# Common (high level) activities implemented

## Activities related to topology

* Reachability to an assets (containment): considering that we target the angle of topology we expect that the modelling of a system (i.e. CPS) contains locations that an activity initiator would traverse through to reach a certain location containing the target asset.
  + For example, a visitor entering the *hallway* (location2) from *entrance area* (location1), then enters a *reception office* (location3). Then we can define an activity which has the containers of the initiator to be *entrance area* (location1) and *reception office* (location3).
  + Current criteria for when this is applied:
    - Activity initiator in the first activity is contained by different entities in the precondition and postcondition.
    - Container of initiator in the second activity should be different from the container in postcondition of the first activity.
    - Connection between the containers of the postconditions in the first and second activity should exist.
  + This type of activity is not associated with a specific vulnerability or attack.
* Connectivity to an asset: similar to the containment we consider connectivity to be a part of the topology as we present both containment and connectivity of a topology.
  + For example, a visitor connects to smart light in toilet, then connects to a hvac in the server room. Then we can create an activity that has the initiator connected to the hvac (target asset) through a smart light (exploited asset).
  + Current criteria for when this is applied:
    - Check basics (same action name, same initiator, same activity type).
    - Activity initiator in the first activity contains an entity that has connections that increases from pre to post. These connections should be associated with the target asset.
    - In the second activity, the same entity in the postcondition has new connection different from that in the post of the first activity postcondition.
  + This type of activity is not associated with a specific vulnerability or attack. On the contrary of other attack patterns stated in other resources such as CAPEC.

## Others (mainly inspired from CAPEC)

* Collect data (main source: <https://capec.mitre.org/data/definitions/150.html>) [not implemented]: collect data from a target asset in the system. Target asset can be characterised to be a *[common] resource* of data that can be exploited to undermine a system’s security.
  + For example, collecting data from a bus network.
  + Characteristics:
    - initiator: offender
    - target asset: common asset that can contain information about the system.
    - If the target asset is a computing device, then:
      * This computing device should contain information about the system.
      * initiator should have a connection to this computing device.
  + Several actions (moving, connecting) that lead to a target asset to collect data can be abstracted to a collect data activity.
  + Collect data can be considered a standard-activity (it is classified as standard attack pattern in CAPEC) i.e. it describes an abstract technique that an attacker could use to exploit a vulnerability or reduce a system’s security. In CAPEC, data collection comes under the meta pattern *Excavation*.
* Data interception (from CAPEC <https://capec.mitre.org/data/definitions/117.html>): intercepting data transmitted from a target asset over a data stream (usually network traffic). A difference between this activity and the collect data activity is that the attacker is not the intended recipient of data. This activity is defined as a meta pattern in CAPEC.
* Footprinting (source CAPEC <https://capec.mitre.org/data/definitions/169.html>): collect information about a target asset (e.g., configuration, topology, mechanisms (e.g., security) used). Methods include social engineering, protocol manipulation (<https://capec.mitre.org/data/definitions/272.html>).
  + Specific footprinting: process footprinting, service footprinting.
* Fingerprinting (source CAPEC <https://capec.mitre.org/data/definitions/224.html>): collect data about a target asset and compare it to known indicators to determine aspects of an asset such as running operating system and firm version.
* Data injection (source CAPEC <https://capec.mitre.org/data/definitions/152.html>): injecting data (normally malicious) to control or disrupt the operation of a target asset. Injection includes code injection, protocol injection, command injection, and fault injection.