Seminar session 2: Secure System Architecture – Modelling Tools / Security and Separation

Outline:

- Announcements
- The importance of Separation
- Methods of Separation
- Containers and alternatives
- Security and AD Trees

Announcements:

- Forums and Posts
- Peer Review forms
- → Recommend to give people without a reason a 3 (average) grade.

The Importance of...:

- Separation:
- → Historical: Air Gap, Vlans
- → Latterly: Virtualisation (e.g. Qubes)
- → Issues: Shared code/libraries/resources (e.g. KVM)
- → Separation Kernel (Rushby, 1981) – e.g. PikeOS
- → Often recommended by Formal Methods e.g. seL4
- → Are containers good for separation?

Micro Services Architecture:

- "The term 'Microsevice Architecture' has sprung up over the last few years to describe a particular way of designing software applications as suites of independently deployable services." (Fowler, 2014)
- Pros:
  - ➔ Developer friendly – more control over language, tools and libraries
  - ➔ Business focused – Micro Service should be a single business function
  - ➔ API focus – helps with encapsulation and isolation
  - ➔ Use of containers – helps with isolation, portability
- Cons:
  - ➔ Security stance – what is IN the containers
  - ➔ Version control – Could lead to break the system/API
  - ➔ Need Integration Engineer?

- ➔ Libraries are a major improvement for developers, but also come with huge threats. It is not known which version of it is used, if it is up to date or even not corrupted.
- ➔ Outsource the security aspects to a third party
- ➔ Version control can be quite difficult and by wrong handling could lead to breaking the API.

- Some argue, that Microsvervice Architecture are improvements for developer teams and enables easier development. Another way to see it, is that risks and cyber security aspects (as part of the development process) are outsourced. The development is therefore not easier, part of the work is just transferred to others.

Containers and Alternatives:

- Processes provide limited segregation
- Initially Chroot jails added security
- Solaris containers/ zones
- Docker/LXC/Hyper-V, rkt – use CGROUPS and namespaces
- FOR ASSIGNMENT:
  - ➔ Need some separation – containers ideal BUT don't work in Codio
  - ➔ Launch Python in separate processes
  - ➔ Use **venv** (bit like chroot)

Review: Threats and Vulnerabilites:

- Vulnerabilities: "Weaknesses in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source" (CSRC, 2021)
- Threat: " Any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, or individuals through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service. Also, the potential for a threat-source to successfully exploit a particular information system vulnerability" (CSRC, 2021).

Threat Modelling:

- Various models:
  - ➔ SRIDE & DREAD
  - ➔ CVSS

- ➔ OCTAVE
- ➔ PASTA
- ➔ Attack Trees

Attack Trees and Graphs:

- Attack Trees: Salter et al. In 1998 coined the term attack trees for the first time
- Often attributed to Schneier (1999)
- “In the attack tree, the attacker’s goal is specified and depiceted as root of the tree. The goal is then disjunctively or conjunctively refined into sub-goals. The refinement is repeatedly looped until the sub-goals represent basic actions. Basic actions resemble atomic components for simple understanding and quantification. Conjunctive refinements indicate different steps an attacker takes to achieve a goal. While Disjunctive refinement indicates different alternatives of how a goal can be achieved” (Qin, X. & Lee, W., 2004).

AD Trees:

- Attack Defence Trees – tools that can be used for vulnerability & threat modelling
- “The Attack – Defence Tree Tool (ADTool) allows users to model and display attack-defence scenarios and allows the analysis +++

AD Tree Demo:

- To evaluate in a quantitative way: Take a look at CVSS

Next Session Unit 4:

- Unit 3: Submit assessment – AD-Trees
- Next time: review distributed system challenges; Read Hart (2015) and Bonnet et al. (2016). Create a SysML version of your proposed system.