Introduction

- Aivo Toots
  - Security engineer at Cybernetica AS
  - MSc, Cybersecurity
  - Penetration testing, software development experience

- Andres Jõgi
  - Junior security engineer at Cybernetica AS
  - Cyber Security MSc student
  - Penetration testing, system administration experience
Why

- Shortage of professionals with Cybersecurity skills
- Theory vs Practice:
  - Secure Programming Techniques - Theory
  - Secure Programming Techniques Project - Practice
- Employers prefer practical experience!

For some reason this still happens [2]
Goals

- You gain:
  - Practical experience
  - Improved writing and communication skills
  - Potential addition to your portfolio

- Community gains:
  - Discovered & fixed security problem(s) for real world software

You will break software during this course. But you have to help to fix it later. [1]
How

1. You are here
2. Pick interesting project
3. Search for security issue(s)
4. Document issue(s)
5. Fix issue(s)
6. Pull request
7. Success
Grading

- **40%** for midterm presentation & report
  - 10% For finding a bug
  - 10% For documenting it
  - 20% For report & presentation quality

- **60%** at the end of the semester
  - 10% For fixing the bug
  - 10% For sending a patch upstream
  - 10% Submitting final report in time
  - 30% Final report & presentation quality

We will be a bit more forgiving[3]
Timeline

- 11.02.2022 - First meeting, intro [seminar]
- 25.02.2022 - Code auditing demo with scanners [seminar]
- 31.03.2022 - Security bug found and reported [deadline]
- 08.04.2022 - Midterm meeting & bug documented [seminar, deadline]
- 29.04.2022 - Final report deadline [deadline]
- 13.05.2022 - Final presentation [seminar]
- 20.05.2022 - Final presentation (if needed) [seminar]
Project ideas & examples

- Your favourite open source project
- Your current work project (given that you can show us the code)
- If you really can’t find anything, ask us for help

Users will find a way to break your software in one way [4] or another ... [5]
How not to approach this project

- Do not attack live systems!
  - Unless you have reached an agreement with the owner beforehand

- Do not rely on automatic scanners alone
  - You must critically assess output of automated tools
  - Validate that real issue exists in source code

- You are learning to be security professionals, act like it
  - Reporting false positives is a waste of time for everybody involved
How to approach this project

● Read about common vulnerabilities:
  ○ OWASP top 10
  ○ Mitre CVE list
  ○ OWASP ASVS checklist

● Look through interesting software projects seeking for common problems/Issues
  ○ Less active older projects probably have more issues
  ○ Look for vulnerable dependencies

● If you have any doubts or questions about the validity of the chosen project or discovered bug, contact us
Tools

● Source code analysis tools (SAST)
  ○ Laravel enlightn
  ○ Graudit
  ○ And more

● Penetration testing tools (DAST)
  ○ Burp Suite
  ○ OWASP ZAP

● Good old manual labour
  ○ Ctrl+f, cat+grep and intuition
Practical aspects

● Held online

● Communications
  ○ Lectures in Zoom
  ○ Questions during lectures and via e-mail & Zulip

● Start looking for the project now :)  

● Next time we will do a demo bug hunt
Questions?
Thank you!

Andres Jõgi
Aivo Toots
References

- [https://i.redd.it/vnk6h88p80071.png](https://i.redd.it/vnk6h88p80071.png) (Accessed on 8. Feb 2022)