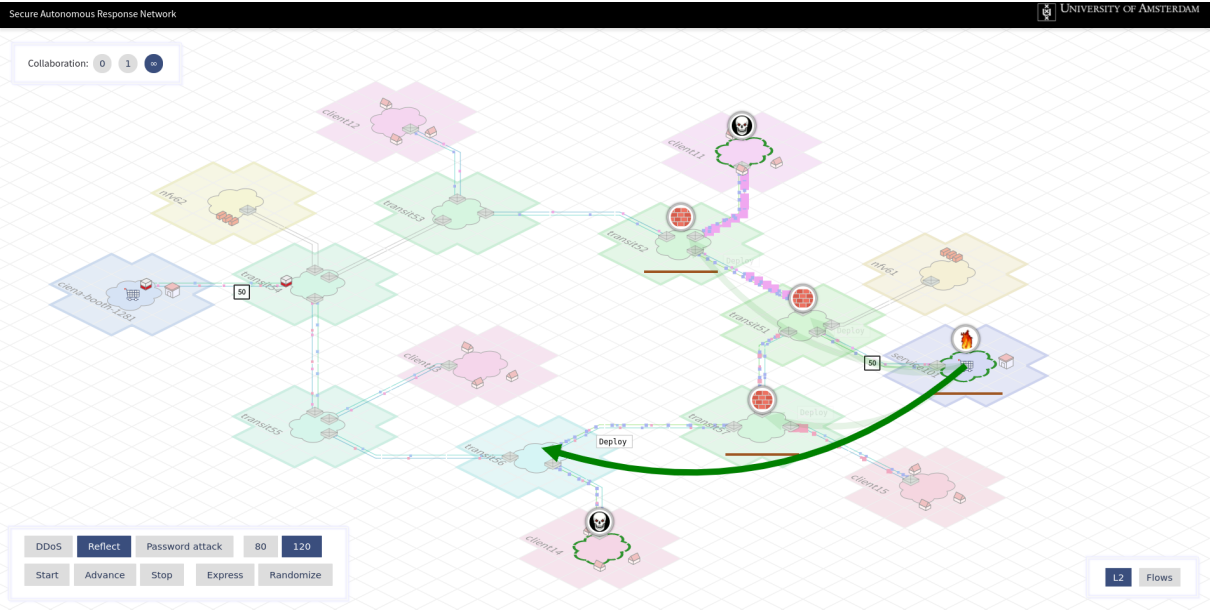
SARNET: Operational update

Ralph Koning

SNE Research Group
University of Amsterdam

SuperComputing 2017 demo



SuperComputing 2017

service101.as101.sarnet-sc17-dev

Observables

SARNET log

11:17:05 | service101: Attack password bruteforce resolved

11:17:05 | service101: Observable traffic_rate_rx@eth2 became healthy

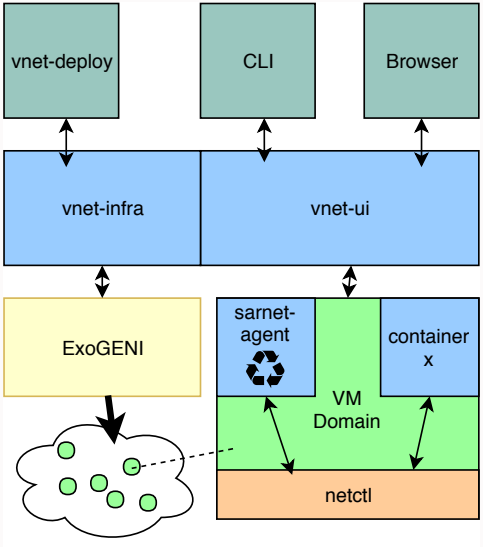
11:17:05 | service101: Observable sales_rate@service64 became healthy

11:17:05 | service101: Observable logfail_rate@service64 became healthy

11:17:05 | service101: Attack DDoS resolved

Technical details

- **SARNET Agent** defends attacks autonomously.
- **Multi** domain.
- Technologies: Alpine, **mqtt**, **ddos-tools**, quagga, BGP, docker.
- Attacks: DDoS, **Reflect**, Password.
- Defences: rate, filter, **nfv**.
- VM types: **domain**.
- **Containers**: client, service, honeypot, reflector.
- iPad + extra screen.



Results

- Increasing level of collaboration increase *effectiveness*.
- Collaboration does not necessarily increase *efficiency*.
- Successful response to reflection attack by catching *attackers*.
- The domain agent is tested on physical domains (SURF, Ciena).

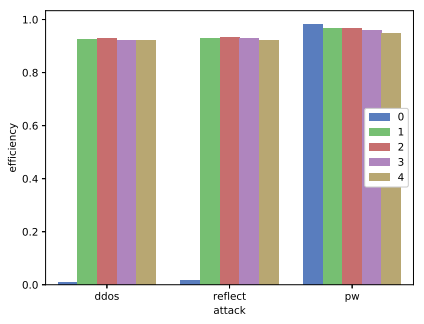
Current work

- UI client that can iterate through and repeat scenario.
- Support more collaboration levels.
- Implemented efficiency calculation in ui client.
- Per domain costs (fixed, periodic).
- Per domain behaviour (delay, success rate).

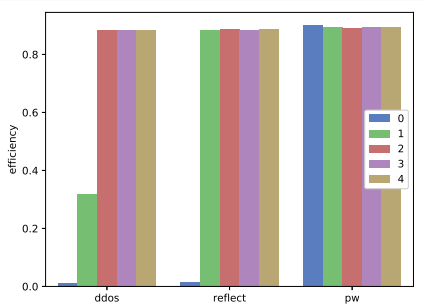
```
$ ./vnet-cli -v results
: prepared 2 scenarios, timeout 60. max runtime 900s
: stopping all running attacks
: cleanup: start
: cleanup: done
: learning: start
: learning: ended
: start: pw(service101,['client15', 'client12'],local)
: cleanup: start
: cleanup: done
: runner: start attacks (level=local)
: client: broadcasting domain behaviour parameters
: attack detected at: 3
: runner: victim reports attacked
: timeout at: 63, 60
: runner: timeout occurred
: attack duration: 63s
: runner: waiting a bit
: runner: stopping attacks
: runner: end
```

Figure 1: Sample output of cli based ui running a scenario

Preliminary results



Efficiency for each collaboration level **light** attack.



Efficiency for each collaboration level **heavy** attack.

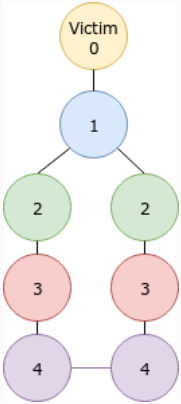


Figure 2: Collaboration levels

Future work

- ! Moving efficiency calculation to domain agents.
- ! Managing initialization of local state for domain agents.
- Multi domain experiments.
- Implementation of per domain social trust (benevolence, integrity, competence).
- ? Switching to more scalable observable framework?
- See if we can use OPTOSS AI for AI based detection.

Output

Papers:

- FGCS submission Measuring the Efficiency of SDN Mitigations Against Attacks on Computer Infrastructures. (minor revision)

Student supervision:

- E. Kooistra: Hardening virtual environments against cache based side channel attacks.
- B. Jansen: A comparative security evaluation of default configurations.
- T. Carpay: Using AI to detect successful web application exploits over https.