Chapter 4: Software Security Basic Goals: Specifying Security Requirements Using UML

BIE 33003 Software Engineering Security

Acknowledgments

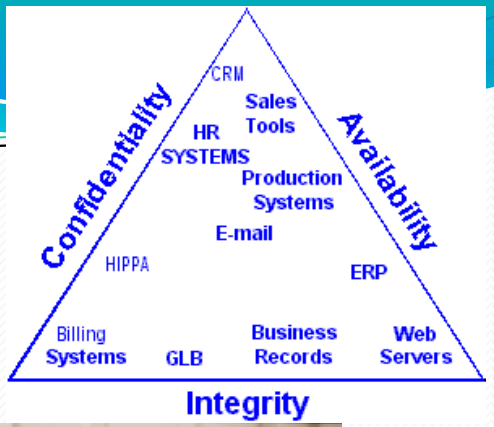
References are provided per page. Most diagrams are original, but ideas are adapted from references.

Author: Susan J Lincke, PhD
Univ. of Wisconsin-Parkside
Contributors/Reviewers:
Tim Knautz, Janine Spears PhD, David Green PhD, Megan Reid

Funded by National Science Foundation (NSF) Course, Curriculum and Laboratory Improvement (CCLI) grant 0837574: Information Security: Audit, Case Study, and Service Learning.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and/or source(s) and do not necessarily reflect the views of the National Science Foundation.

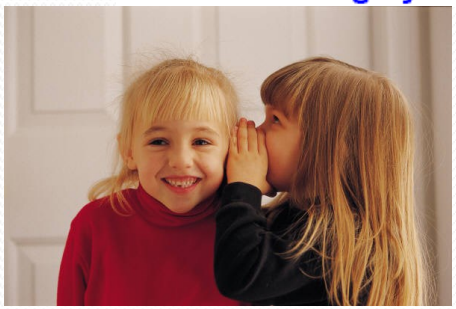
Security Assures … CIA

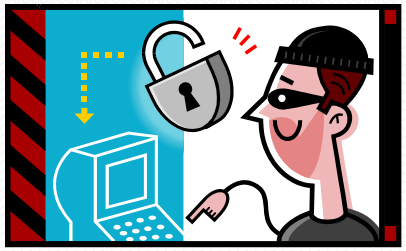


Confidentiality: Limits access of authorized users and prevents access to unauthorized users

Integrity: The reliability of information resources and data have not been changed inappropriately

Availability: When something needs to be accessed by the authorized user, it is available





Security Vocabulary

Asset: Diamonds

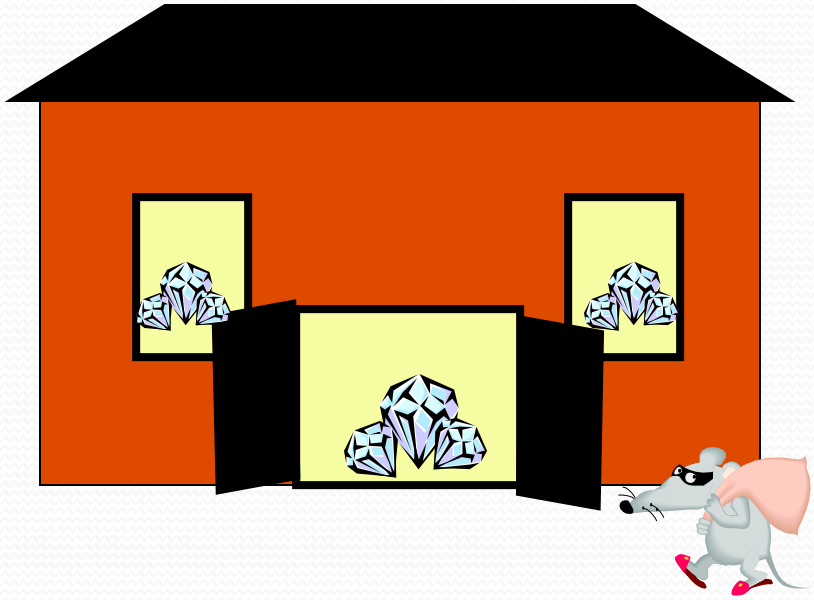
Threat: Theft

Vulnerability: Open door or windows

Threat agent: Burglar

Owner: Those accountable or who value the asset

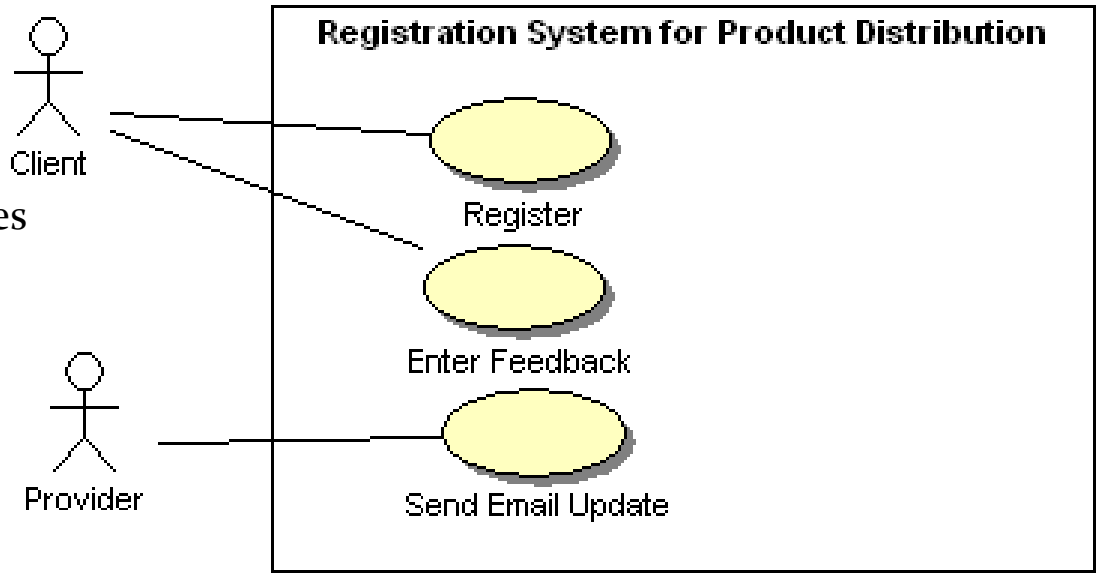
Risk: Danger to assets



Registration System Use Case

Register: Clients register to obtain documentation by providing name, email, job function

Provider: Send periodic updates to Clients to indicate changes in materials



OCTAVE

- OCTAVE stands for Operationally Critical Threat, Asset, and Vulnerability Evaluation.
- It was developed by the computer Engineering Institute (CEI) at Carnegie Mellon University. It is a qualitative risk analysis methodology

OCTAVE Security Requirements Process

Risk: Threat and vulnerability(s) -> negative impact

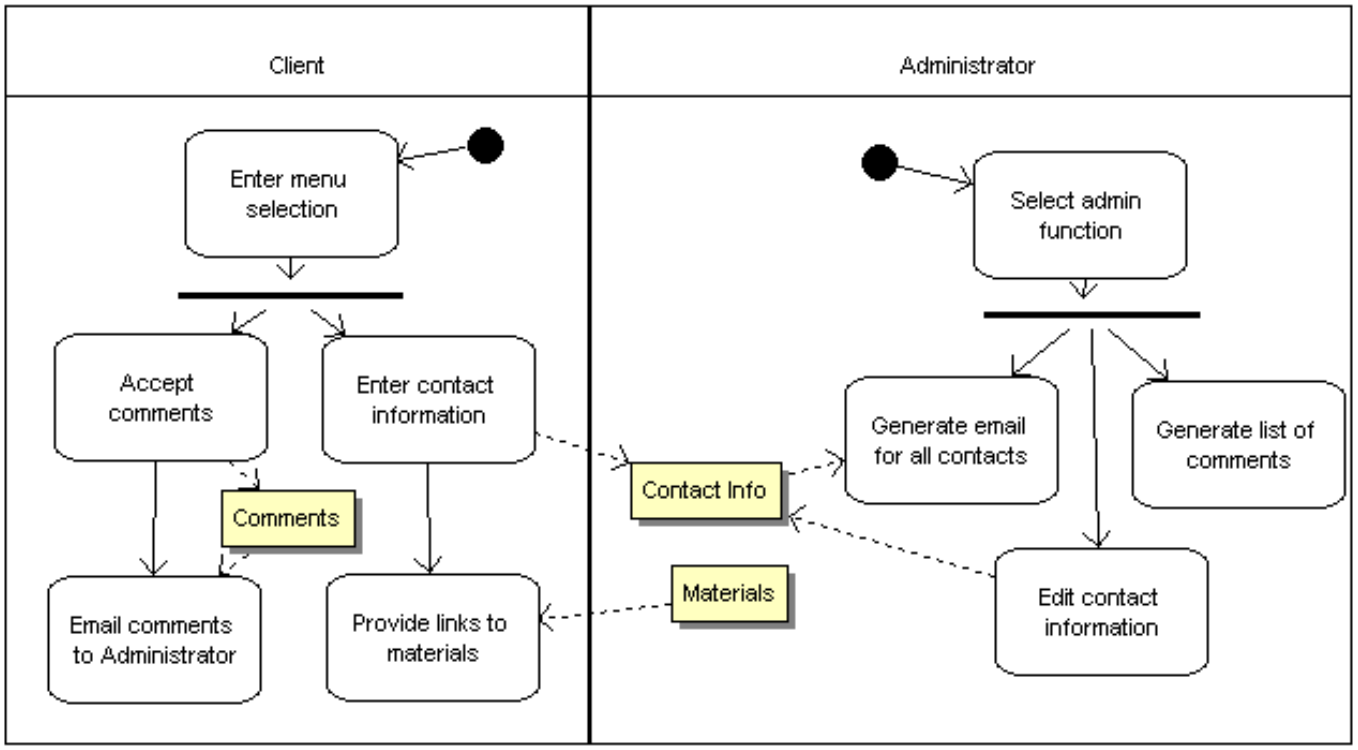
1. Identify critical assets
2. Define security goals
3. Identify threats
4. Analyze risks
5. Define security requirements

Identify Critical Asset

- For any software development effort typical assets will be sensitive or business critical information that the application displays, stores, processes, modifies, or transmits. However, assets in a Software Development effort may also include such items as business rules, external databases, encryption keys, authentication processes, passwords, or methods used in the secure transmission of data. These types of assets include but are not limited to the following:
  - Information assets
  - Business Rules
  - Services or functions
  - Software
  - Propriety formulas
  - Encryption methods and keys
  - Databases
  - People, or specifically the knowledge or skill set possessed by individuals

Step 1. Identify Critical Assets
via Business Process Diagram

- **Contact Info**: Name, email, job function
- **Materials**: Course materials
- **Comments**: Feedback, saved & sent as email



Step 2. Define Security Goals

Assets	Confidentiality	Integrity	Availability
Contact Info	** No PII maintained	*** Require accurate list of interested persons	* Weekly backup
Materials	* Public with login	*** Accurate – tamper-proof	** 24/7 preferred
Comments	** Confidential pref.	*** Accurate – tamper-proof	* Weekly backup, email

Impact Rating:
* Low Priority
** Medium Priority
*** High Priority

Identify Threat

- In a typical software development effort there are two types of threats to consider, business threats and system threats.
- Business threats: are threats to the business function that may cause disruption in or damage to the business function or supporting business and resources.
- System threats: are usually much easier to recognize during software development efforts. A system threat is a direct threat to an application or one of its subsystems.

- Human threat represent threats of systems based upon the behaviour of individual

Step 3: Identify Threats

STRIDE General Threats

	What it is	Software Techniques	Advanced Security
Spoof Identity	Impersonation Social Eng.	Encrypt passwords No backdoor entry	Secure Password Digital Certificate Sec. Awareness
Tamper w. Data	No Integrity / Fraud	Validate input Stop buffer overruns Close unused resources	Authentication Access Control Message Digest
Repudiation	"I did not do that!"	Require passwords Log user transactions	Trans. Logging Digital Signature
Info Disclosure	No Confidentiality	Encrypt data Encrypt packets Minimal permissions	Encryption Authentication Secure Handling
Denial of Service	No Availability System/Network /DB Full	Validate real user via CAPTCHA Test for Failures	Firewall Intrusion Prevention System
Elevation of Privilege	Exceed Authority	Take care with error messages Hide system files Require approval	Access Control Segregation of Duties

Step 3. Identify Threats via Misuse Case Diagram

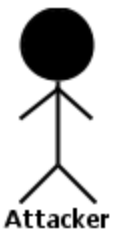
Which misuse cases relate to:

- Confidentiality?
- Integrity?
- Availability?

Definitions:

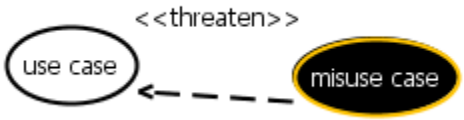
DOS = Denial of Service

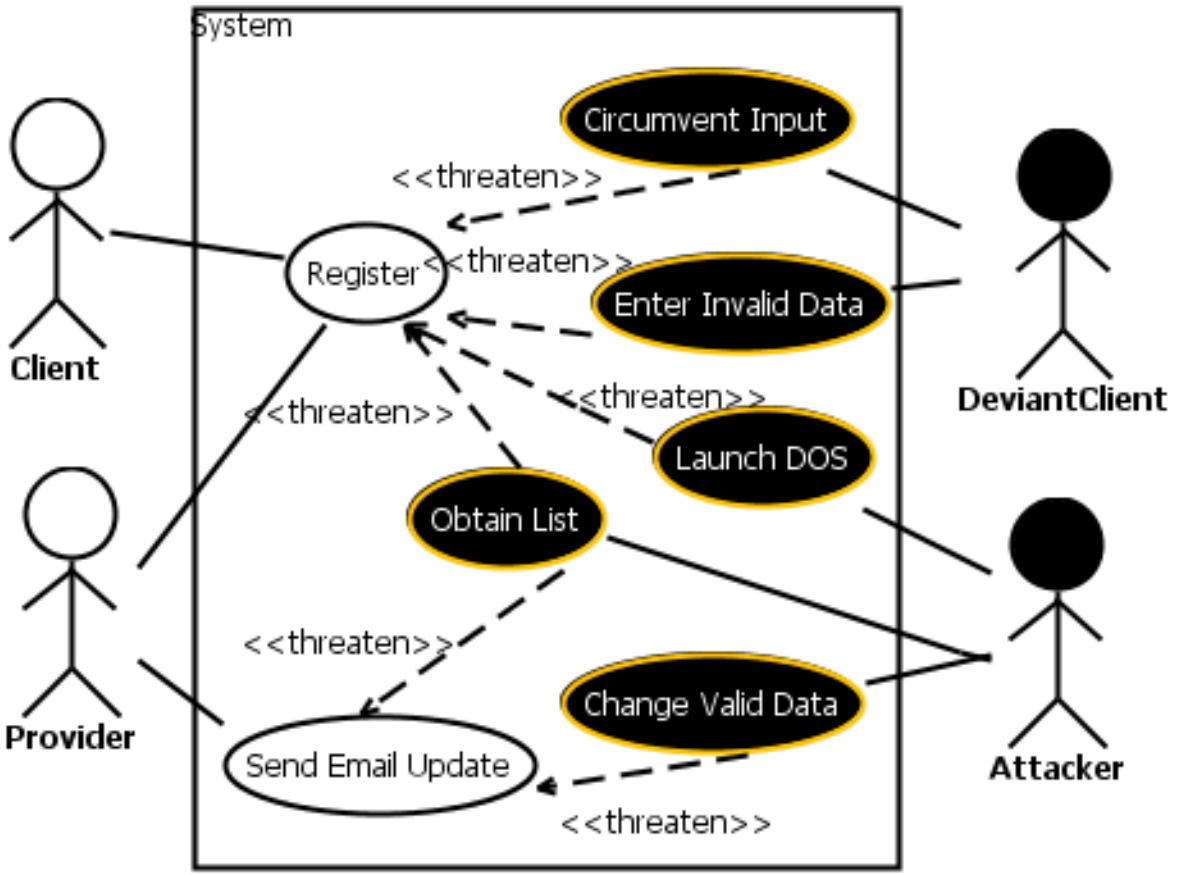
←misuser



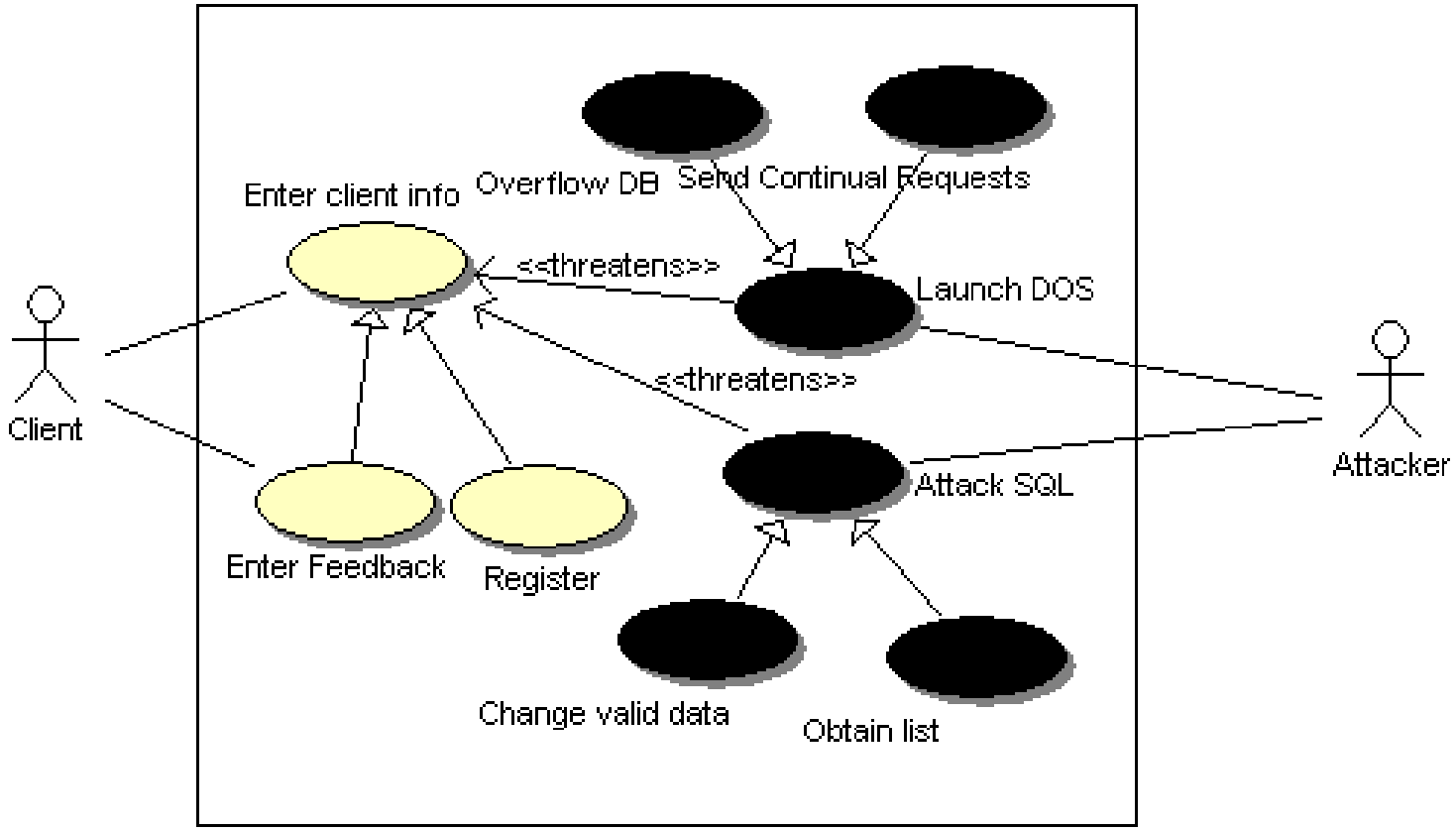
Misuse case

↓



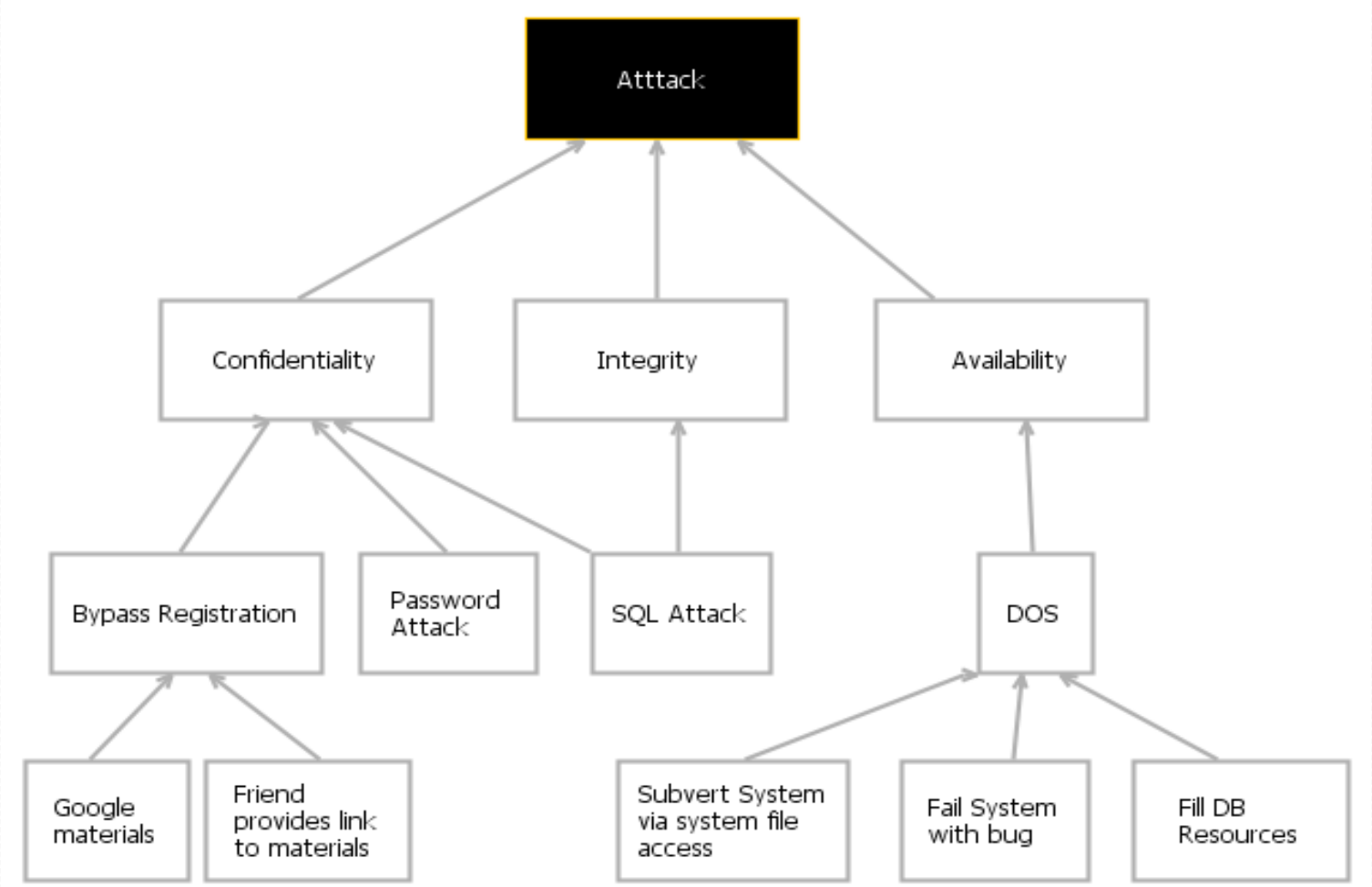


Step 3 (cont'd):
Expand DOS Misuse Case



Overflow DB: Fill disk with records
Send Continual Requests: (Distributed Denial of Service) No processor remains

Step 3 (optional)
Threat Tree



Step 3 cont'd: Lightweight Misuse Case:

Change Valid Data

User Intention	System Response	Security Threat
User requests Reg. form	System provides form	
Attacker enters form input but appends additional SQL commands		SQL injection
	System processes input	
		Obtain (client) list Change valid data

Step 3 Cont'd: Mid-weight Misuse Case

DOS

Misuse Case: Denial of Service
Summary: An attacker issues repeated Registrations, resulting in filling the database with fake data, and depleting system and file resources.
Basic Path: 1. Do forever 2. The attacker requests a Registration form 3. The attacker sends random fake data in the form 4. Enddo
Alternative Paths: AP1. Repeat data is entered
Mitigation Points: MP1. At BP Step 2-3 use CAPTCHA in Registration form to avoid bot attack. MP2. At BP Step 3 validate data: no duplicates, data type matching

Step 3 Cont'd: Mid-weight Misuse Case:

Circumvent Input

Misuse Case: Circumvent Input
Summary: Deviant Client bypasses registration by going directly to the download web page.
PreCondition: Client does Google search and finds link to download web page OR obtains link reference from a colleague
Basic Path: 1. DeviantClient obtains web reference from Google or friend. 2. DeviantClient uses web reference to download materials without registering.
Mitigation Points: MP1: Web page has no other web references. MP2: Create dynamic web page with unique reference. This web page is accessible only if a key is provided during registration. Key expires in one week.
Related Business Rule: Users must register to obtain materials.
Mitigation Guarantee: MP1 and MP2 solves Google search problems. MP2 could be used by friends for one week, which is acceptable.

Step 4: Analyze Risks

Threat	Impact	Likelihood	Priority = I*L
DOS	***	***	9
SQL Attack (affects integrity, confidentiality)	***	***	9
Invalid Input	*	***	3
Circumvent input	**	***	6

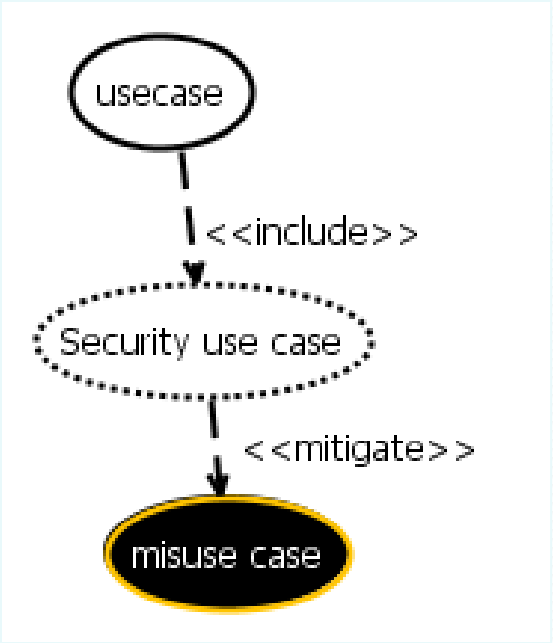
This is straight from risk management.
* = low priority = 1
** = medium priority = 2
*** = high priority = 3
Priority = Impact * Likelihood

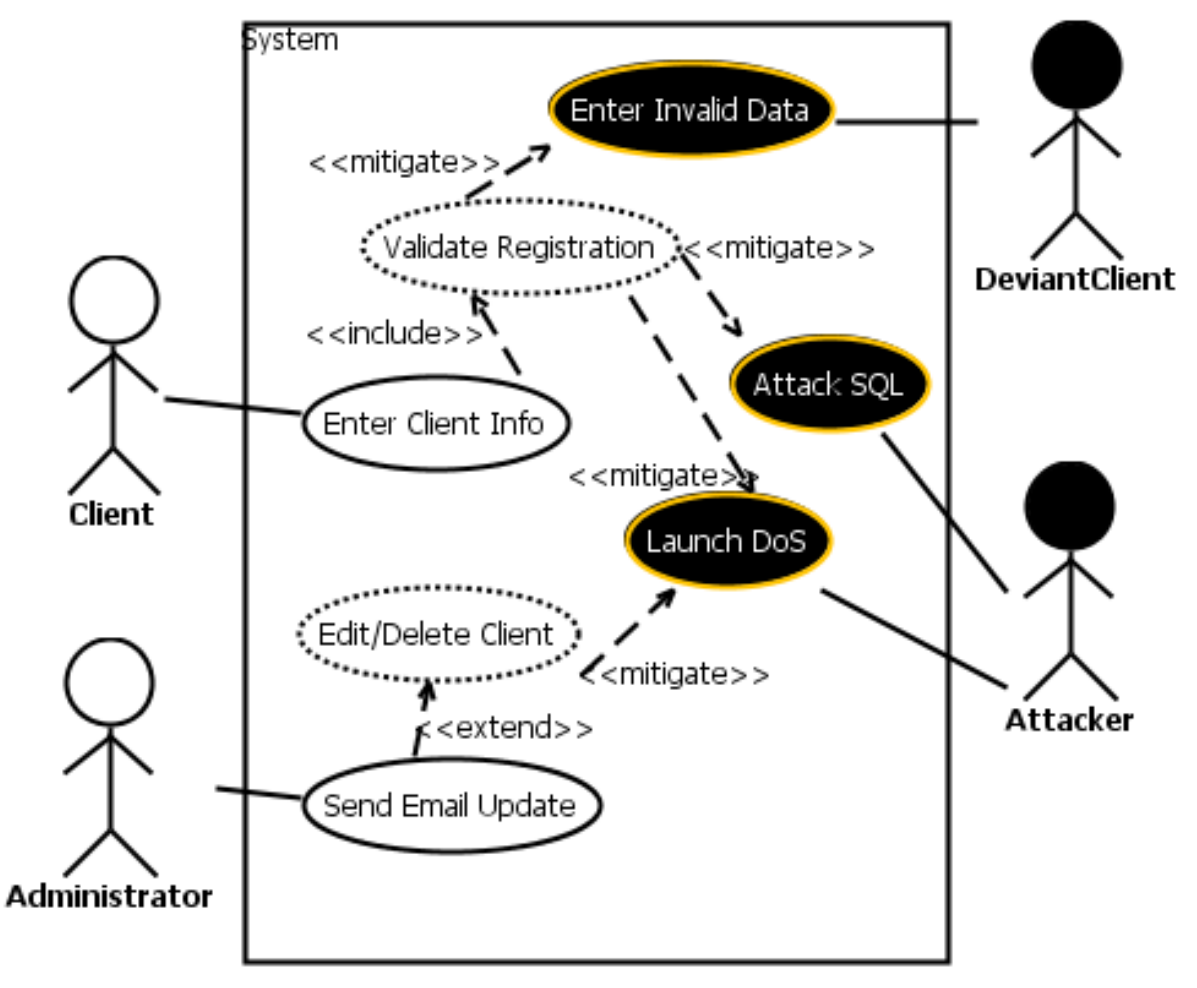
Priority Levels

0 to <3	LOW
3 to <6	MEDIUM
6 to 9	HIGH

Step 5:
Define Security Requirements

Definitions





Stage 5: Define Security Requirements

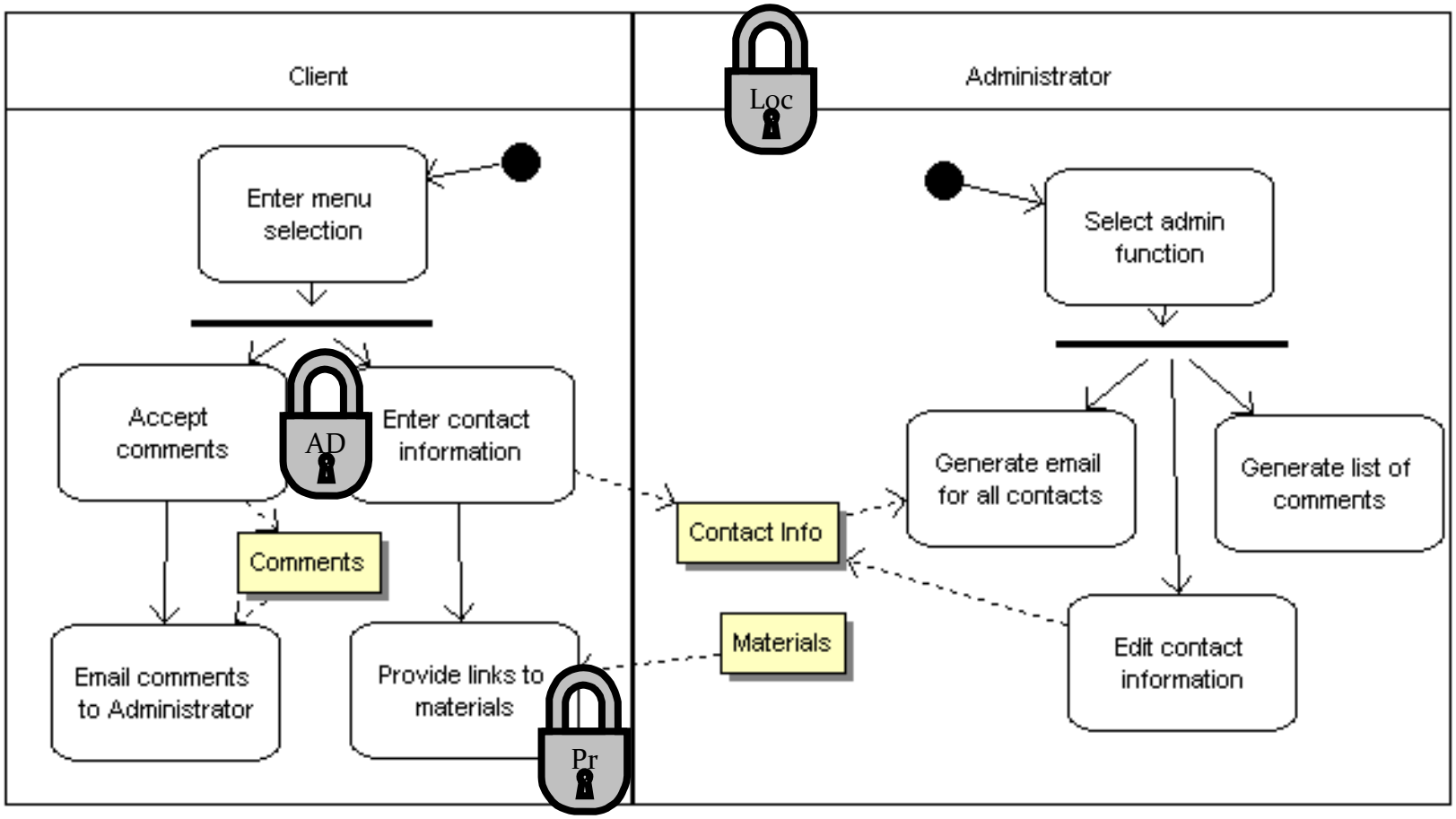
Modify Register Use Case Desc.

Use Case: Register
Summary: Client registers to obtain access to download materials.
Preconditions: Client is at Welcome Web Page
Basic Path: 1. The client selects the Obtain Materials link. 2. The system asks the client for name, email address, job function, and CAPTCHA. 3. The client enters all three required information. 4. *Include (Validate Registration)* 5. The system displays the URL for the download materials.
Alternative Path: AP1. If an attack is detected, no URL is displayed.
Postcondition: The client has access to the download materials. The database contains the client contact information.

Stage 5: Define Security Requirements: Validate Registration Security Use Case

Use Case: Validate Registration
Summary: This include validates a registration.
Precondition: A name, email, job function, and Captcha are provided.
Basic Path: 1. The user enters a name, email, and job function in Step 3 of Register 2. Do until valid CAPTCHA. 3. Rerequest form with new CAPTCHA 4. The system checks for valid characters, to prevent SQL injection. 5. The system checks for valid name, email and job function 6. If email is unique in database 7. Save record to database 8. The system returns success.
Postconditions: The input has been checked for bot attempt, SQL attempt, and validity.

Business Process Diagram Enhancement



Loc
Local Access

AD
Attack Detection

Pr
Privacy

To be continued: