

# Operating Systems Security

## General information about this course

Radboud University, Nijmegen, The Netherlands



Winter 2017/2018

## About this course

- ▶ Lecture (hoorcollege): Tuesday, 10:30–12:30 in HG00.303
- ▶ Exercise class (werkcollege): Thursday, 8:30–10:30 in HG00.062
- ▶ Exam on Thursday, January 25, 12:30–15:30 in LIN 4/LIN 5
- ▶ Exam grade is your final grade for this course
- ▶ 3 EC points
- ▶ Website:  
<https://cryptojedi.org/peter/teaching/os-security-2017.shtml>
- ▶ Language of the lectures: English

# Teachers

## **Peter Schwabe**

Office: Mercator I, 3.18

`peter@cryptojedi.org`

## **Veelasha Moonsamy**

Office: Mercator I, 3.20A

`v.moonsamy@cs.ru.nl`

## **Carlo Meijer**

Office: Mercator I, 3.11

`C.Meijer@cs.ru.nl`

## **Wouter Kuhnen**

`w.j.a.kuhnen@student.ru.nl`

## **Thom Wiggers**

`t.wiggers@ru.nl`

# Homework

- ▶ Homework assignments will be online (at the latest) Thursday morning
- ▶ Homework assignments are due Thursday (one week later) by midnight (sharp!)
- ▶ Homework submission through Blackboard
- ▶ Homework submission in groups of 2 (preferably)
- ▶ Grading of homework in **g**, **v**, **o**, and **NSI**
- ▶ Grading has no effect on final grade, but:

**More than one NSI and you're not admitted to the exam!**

# Homework environment

- ▶ Programming courses need a computer (with compiler etc.)
- ▶ Network security course needs a network that you can break

# Homework environment

- ▶ Programming courses need a computer (with compiler etc.)
- ▶ Network security course needs a network that you can break
- ▶ Operating systems security course needs an operating system

# Homework environment

- ▶ Programming courses need a computer (with compiler etc.)
- ▶ Network security course needs a network that you can break
- ▶ Operating systems security course needs an operating system
- ▶ Highly recommended: set up Linux in a virtual machine
- ▶ Practical Exercises will mainly use Linux

## Examples of what you will learn

- ▶ How authentication and authorization works (and fails)



## Examples of what you will learn

- ▶ How authentication and authorization works (and fails)
- ▶ How processes are separated

## Examples of what you will learn

- ▶ How authentication and authorization works (and fails)
- ▶ How processes are separated
- ▶ How the OS helps to make memory attacks harder

## Examples of what you will learn

- ▶ How authentication and authorization works (and fails)
- ▶ How processes are separated
- ▶ How the OS helps to make memory attacks harder
- ▶ Why traditional UNIX security is insufficient today
- ▶ Malware and how it hides from malware scanners

## Examples of what you will learn

- ▶ How authentication and authorization works (and fails)
- ▶ How processes are separated
- ▶ How the OS helps to make memory attacks harder
- ▶ Why traditional UNIX security is insufficient today
- ▶ Malware and how it hides from malware scanners
- ▶ How operating-systems can be “hardened”:
  - ▶ Enforcing mandatory access control
  - ▶ Compartmentalization and virtualization
  - ▶ Examples: Subgraph OS, Qubes, Android

# Disclaimer

- ▶ Some things taught in this course are illegal when you do it “in the wild”
- ▶ You’re grown up, use your skills responsibly
- ▶ If you want to try something out, get consent

# Disclaimer

- ▶ Some things taught in this course are illegal when you do it “in the wild”
- ▶ You’re grown up, use your skills responsibly
- ▶ If you want to try something out, get consent
- ▶ In the homework, don’t break anything that others still need
- ▶ Be careful when attacking your own machine:
  - ▶ Make sure that you attack the *virtual* machine
  - ▶ Make sure that the attack only affects the virtual machine