**[Homework 12-13](https://github.com/hendraanggrian/IIT-ITM511/blob/assets/assignments/hw12-13.pdf): Safety and security engineering**

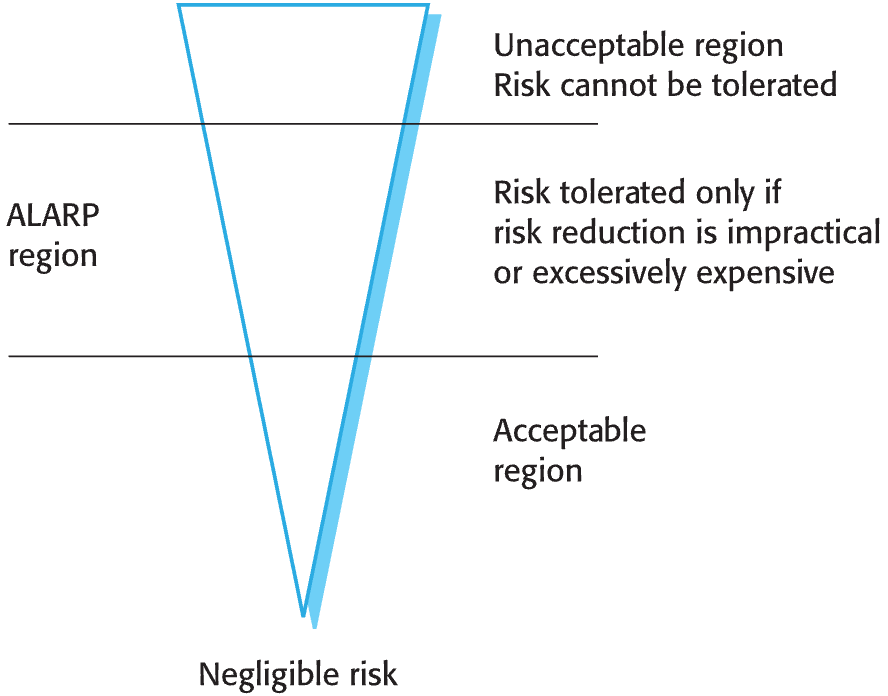
**Problem 1**

*Identify six consumer products that are likely to be controlled by safety-critical software systems. Explain your reasoning for each.*

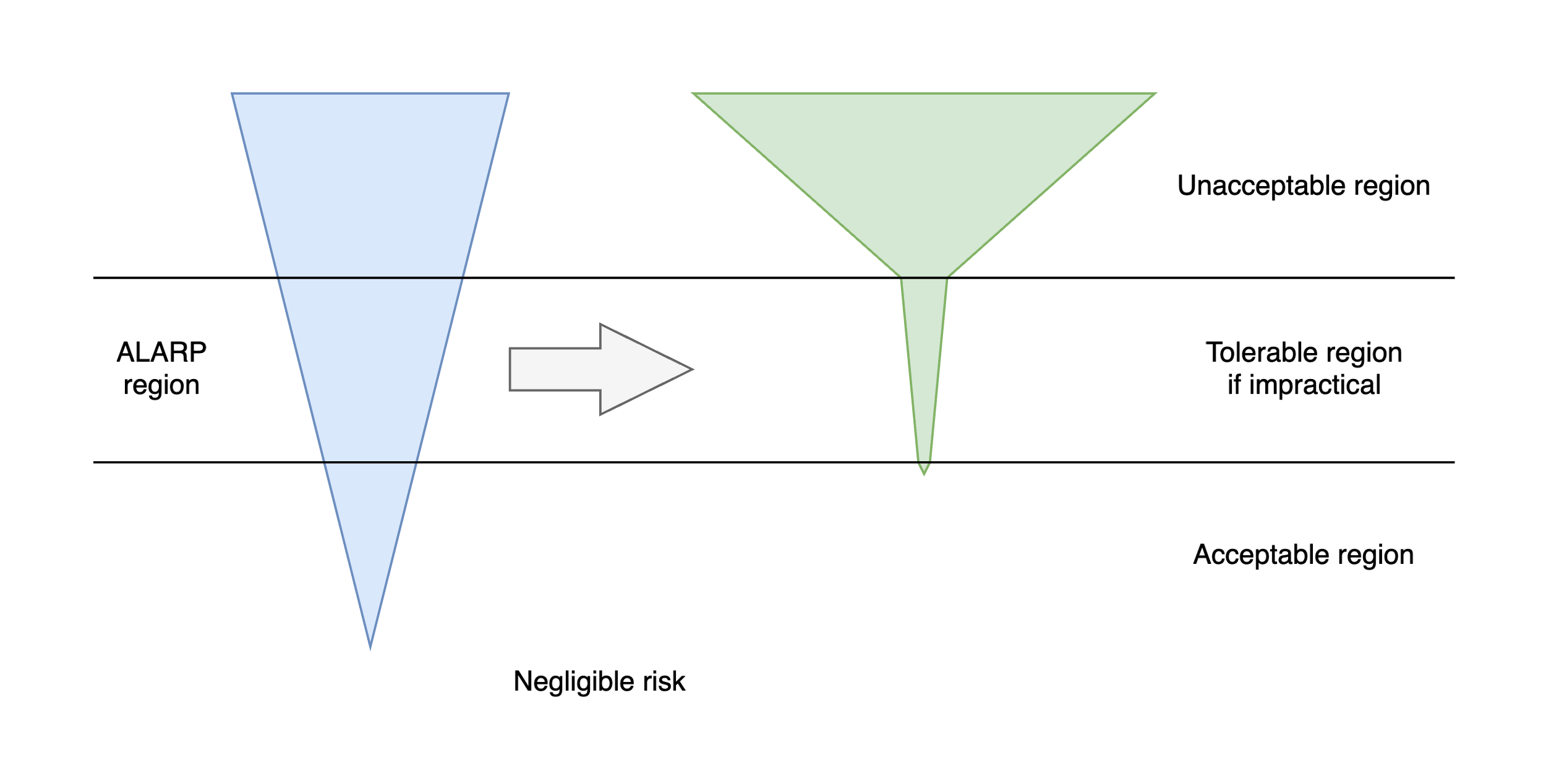
1. **Banking website:** Financial institutions are wary of potential security breaches causing monetary damage and loss of public confidence. Banking websites typically have multiple guardrails like connection encryption and valid security certificates.
2. **Railway scheduler:** Whether carrying passengers or freight, trains depend on a centralized command control that manages their timetable. These rail systems are essential to dodge collision and derailment.
3. **Elevator platform:** Modern elevators are controlled by software that prevents the platform from being used when the maximum weight limit is reached. The software should also check the health of safety-critical machinery such as motors, doors and electric wiring.
4. **Semiconductor device:** Processors and system-on-chip will automatically lower clock speed to protect the chipset from overheating. Thermal throttling is also used to improve the device's battery life.
5. **Handheld drone:** Safety-critical software of drones ensures that it cannot be flown above a predetermined altitude for flight stability. The aircraft should also return to its base in case of a low battery to avoid a crash.
6. **Emergency alarm:** Hazard-detection systems like fire, smoke and home-invasion alarms require immediate response. A software malfunction could have devastating consequences for the environment.

**Problem 2**

*A software system is to be deployed for a company that has extremely high safety standards and allows for almost no risks, not even minor injuries. How will this affect the look of the risk triangle in* ***Figure 12.3*** *in the textbook?*

**

***Figure 12.3*** *The risk triangle*



[View source](https://github.com/hendraanggrian/IIT-ITM511/blob/main/assignments/hw12-13/figure2.drawio)

The unacceptable region of the updated risk triangle will grow substantially due to the company's stance on the safety-first protocol. Meanwhile, the size of As Low As Reasonably Practicable (ALARP) and acceptable regions should shrink, if not completely vanish. In a practical term, the project's budget and deadline will increase to improve the quality of the software.

**Problem 3**

*Explain when it may be cost-effective to use formal specification and verification in the development of safety-critical software systems. Why do you think that some critical systems engineers are against the use of formal methods?*

* **Pro:** The formal software development guidelines are the product of utilization and testing over many years. They have a proven track record of the highest safety standards. The guidelines are suitable for software systems where risk is intolerable, for instance, financial loss, property damage, or life endangerment.
* **Con:** Complying with a safety regulatory standard significantly increases the project's complexity. The formal specification requires extra workloads to perform rigorous tests adhering to the standard. It takes away limited resources and may overwhelm a small software team.

**Problem 4**

*List four types of systems that may require software safety cases, explaining why safety cases are required.*

1. **Air traffic control:** Safe landing is authorized by the pilot report and clearance by air traffic control. Aviation authorities are responsible for risk assessment and establishing safety standards.
2. **Self-driving car:** Autonomous vehicles are preinstalled with sophisticated sensors to measure physical boundaries to navigate their surroundings. Safety cases minimize accidents caused by unpredictable environments.
3. **Manufacturing plant:** Software in a manufacturing plant automates robotic arms, conveyor belts, and other programmable controllers for industrial purposes. Regulatory bodies closely oversee these systems to ensure worker safety.
4. **Defense system:** Military software that controls missile deployment, surveillance, and cybersecurity is regulated by government institutions. They work together in resolving ethical and policy concerns over activities conducted by the software.

**Problem 5**

*The door lock control mechanism in a nuclear waste storage facility is designed for safe operation. It ensures that entry to the storeroom is only permitted when radiation shields are in place or when the radiation level in the room falls below some given value (dangerLevel).*

1. *So,*
2. *If remotely controlled radiation shields are in place within a room, an authorized operator may open the door.*
3. *If the radiation level in a room is below a specified value, an authorized operator may open the door.*
4. *An authorized operator is identified by the input of an authorized door entry code.*
5. *The code shown below controls the door-locking mechanism. Note that the safe state is that entry should not be permitted. Using the approach discussed in this chapter, develop a safety argument for this code. Use the line numbers to refer to specific statements. If you find that the code is unsafe, suggest how it should be modified to make it safe.*

1 entryCode = lock.getEntryCode();

2 if (entryCode == lock.authorizedCode)

3 {

4 shieldStatus = Shield.getStatus();

5 radiationLevel = RadSensor.get();

6 if (radiationLevel < dangerLevel)

7 state = safe;

8 else

9 state = unsafe;

10 if (shieldStatus == Shield.inPlace())

11 state = safe;

12 if (state == safe)

13 {

14 Door.locked = false;

15 Door.unlock();

16 }

17 else

18 {

19 Door.lock();

20 Door.locked := true;

21 }

22 }

1. *Code above as a text box for editing.*

entryCode = lock.getEntryCode();

if (entryCode == lock.authorizedCode) {

shieldStatus = Shield.getStatus();

radiationLevel = RadSensor.get();

if (radiationLevel < dangerLevel) {

state = safe;

} else {

state = unsafe;

}

if (shieldStatus == Shield.inPlace()) {

state = safe;

}

if (state == safe) {

Door.locked = false;

Door.unlock();

} else {

Door.lock();

Door.locked := true;

}

}

While the quoted code meets the requirements **i (line 10**) and **ii (line 6)**, the safety checks are performed in incorrect order. Furthermore, the current check may override the result of the previous check because they are executed individually. To resolve the problem, we need to use nested conditions with the correct ordering.

entryCode = lock.getEntryCode();

if (entryCode == lock.authorizedCode) {

shieldStatus = Shield.getStatus();

if (shieldStatus == Shield.inPlace()) {

radiationLevel = RadSensor.get();

if (radiationLevel < dangerLevel) {

state = safe;

} else {

state = unsafe;

}

} else {

state = unsafe;

}

if (state == safe) {

Door.locked = false;

Door.unlock();

} else {

Door.lock();

Door.locked := true;

}

}

**Problem 6**

*Describe the security dimensions and security levels that have to be considered in secure systems engineering.*

* Security dimensions:

1. **Confidentiality:** Protect sensitive data such as login credentials and personal social security numbers from unauthorized parties (Sommerville, 2016).
2. **Integrity:** Check the validity of data, which may degrade over time or caused by incidents.
3. **Availability:** Limit access to data from suspicious behaviors like cyberattacks.

* Security levels:

1. **Infrastructure security:** Manages internal communication within an organization.
2. **Application security:** Directed at the client device or a distinctive category.
3. **Operational security:** Secures the day-to-day operations of an organization.

**Problem 7**

*Explain why security is considered a more challenging problem than safety in a system.*

Safety risks are often predetermined and predictable, which makes it easier to anticipate relative to security threats. There is also an argument that safety risks can be reduced to an acceptable level with risk-reducing measures, whereas the negative consequences of security flaws may vary (Leopold, 2018). Moreover, attackers are always trying to seek exposable vulnerabilities and improve on hacking methods.

**Problem 8**

*Explain why it is important to log user actions in the development of secure systems.*

Logging records can show user intents with the software. By recording the user's action, we can identify malicious conduct and the actors behind it. Besides security purposes, logs can also help from user experience and marketing perspectives as they track user interest in specific system components.

**Problem 9**

*Explain why it is important when writing secure systems to validate all user inputs to check that these have the expected format.*

Input validation enforces the correct formatting of information submitted by users. The formats are predefined according to business logic or company preferences, making them prerequisites to a working system. Concerning computer security, an incorrect data format may be deliberately designed to deceive and manipulate the system, so input validation necessary to rule out this possibility.

**Problem 10**

*Suggest how you would go about validating a password protection system for an application that you have developed. Explain the function of any tools that you think may be useful.*

Instead of storing login credentials in plain text, I would utilize hashing techniques to obfuscate the password into a non-human readable format. Unlike encryption, in which the encrypted data can be converted back to the original form with the appropriate private key, hashing has less risk of exposure by permanently altering the data.

However, if I am developing a client-centric application, I prefer the default password manager of the operating systems I am deploying into. For example, Keychain Access in Apple devices, GNOME Keyring in GTK-based Linux, and Android has a password manager integrated within its Google Play Services.

# Bibliography

Leopold, D. (2018, Feb 2). *itemis*. Retrieved from Why security is one of the biggest engineering challenges ahead: https://blogs.itemis.com/en/why-security-is-one-of-the-biggest-engineering-challenges-ahead/

Sommerville, I. (2016). Software Engineering. In *Security engineering* (10 ed., p. 374). Pearson Education.