Outline of planned DetGen extension

What we can currently do

We have 26 scenarios for different protocols and malicious activity. From these, we can generate "atomic" traffic traces (and now also system logs).

Characteristics:

- Ground truth due to "atomic" nature of traces
- Modularity since senarios are independent
- Scalability due to stand-alone nature of container scenarios
- Variability due to randomisation, network emulation, subscenarios

What we are not doing so far is embed these traces in a larger network context.

Motivation

- Several testbeds (give examples) exist to allow the simulation of network-like settings in a scalable manner. However, these settings are always static, i.e. any generated data comes from the same network topology with the same services etc.

  Furthermore, there are no attempts at fusing network traffic and other types of data into one dataset

- Larger network traffic datasets do no contain any ground truth labels about the stuff going on (apart from malicious/benign labels)

- Attack simulation tools and testbeds exist, but to my knowledge only for specific individual attacks, not for multi-stage attack.

- Due to the self-sufficiency of the docker containers, our existing framework provides an easy way to create communicating networks in a randomised manner.

Goals

- Embed all scenarios in a Mininet framework in order to allow the fast inclusion of switches, routers, etc. (called Containernet)
- Design a launch script that
  - creates a network topology that is randomised to some degree
  - populates the topology with docker containers and corresponding IP addresses
  - launches scenarios for the containers in randomised intervals according to a suitable distribution

- Include the collection of application logs and system call logs for each container.
    - add a tag mechanism to match logs with the corresponding traffic captures and the same ground truth labels.

We have already figured out how to embed the scenarios in Mininet and how to collect the system logs. The other goals here so far should be relatively easy to fulfil.

- Another goal inspired by Gudmund is to add mechanism (he called it domain specific language) to combine the implemented attack scenarios into multi-step attacks. The specific combination of attacks should be variable to add variability and different angles/dimensions to the data of the multi-stage attack.
    - Creating one or more attack profiles (such as the kill-chain, but more refined) that follow define the purpose of the attack and what actions (reconnaissance, pivoting, backdoors, etc.) are involved .
    - For each action, collect a number of potential attacks that are suitable tools to achieve the action (such as SQL injections to steal user credentials, brute-forcing to gain access on a machine)
    - Create an attack launch script for each attack profile that
    - defines how many actions of each type are needed for the given topology to get to the destination
    - for each action, selects an appropriate implemented attack (in a randomised manner)
    - selects the hosts that are targeted in the attack (in a randomised manner) Furthermore, for each action there might be unsuccessful attemps at a number of hosts before success, which should be included
    - Conducts each action in the sequence with some time interval separating the actions