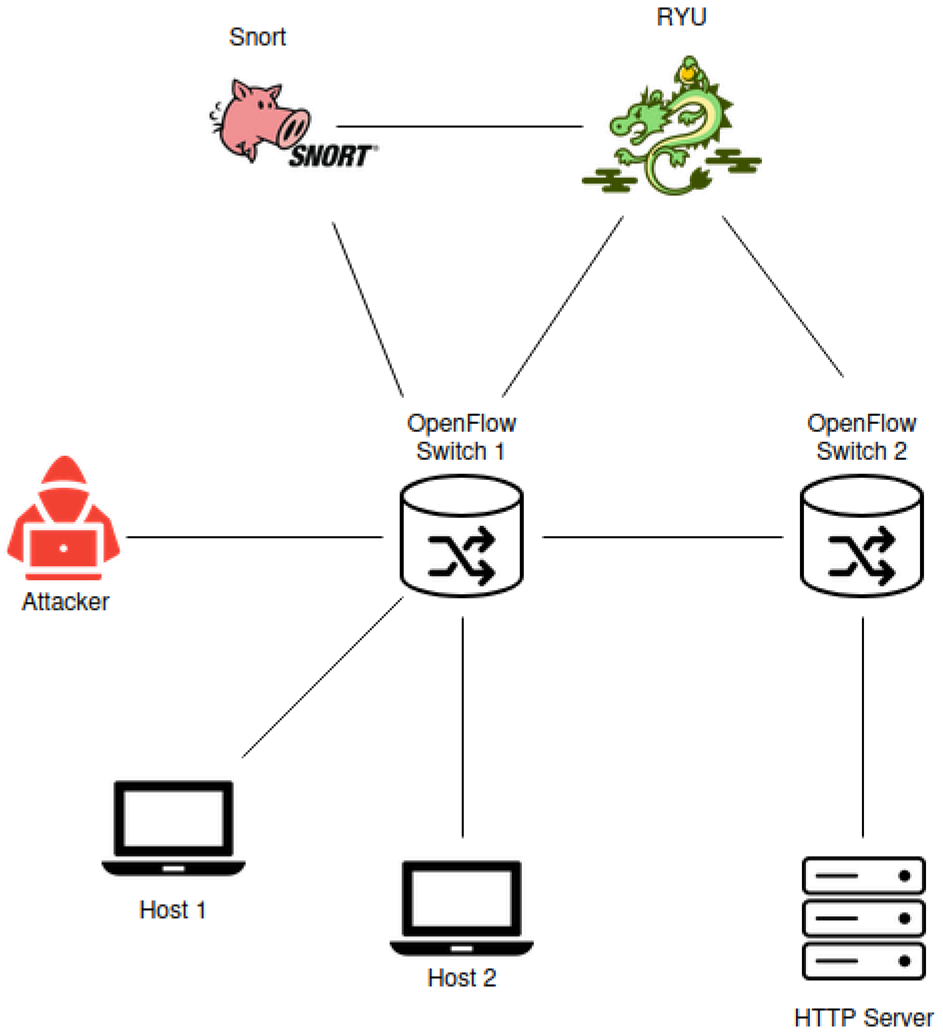
Project: Attacks on Topology

The members of the project are:
Viyaleta Palto, Valdemar Bång, Lucie Correia, Ahmed Ibraheem.

Network Topology:



- 1 SDN controller (RYU) – Monitors flows, makes routing decisions, installs flow rules.
- 2 OpenFlow switches – Forward packets based on controller rules.
- 1 Web server (HTTP) – Target for DDoS/port scanning.
- 1 Snort IDS – Detects attack signatures and alerts controllers.
- 2 Hosts (client machines) – Act as legitimate traffic sources.

- 1 Attacker machine – Initiates various attack vectors.

Attacks to be performed :

1. DDoS (Ping Flood): Use hping3 to flood the server or switch with ICMP packets. Monitor controller and switch load.
2. Port Scanning: Use nmap to scan web server ports; detect unusual scan behavior.
3. (Try to do ARP Spoofing (Use arpspoof or a Python script to poison ARP cache and reroute traffic.)
4. (Try to do a Table-Miss Striking Attack. Flood the switch with unmatched packets to exhaust flow table entries.)

How you plan to detect the attacks

- Wireshark: Use for validating traffic patterns and timestamps during the attack.
- Snort: Describe which rules or signatures will be used/customized
- CLI tools: Monitor switch port statistics and flow tables
- RYU: Program logic to recognize anomalies

What actions you plan to take when you detect the attack so as to mitigate their impact etc.

- ICMP Flood (DDoS):
  - Drop packets (too many ICMP packets too fast, block IPs which send a lot of packets for port scanning, etc.)
  - Implement rate-limiting rules in the SDN controller.
- Port Scan:
  - Detects scan patterns using Snort.
  - Blacklist source IPs for a time duration using dynamic flow rules.
- ARP Spoofing:
  - Add static ARP entries on critical nodes.
  - Drop spoofed ARP packets using controller logic.
- Table-Miss Striking:
  - Set timeouts for flow entries.
  - Rate-limit or drop unknown packets when too many table misses are detected.