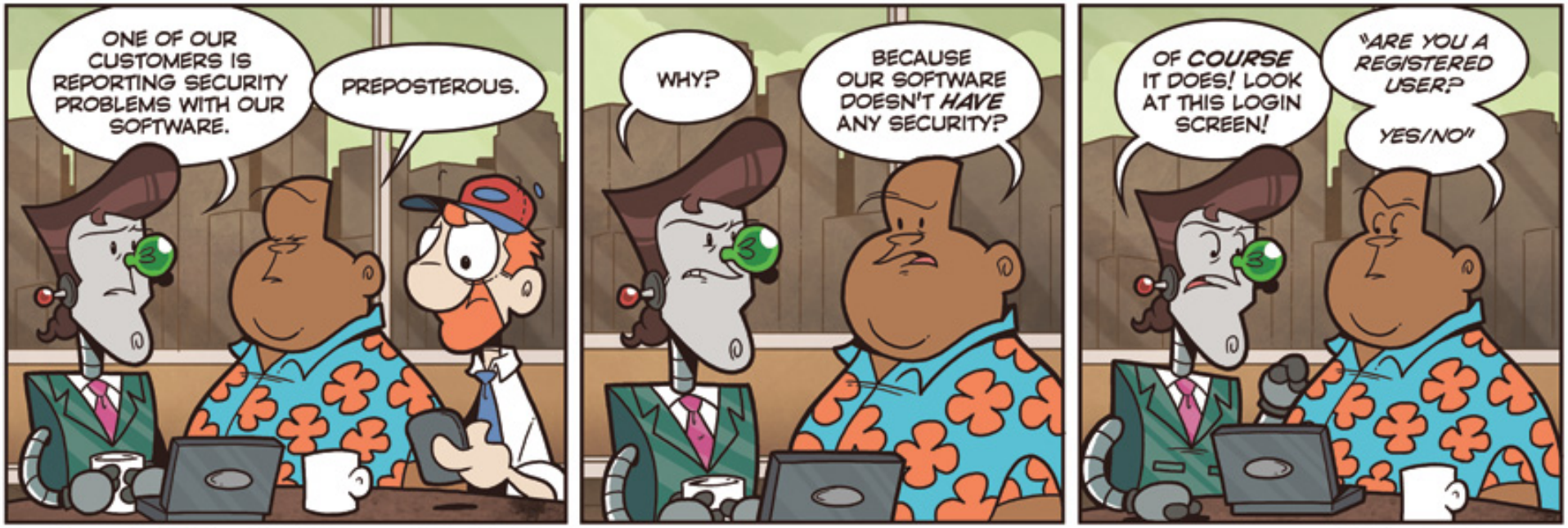
Introduction to the course

Segurança em Software

Pedro Adão, Ana Matos

TÉCNICO LISBOA



Teaching staff

- Pedro Adão - *coordinator* + *lectures* + *labs*
  - Office at Alameda – Office 3, Informática 2
  - Office at IST TagusPark – 2N3.3

- Ana Matos - *coordinator* + *lectures* + *labs*
  - Office at Alameda – TBD
  - Office at IST TagusPark – 2N3.9

Objectives

- to give the students the mental tools necessary to understand the problem of the security of the computer and its software, vis-à-vis the security of the communication or distributed system

- to give a deep insight into the security problems in modern software systems, and present paradigms, models and tools to tackle these problems

Program overview

- Principles of Computer Security
- Software Vulnerabilities
- Development of Secure Software
- Control of the Execution Environment
- Language-Based Security
- A Case Study: Java Security

Program in detail

- Principles of Computer Security
  - Basic properties and concepts; Software security design principles.
- Software Vulnerabilities
  - Web applications and databases; Conventional applications (buffer overflows, race conditions)
- Development of Secure Software
  - Software auditing; Validation and encoding.
- Control of the Execution Environment
  - Dynamic protection; Mitigations of vulnerabilities.
- Language-Based Security
  - Information flow analysis; Security type systems; Secure low-level code; Proof carrying code; Security monitors.
- A Case Study: Java Security
  - Sandboxing and stack inspection; Java security flaws; Java secure programming guidelines.

Language-Based Security

techniques based on programming language theory and implementation, including semantics, types, optimisation and verification, brought to bear on the security question

Schneider et. al, 2000

In sum:

Language-Based Security

techniques based on programming language theory and implementation, including semantics, types, optimisation and verification, brought to bear on the security question

Schneider et. al, 2000

In sum:

Language-Based Security

techniques based on programming language theory and implementation, including semantics, types, optimisation and verification, brought to bear on the security question

Schneider et. al, 2000

In sum:

- Target of attacks: Programs

Language-Based Security

techniques based on programming language theory and implementation, including semantics, types, optimisation and verification, brought to bear on the security question

Schneider et. al, 2000

In sum:

- Target of attacks: Programs
- Defence: Language-based enforcement techniques.

Security by design

Security by design

- Software applications are implemented in programming languages

Security by design

- Software applications are implemented in programming languages
- systems are modelled at different levels of abstraction (using different languages)

Security by design

- Software applications are implemented in programming languages
- systems are modelled at different levels of abstraction (using different languages)
- security policies can be expressed and analysed at each of these levels

Security by design

- Software applications are implemented in programming languages
- systems are modelled at different levels of abstraction (using different languages)
- security policies can be expressed and analysed at each of these levels
- security-by-design: using language-based analysis techniques to enforce specified security properties with strong guarantees

Secure? (w.r.t. ...)

Secure? (w.r.t. ...)

- $y_H := x_L$

Secure? (w.r.t. ...)

- $y_H := x_L$
- $x_L := y_H$

Secure? (w.r.t. ...)

- $y_H := x_L$ ✓
- $x_L := y_H$

Secure? (w.r.t. ...)

- $y_H := x_L$ ✓
- $x_L := y_H$ ✗

Secure? (w.r.t. ...)

- $y_H := x_L$ ✓
- $x_L := y_H$ ✗

- if y_H then $x_L := 0$ else $x_L := 1$

Secure? (w.r.t. ...)

- $y_H := x_L$ ✓
- $x_L := y_H$ ✗

- if y_H then $x_L := 0$ else $x_L := 1$
- while y_H do skip ; $x_L := 0$

Secure? (w.r.t. ...)

- $y_H := x_L$ ✓
- $x_L := y_H$ ✗

- if y_H then $x_L := 0$ else $x_L := 1$ ✗
- while y_H do skip ; $x_L := 0$

Secure? (w.r.t. ...)

- $y_H := x_L$ ✓
- $x_L := y_H$ ✗

- if y_H then $x_L := 0$ else $x_L := 1$ ✗
- while y_H do skip ; $x_L := 0$ ✓

Secure? (w.r.t. ...)

- $y_H := x_L$ ✓
- $x_L := y_H$ ✗

Explicit leak

- if y_H then $x_L := 0$ else $x_L := 1$ ✗
- while y_H do skip ; $x_L := 0$ ✓

Secure? (w.r.t. ...)

- $y_H := x_L$ ✓
- $x_L := y_H$ ✗

Explicit leak

- if y_H then $x_L := 0$ else $x_L := 1$ ✗
- while y_H do skip ; $x_L := 0$ ✓

Implicit leak

Secure? (w.r.t. ...)

Secure? (w.r.t. ...)

- $y_H := x_L$

Secure? (w.r.t. ...)

- $y_H := x_L$
- $x_L := y_H$

Secure? (w.r.t. ...)

- $y_H := x_L$ ✓
- $x_L := y_H$

Secure? (w.r.t. ...)

- $y_H := x_L$ ✓
- $x_L := y_H$ ✗

Secure? (w.r.t. ...)

- $y_H := x_L$ ✓
- $x_L := y_H$ ✗
- if y_H then $x_L := 0$ else $x_L := 1$

Secure? (w.r.t. ...)

- $y_H := x_L$ ✓
- $x_L := y_H$ ✗
- if y_H then $x_L := 0$ else $x_L := 1$
- while y_H do skip ; $x_L := 0$

Secure? (w.r.t. ...)

- $y_H := x_L$ ✓
- $x_L := y_H$ ✗
- if y_H then $x_L := 0$ else $x_L := 1$ ✗
- while y_H do skip ; $x_L := 0$

Secure? (w.r.t. ...)

- $y_H := x_L$ ✓
- $x_L := y_H$ ✗
- if y_H then $x_L := 0$ else $x_L := 1$ ✗
- while y_H do skip ; $x_L := 0$ ✗

Secure? (w.r.t. ...)

- $y_H := x_L$ ✓
- $x_L := y_H$ ✗
- if y_H then $x_L := 0$ else $x_L := 1$ ✗
- while y_H do skip ; $x_L := 0$ ✗

Termination leak

Labs/practical classes

- Labs (hands-on)
  - Cross site scripting
  - SQL injection
  - Buffer overflows
  - Format string vulnerabilities
  - Race conditions
- Practical classes
  - Security principles
  - Access control models and information flow
  - Formal semantics
  - Information flow analysis in imperative languages
  - Information flow analysis in low level languages
  - Information flow and compilation

Labs

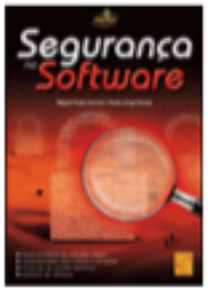
- More info soon
  - CTF-style labs
  - BoD
- Previous years:
  - SEED Labs
  - http://www.cis.syr.edu/~wedu/seed/
- You can install the virtual machine they provide in VirtualBox or Vmware
  - SEEDUbuntu16.02
  - http://www.cis.syr.edu/~wedu/seed/lab_env.html

Bibliography

- ***Segurança no Software***

  Miguel Correia and Paulo Sousa

  FCA, September 2010/2017



- Complementary:
  - ***The 24 Deadly Sins of Software Security: Programming Flaws and How to Fix Them***, Michael Howard, David LeBlanc and John Viega, 2009, McGraw-Hill ISBN 9780071626750
  - ***Building Secure Software: How to Avoid Security Problems the Right Way***, John Viega and Gary McGraw, 2002, Addison-Wesley ISBN 9780201721522
  - ***Introduction to Computer Security***, Matt Bishop, 2005, Addison-Wesley
- Alternative texts for non-Portuguese speaking students (email me)

Evaluation

- 2 Tests (25%+25%) both or each can be repeated
- Practical components:
  - Project1 (20%)
  - Project2 (30%)
- Groups of 2 or 3 students for Practical components
  - All students are expected to participate, and are responsible for, all practical components
- Min. grade: 9.5 in the average of Tests and of Practical components
- Partial grades from previous years not reused

☞ Read “Métodos de Avaliação”, Fénix

Tests

- Important Dates
  - Test 1 - 05 November, 18:00
  - Test 2 - 18 January, 11:30
  - Repetition - 05 February, 18:30
- Cover Theoretical and Lab classes
- Can be answered in Portuguese or English
- Tests from last years will be made available, but note:
  - Detailed content and highlights are adjusted every year.
  - Use slides and summaries as reference.

☞ Check “Avaliação / Evaluations”, Fénix

Practical Components

- Project1 - Build, Break and Fix
  - Build - 24Sep-08Oct
  - Break - 12Oct-25Oct
  - Fix - 26Oct-08Nov

- Project2
  - Solution and its implementation - 16Nov-06Dec
  - Report - due 13Dec
  - Project Discussions: last week of semester

☞ Check “Avaliação / Evaluations”, Fénix

Communication

- Pedro Adão
  - Contact hours: TBD
  - Email: pedro.adao@tecnico.ulisboa.pt
- Ana Matos
  - Contact hours: TBD
  - Email: ana.matos@tecnico.ulisboa.pt
- Web site: https://fenix.ist.utl.pt/ ...

Classes

ALAMEDA **TAGUS**

	Seg 9/21	Ter 9/22	Qua 9/23	Qui 9/24	Sex 9/25
07:00					
08:00		08:00 - 09:30 L LAB 7			
09:00	09:30 - 11:00 T FA1	09:30 - 11:00 L LAB 11			09:30 - 11:00 L 0 - 14
10:00					
11:00	11:00 - 12:30 L LAB 8	11:00 - 12:30 T VA5		11:00 - 12:30 T 1 - 22	11:00 - 12:30 T 1 - 22
12:00					
13:00					
14:00					
15:00					
16:00				15:30 - 17:00 L 1 - 17	

Study materials

- Book / other texts
- Papers
- Lab guides
- Slides
- Problem sets

Study materials

- Book / other texts
- Papers
- Lab guides
- Slides
- Problem sets

- Videos and online content

Cyber-Security specialization

- New in the restructured MEIC (start: Sept. 2015)
- Implements the Information Assurance and Security Knowledge Area of the ACM/IEEE Computer Science Curricula 2013
- Aims to give students the technical skills necessary to analyse, protect and manage the security of personal, corporate and governmental computer systems from cyber threats

Cyber-Security specialisation

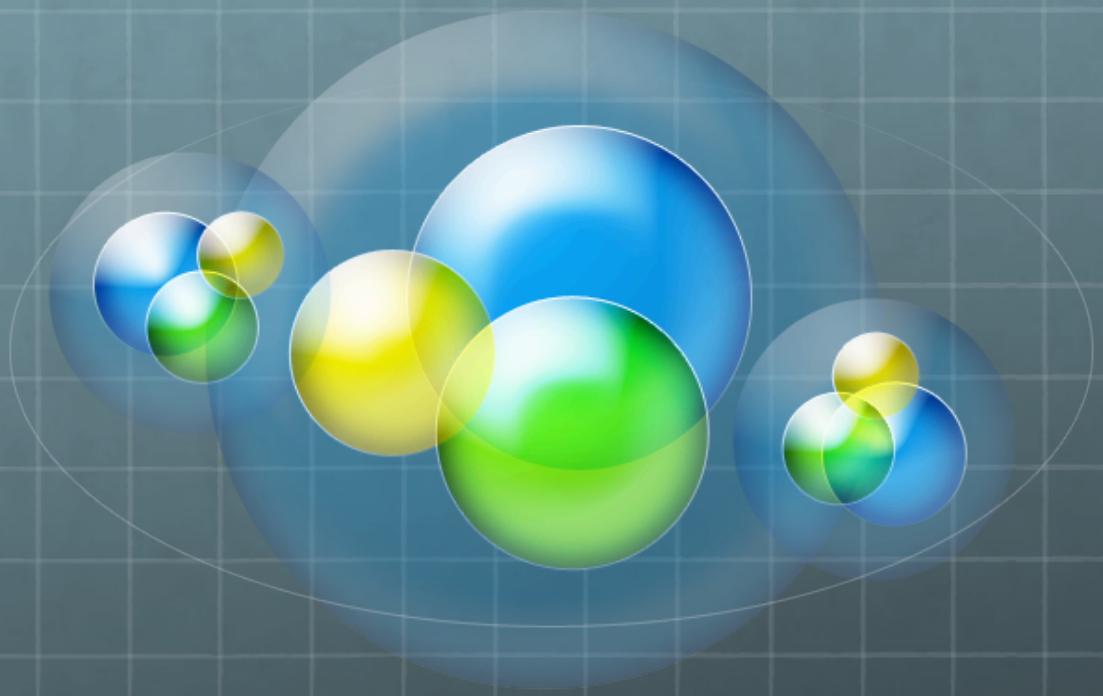
- Courses:
  - Network and Computer Security (SIRS)
  - Software Security (SSof)
  - Forensics Cyber-Security (CSF)
  - Cryptography and Security Protocols (CPS)
  - Highly Dependable Systems (SEC)

Ethics and law

- The purpose of the course is to learn how to protect computer systems from cyber-attacks
  - but some of the things you learn may also be used to attack them
- Notice that
  - **Attacking systems is unethical and punished by law**
  - Even just “testing” systems without written permission may be punished by law

Ethics and law

- The purpose of the course is to learn how to protect computer systems from cyber-attacks
  - but some of the things you learn may also be used to attack them
- Notice that
  - **Attacking systems is unethical and punished by law**
  - Even just "testing" systems without written permission may be punished by law
- *Don't try this at home* → *Try this just at home*



Who wants to hack?

Creating a (ethically-responsible) hacking team@IST

Pedro Adão

How do we want to do it?

- Invite ALL students with interest in Security to participate
- Teach Computer Security in an ethically responsible and competitive environment
- Meet regularly (twice a month) to learn new tricks

How do we want to do it?

- Invite ALL students with interest in Security to participate
- Teach Computer Security in an ethically responsible and competitive environment
- Meet regularly (twice a month) to learn new tricks
- Participate in CTF competitions

How successful have we been?

- Consistent top-50 in CTFs since 2017

How successful have we been?

- Consistent top-50 in CTFs since 2017
- Several top-20 and top-10 classifications

- International on-site participations after competitive qualifications
  - Volga CTF, Russia (Sep 2017, Sep2020)
  - CSAW European Finals (Nov 2017 (5th), Nov 2018 (3rd), Nov2019, Nov2020)
  - RuCTF Finals (Apr 2018, Apr 2019)

But with great power comes great responsibility

Ethics

Ethics

- Exploiting others' vulnerabilities is illegal
- Each of us is responsible for his own behavior

Ethics

- Exploiting others' vulnerabilities is illegal
- Each of us is responsible for his own behavior