# Computer Security Semester Review

Guilherme Gomes Haetinger

*- For thy worry will be our gather and Nick Weaver's defeat will be our glory -*

## Contents

## 1 Security Principles

We have a number of security principles that must be considered once we are developing software. Some of them can be enumerated as the most important. These are:

- **Security is Economics**: Only spend as much money as what you are trying protect is worth. Don't buy a $10000 lock for a $10 bike.
- **Least Privilege**: Only give a program the actual amount of privilege it needs to do its purpose. We should not give ROOT access to a program that plays the nyan cat video.
- **Fail-safe Defaults**: Safe defaults in the sense of "*if something fails, what should be the current state?*". It's recommended that we use *default-deny* policies. The light goes down on a server's building. Should the electronic lock on the server access door be unlocked or stay locked? → If we want *default-deny* policies, the door will stay shut, keeping the server's hardware safe from whomever wants to access it, which will keep it secure from someone who jammed the building's circuits just to get to the computers.
- **Separation of Responsibility** Separate privilege: "Nobody has full privilege by itself". Nuke triggers need multiple people turning a key to work (at least in movies).
- **Defense in Depth** Create redundant layers of protection. In the medieval times, castles were protected by an outer wall and an inner wall, so that enemies would have to go through 2 different walls to truly invade it. This made the process much harder.
- **Psychological Acceptability**
- **Human Factors**

- **Complete Mediation**
- **Don't rely on Security through Obscurity**
- **Design Security from the start**
- **Kerkchoff's Principle**