

# Security Lab

## Kick-off



# Alex Scherphof

Security specialist / Ethical hacker @ SecureIT  
Security Lecturer / Coordinator @ RUAS

BSc in Computer Science Engineering  
MSc in Cyber Security

Hiking, urbexing, reading, gaming



# Agenda

Programme introduction  
Rules of engagement  
Requirements  
Weekly schedule  
Projects  
Examination and grading  
Community platform  
Hâck The Hague

Teams and introductions  
Kick-off project 0: Code Review





# Introduction





# Introduction

- A programme designed around the practical application of cyber security knowledge and skills.
- Fundamental security knowledge through lectures.
- Application of knowledge through assignments and projects.
- Self-sustainability; the ability to find solutions instead of just applying them.

In short: obtaining the hacker mindset.



# Lectures

- Threat modeling
- Application security
- Reporting
- Cryptography
- Human factors
- Project management
- Phishing attacks
- Network security
- Computer security
- Malware on mobile platforms
- OpenSource Intelligence(OSINT)
- Pentesting and auditing
- Some supporting lectures



# Projects

- Project 0: Code Review
- Project 1: Security Awareness
- Project 2: Malware analysis
- Project 3: Security Audit



# Rules of engagement

- IEEE Code of Ethics
- NCSC Coordinated Vulnerability Disclosure Guidelines
- Always keep me in the loop / don't surprise me.





# Requirements

- Laptop/PC capable of running common cyber security software.
- Laptop/PC capable of running Linux Kali in a virtual machine.
- Laptop/PC with camera/webcam and microphone.
- Sign a non-disclosure agreement(NDA) with the Rotterdam University of Applied Sciences



# Weekly schedule

	Monday	Tuesday	Wednesday	Thursday	Friday
<b>Morning 1</b> 09:00 – 10:30	News recap / discussion	Lectures		Unsupervised project time	Unsupervised assignment / project time
<b>Coffee break</b>					
<b>Morning 2</b> 10:40 – 12:10	Lectures	Lectures		Unsupervised project time	Unsupervised assignment / project time
<b>Lunch</b>					
<b>Afternoon 1</b> 13:00 – 14:30	Lectures / assignment / project time	Lectures / assignment / project time	Progress review	Unsupervised project time	Unsupervised assignment / project time
<b>Coffee break</b>					
<b>Afternoon 2</b> 14:40 – 17:00	Assignment / project time	Assignment / project time	Progress review / assignments / project time	Unsupervised project time	Unsupervised assignment / project time





# Projects



# Project 0 – Code Review

The analysis of a free and open-source software(FOSS) web-application. By reviewing the code as well as performing simple vulnerability scanning methods each team is required to assess the security of their chosen application.

**15% of final grade**

## **Learning objectives**

After completing this project, the student:

- Can recognize common vulnerabilities in source code.
- Has demonstrated the ability with various manual and automated methods for vulnerability scanning, and source code analysis and testing.
- Can document the methods in a professional, transparent and reproducible manner.
- Can propose detailed, actionable mitigations for found vulnerabilities in source code.
- Is familiar with common secure coding principles and security-by-design.





# Project 1 – Security Awareness

Performing a security awareness audit on the students and/or personnel of the Rotterdam University of Applied Sciences. Each team will be assigned a target audience on which a security awareness test has to be performed through a targeted phishing campaign.

**15% of final grade**

## **Learning objectives**

After completing this project, the student:

- Is familiar with the psychological aspects of cyber security.
- Is capable of applying these psychological aspects to design a targeted phishing attack.
- Can design and develop an awareness campaign that is in compliance with data protection regulations.
- Can execute a targeted phishing attack during an awareness campaign.
- Can analyze and report on findings within a professional setting.



# Project 2 – Malware analysis

Individually each student is required to analyze a potential malicious android application. Each team is required to establish a testing protocol that individual team members must follow. The testing protocol as well as the findings of each app are aggregated in a team report.

**30% of final grade**

## **Learning objectives**

After completing this project, the student:

- Is capable of setting up an appropriate and safe testing environment for android malware analysis.
- Can use publicly available tools to perform a preliminary analysis on a malicious application.
- Understands and can explain the permissions requested by android applications.
- Can analyze the behavior, network usage, code and process of a malicious android application.
- Is capable of recommending preventive, detective and reactive measures as well as write YARA rules to automatically detect similar malware in the future.
- Can professionally report on the findings.





# Project 3 – Security Audit

Performing a professional security audit on the production environment of the Rotterdam University of Applied Sciences. This includes all legal requirements, scoping meetings, reconnaissance, testing, reviewing, reporting and presenting.

**40% of final grade**

## **Learning objectives**

After completing this project, the student:

- Has demonstrated the ability to do due diligence and minimize liability for him/her and the client.
- Can perform a penetration test on a given (set of) live system(s) to the specifications of a client.
- Is familiar with the different stages in the unified cyber kill chain.
- Has demonstrated the ability to disclose found vulnerabilities in a responsible and professional manner.
- Can document the methods in a transparent and reproducible manner.
- Can propose concrete, actionable mitigations and recommendations for found vulnerabilities.





## Examination and grading





# Examination and grading

Component	Grading system	Requirements
<b>Individual/Team Assignments</b>	Pass / Fail	All assignments need to be completed successfully to receive final grade.
<b>Project 0 Code review</b>	1 – 10	15% of final grade.
<b>Project 1 Security awareness</b>	1 – 10	15% of final grade.
<b>Project 2 Malware analysis</b>	1 – 10	30% of final grade.
<b>Project 3 Security audit</b>	1 – 10	40% of final grade.





# Examination eligibility

For a student to be eligible for examination the student needs to have successfully completed all assignments, have passed all projects with a minimum grade of 5.5 and have a maximum absence of 20% of supervised hours. If the student hasn't completed the assignments or has a failing grade for a maximum of one minor- and one major-project between 3.1-5.5 the student will have an opportunity for re-examination after repairs.

The 20% maximum absence rule applies to the examination of individual projects. The 20% is calculated over the weeks of each project respectively.



## Re-examination eligibility

If the student does not successfully complete all assignments the lecturers will provide the student with either another opportunity to complete the assignments or have the student write a report on the assignment subject that the student failed. It is up to the lecturers to choose which of the options is provided to the student.

If a team does not meet the required passing grade of  $>5.5$  for a project they get a chance to repair the project. The team will be informed by the lecturers on what part of the project needs to be repaired. Repairs have to be made within two weeks after the grades have been returned to the teams. In the event that one of the projects is graded with a 3.0 or lower it is considered not repairable within a realistic time-frame and the team will not be eligible for re-examination.

In the situation that the individual modifier causes a student to get a failing grade ( $<5.5$ ) the students will be given an individual repair assignment that can consist of individual repairing team deliverables or a separate replacement assignment.



# Individual grading component

Each project grade will carry a team component and an individual modifier component. Depending on the contribution of the individual student, the student can receive up to two points or be deducted up to two points. The distribution for this individual modifier is as follows:

<b>+2</b>	The student has been vital in the successful completion of the project. The project result is achieved for a large part due to this student's contribution.
<b>+1</b>	The student has done more work than the group average during the project. The project result is made measurably better due to the student's contribution.
<b>+0</b>	The student has done an equal or similar amount of work as the rest of the group during the project.
<b>-1</b>	The student has done less work than the group average during the project. The project result was impacted due to this student's lack of contribution.
<b>-2</b>	The student has consistently done less work than the group average, has been made aware of this fact during the project, and has not made enough effort to improve.





# Community platform

## Discord



# Community platform

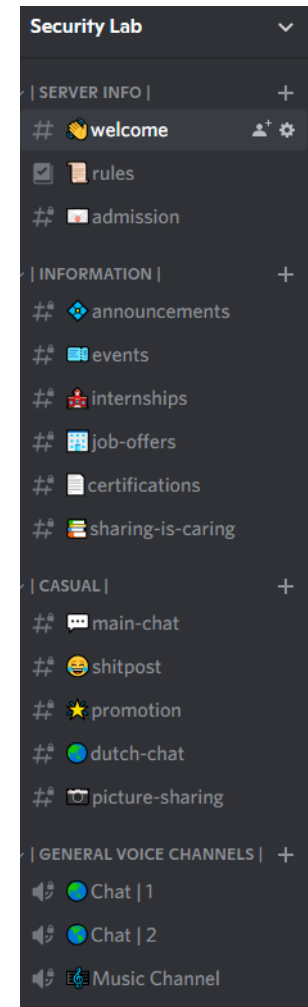
## Discord

To replace the social aspect usually achieved through in-person lectures the programme has a community platform through Discord.

To join, visit [discord.security-lab.nl](https://discord.security-lab.nl) in your browser and follow instructions.

Please make sure to do the following when joining:

- Read the rules carefully
- Respect the boundaries between work hours / non-work hours





# Hâck The Hague





# Hâck The Hague

Monday 27<sup>th</sup> of September

5 seats remaining

1-4 person teams

€500 - €2000 prizes

Hâckademic Award



# Hâck The Hague

Monday 27<sup>th</sup> of September

5 seats remaining

1-4 person teams

€500 - €2000 prizes

Hâckademic Award



# Coffee break

15 minutes





# Teams and introductions





# Team 1

- Hieu Dinh
- Çağlar Gül
- Daan Nekeman
- Mats Pondman
- Max van Vliet



## Team 2

- Arno Rietdijk
- Bilal Azrioual
- Caslay Oort
- Christian van Os
- Emily van de Burgt



## Team 3

- Ahmet Karataş
- Gilles Coolen
- Haydar Pehlivan
- Steven Soekha
- Yasemin Snoek





## Team 4

- Hamza Fethi
- Hicham El Marzgioui
- Nabil Chane
- Niels van de Kerkhof
- Tim van der Perk
- Usman Siddiqui



## Team 5

- Jayden Weverink
- Jordy van der List
- Michael Francis
- Rafael Bieze
- Tim Wolcken



## Team 6

- Carolina Leano Giraldo
- Daan de Heer
- Léa Commaret
- Lucas Oudhuis
- Richard van der Knaap



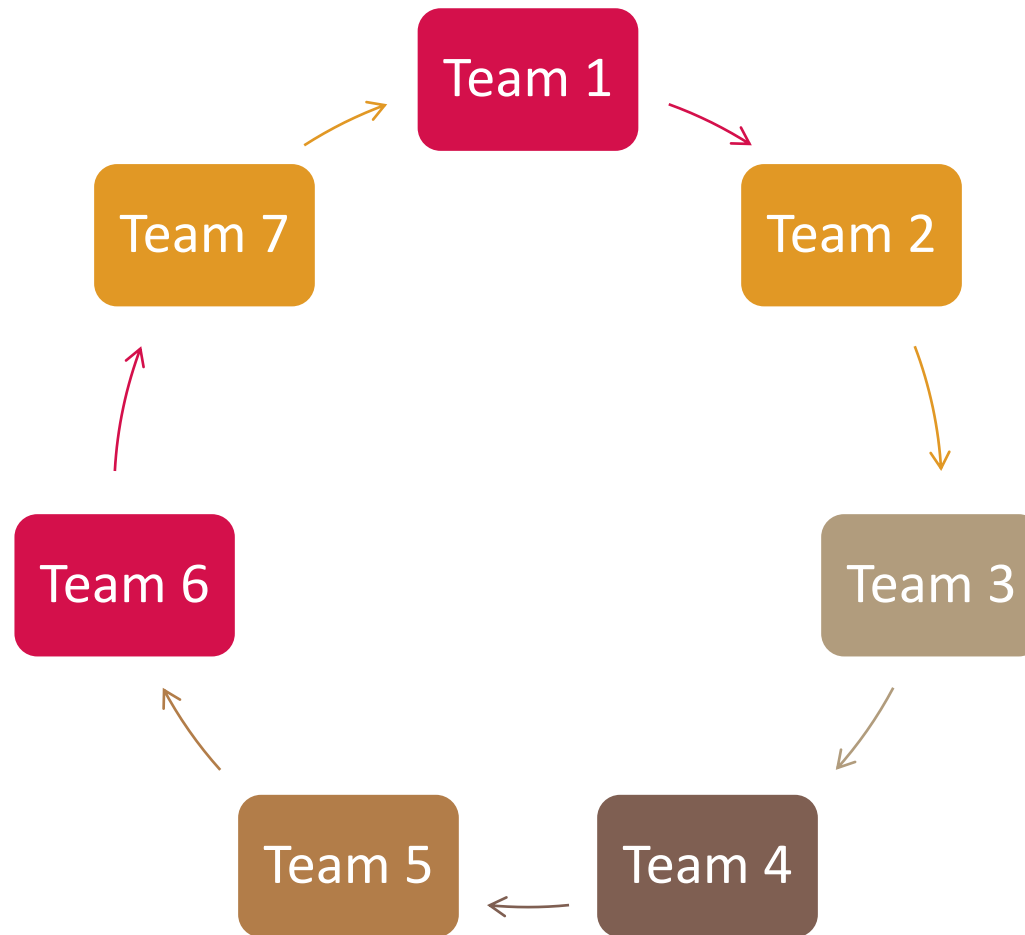


## Team 7

- Hendrikus Crebolder
- Joseph Libier
- Niek van Leeuwen
- Pascal Holthuijsen
- Pelle Nieuwenhuizen



# Cross-team peer evaluation



# Project 0: Code Review

## Kick off





# Introduction Code Review

The analysis of a free and open-source software(FOSS) web-application. By reviewing the code as well as performing simple vulnerability scanning methods each team is required to assess the security of their chosen application.

**15% of final grade**

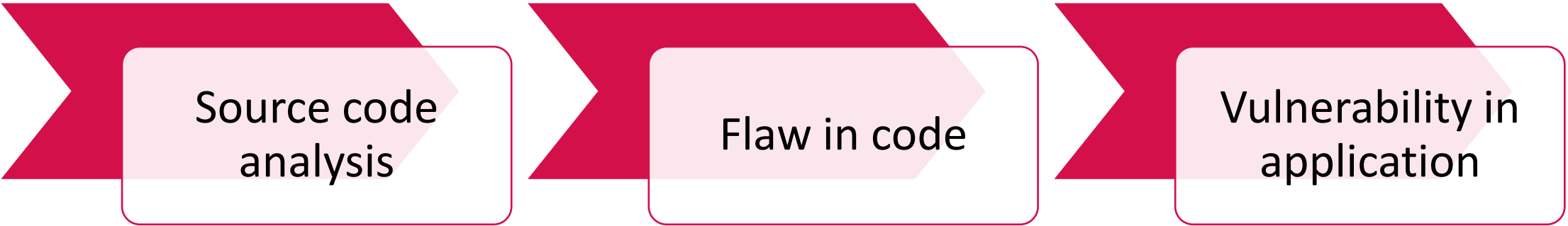
## Learning objectives

After completing this project, the student:

- Can recognize common vulnerabilities in source code.
- Has demonstrated the ability with various manual and automated methods for vulnerability scanning, and source code analysis and testing.
- Can document the methods in a professional, transparent and reproducible manner.
- Can propose detailed, actionable mitigations for found vulnerabilities in source code.
- Is familiar with common secure coding principles and security-by-design.



# Code Review



Source code  
analysis

Flaw in code

Vulnerability in  
application



# Code Review



Recommend fix

Flaw in code

Vulnerability in  
application





# Assignment

- Get to know your fellow team members
- Assign roles
  - Team lead
  - Engineer – Development expert
  - Analyst – Security expert
  - Communication – Professional communication/reporting expert
- Search for a FOSS application as a subject for Project 0.

**Class resumes at 13:00**





**ROTTERDAM UNIVERSITY  
OF APPLIED SCIENCES**

**exceed** expectations