

## Secure Software Development – Seminar 4: Programming Languages / Fog and Future Trends

### Outline:

- Announcements
- Fog, Edge and Osmotic Computing
- Code Reuse
- Seminar Question & Debate
- Further informations

### Announcements:

- Easter Weekend: 15 – 18 April 2022
- Peer Reviews
- Demos: Select times: (you only need a 30 min slot)
- Thursday between 7 – 8 pm
- Friday between 6 – 7 pm
- Monday between 12 – 2 or between 5 – 7 pm

### Fog, Edge and Osmotic Computing:

- Fog: Bonomi et al (2012)
- IEEE 1934-2018
- Edge: CDNs, Akami, now evolving (5G) Dilley et al, 2022)
- Osmotic Computing: Villari et al (2016)
- Challenges: Languages & Architectures
- Heterogeneous vs. Homogenous processing
  - ➔ Central aspect caching close to destination
  - ➔ Challenges with programming languages which are compatible with different devices
  - ➔ New threats arising e.g. man-in-the-middle-attack, vulnerability surface is increased
  - ➔ Challenges with capability with the layers

### Code Reuse:

- Mäkitalo et al (2020) – “On Opportunistic Software Reuse”
- ➔ SW reuse has been available since late sixties
- ➔ O’Reilly (2005) Web 2.0 and ‘Mash-Ups’
  1. What are risks with SW reuse?
    - ➔ Advanced persistent threat (APT)
    - ➔ Taking code from libraries enables the possibility to copy vulnerabilities
  2. Pros & Cons
    - ➔ Pro: Not need to “reinvent the wheel”

➔ Profit of the expertise of others

3. Possible topic for reflection?

## Seminar 6 – Reading & Questions

➔ Mistake: python is interpreted (wanted to say C)

Rust:

- Is compiled ➔ it is not easy by default to grasp information
- Taint tracking
- Quicker than Swift
- Have a memory management like Swift
- Complex to learn
- Designed to write on low level

Reflection:

- Aiming for reflective review (see last seminar sessions)
- Recommendation: 1 or two aspects
- Question: word count 800 + 10% (Yes)

Next Week:

- Submit final report
- Peer review
- Review Reflection grid
- Remember the checklist – make sure you tick each requirement