modifying something one should not
• R – falsely claiming not to have taken an action
• I – exposing information to those unauthorized
• D – denying users access to a service
• E – acquiring access at an elevated level
## STRIDE Elaborated

<table>
<thead>
<tr>
<th>Threats</th>
<th>Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spoofing</td>
<td>Authenticity</td>
</tr>
<tr>
<td>Tampering</td>
<td>Integrity</td>
</tr>
<tr>
<td>Repudiation</td>
<td>Non-Repudiation</td>
</tr>
<tr>
<td>Information Disclosure</td>
<td>Confidentiality</td>
</tr>
<tr>
<td>Denial of Service</td>
<td>Availability</td>
</tr>
<tr>
<td>Elevation of Privilege</td>
<td>Proper Authorization</td>
</tr>
</tbody>
</table>
Options to Address Threats (META)

• Mitigate a threat
  – Increase the work to exploit it

• Eliminate a threat
  – Usually requires elimination of features

• Transfer of a threat
  – Let some other system element cope with it

• Accept a threat
  – Risk acceptance may be less costly than other steps
Source Material

Is Open Source Software a Panacea?

• Software is available for modification under liberal copyright policy.

• Do many eyeballs on the code make it secure?
  – "... in reality that doesn’t happen" Cowan 2002.

• Russia believes it – avoids US software.
  – Putin orders Russian government to move to Open Source Software by 2015. (12/28/2010)

• Problems: No incentive to find bugs. Coders not trained to find them. It is hard!

Review

- Security modeling including access control
- Federal security regulations and standards
- Software vulnerability assessments
- Microsoft’s Security Development Lifecycle
- Introduction to Threat Analysis