Secure Programming Techniques Project

MTAT.07.016

Andres Jõgi
Aivo Toots
Regarding ongoing conflict

- Think before running web-pages / tools to (cyber)attack
- Laws - it is forbidden in Estonia - no difference between “good” and “bad” attacks
- Tools suggested in forums may contain malware
- Geolocation - requests from certain countries may be forwarded (to attack some other systems instead)
- You do not really know what you are attacking and what might get affected in addition
- For practice, CTF portals: [https://portal.ctftech.io/](https://portal.ctftech.io/), [https://www.hackthebox.com/](https://www.hackthebox.com/)
Quick reminder - 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 [1] or another ... [2]
Today - Methodology

● Common approaches:
  ○ Dynamic Application Security Testing (DAST)
    ■ Black-box testing
  ○ Static Application Security Testing (SAST)
    ■ White-box testing
  ○ Manual code review

● Modern:
  ○ Interactive Application Security Testing (IAST)
    ■ Combination of DAST & SAST
Dynamic Application Security Testing

- Non functional manual and automated testing
- No access to the source code
- Vulnerabilities are detected by actually performing attacks
- **Strength:** covers security weaknesses and vulnerabilities present in already running application
- **Weakness:** predefined list of attacks, not tailored for specific application
- In software development: more expensive to fix found issues (found later)
- Tools: [Burp Suite](https://portswigger.net/burp) (in demo), [OWASP ZAP](https://www.owasp.org/index.php/ZAP), etc.
Static Application Security Testing

- Reviewing the source code of the software to identify sources of vulnerabilities
- Automatic scanning, interaction not needed
- **Strength:** covers 100% of the code
- **Weakness:** many false positives; does not consider vulnerabilities in complex sequences of actions
- In software development: cheaper to fix found issues (found earlier)
- Tools: [Graudit](http://graudit.com) (in demo), [SonarQube](http://sonarqube.org) (in demo), etc.
Interactive Application Security Testing

- Modern, not yet very widely used approach
- Combination of DAST and SAST
- Runtime analysis + code scanning in the background
- Running the DAST part, tries to identify what exactly is running when a problem is found
- Rather improved DAST than replacement for both
- Tools: Acunetix, VeraCode, etc.
Manual code review

- Good old manual labour
  - Ctrl+f, cat+grep and intuition

- Reviewing source code and trying to detect issues
- Easier when software development and/or security testing experience
- Effectiveness depends a lot on the expertise of the tester
- Tools: Visual Studio Code, Notepad++, Vim, etc.
Summary

- DAST, SAST, IAST, manual code review - all useful
- Best to try out each
- Choose which suits you best (to achieve the goal)
- In this course - identify and report vulnerabilities in source code:
  - Start with DAST and verify in code
  - Start with SAST and verify (exclude false positives)
  - Manual code review
  - Combination of all
- Relying solely on automatic scans is not what is expected from highly educated specialists :)
Timeline

- 11.02.2022 - First meeting, intro [seminar]
- 04.03.2022 - 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]
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.
Questions?
Demo
Thank you!

Andres Jõgi
Aivo Toots
References