Automated security using SARNET

Ralph Koning

SNE Research Group

UNIVERSITY OF AMSTERDAM

Introduction

Problem:

- Amount of attacks increase in quantity, size, and complexity.
- Security departments need to deal with these threats.
- Security departments want to deal with important or new threats.

Introduction

Problem:

- Amount of attacks increase in quantity, size, and complexity.
- Security departments need to deal with these threats.
- Security departments want to deal with important or new threats.

Solution:

Research question

How do we create a network capable of automated response to attacks?

- How do we research such a network without harming others?
- How do we evaluate defenses?
- How do we measure defense performance?
- Can collaboration help in defending distributed attacks?

SARNET control loop

Detection phase:

Detect, Classify, Analyze

Decision phase:

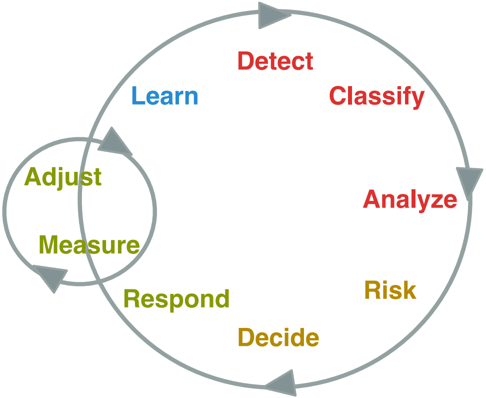
Risk, Decide

Respond phase:

Respond, Measure, Adjust

Learn phase:

Learn (used as input for decide)



Platform and Technologies

Platform

ExoGENI, Openstack

Technologies

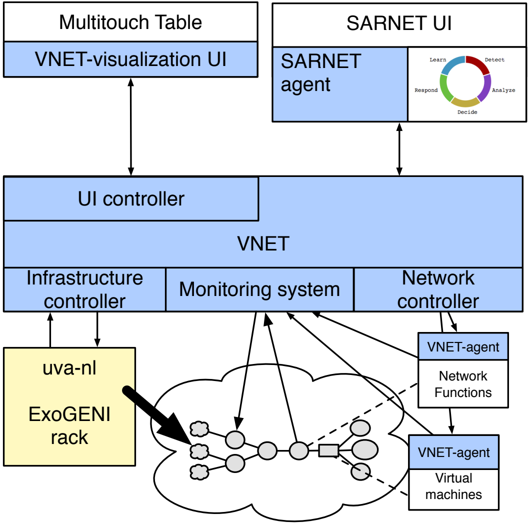
Alpine, mqtt, Quagga(BGP), Docker.

Container types

client, service, honeypot, reflector.

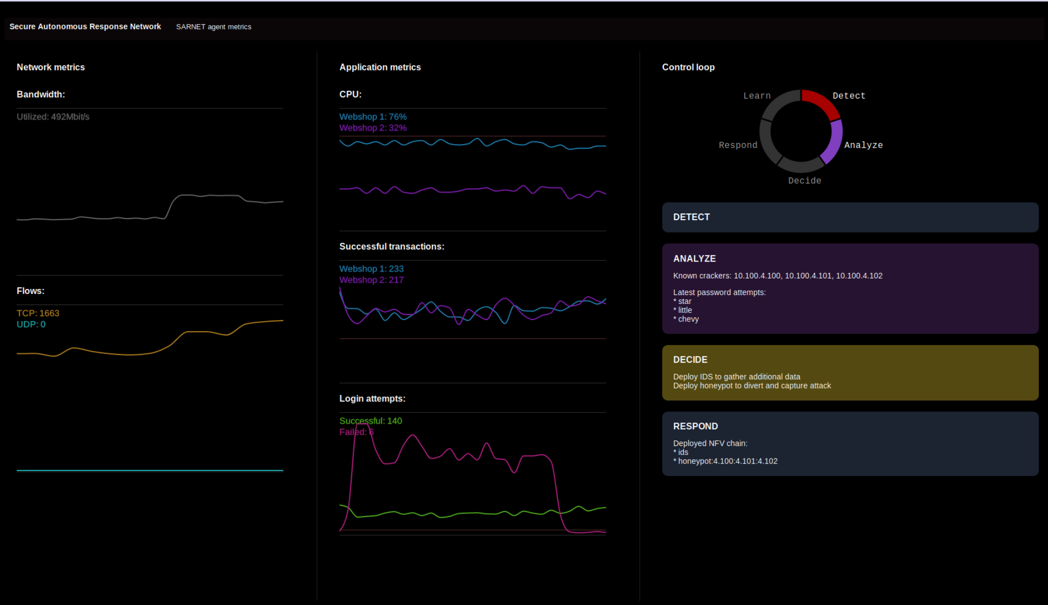
VM types

host, router, switch, nfv/cluster, **domain.**



Metrics, Observables

University of Amsterdam



Secure Autonomous Response Network SARNET agent metrics

Network metrics

Bandwidth:

Utilized: 492Mbit/s

Flows:

TCP: 1663
UDP: 0

Application metrics

CPU:

Webshop 1: 76%
Webshop 2: 32%

Successful transactions:

Webshop 1: 233
Webshop 2: 217

Login attempts:

Successful: 140
Failed: 6

Control loop

DETECT

ANALYZE

Known crackers: 10.100.4.100, 10.100.4.101, 10.100.4.102

Latest password attempts:
* star
* little
* chevy

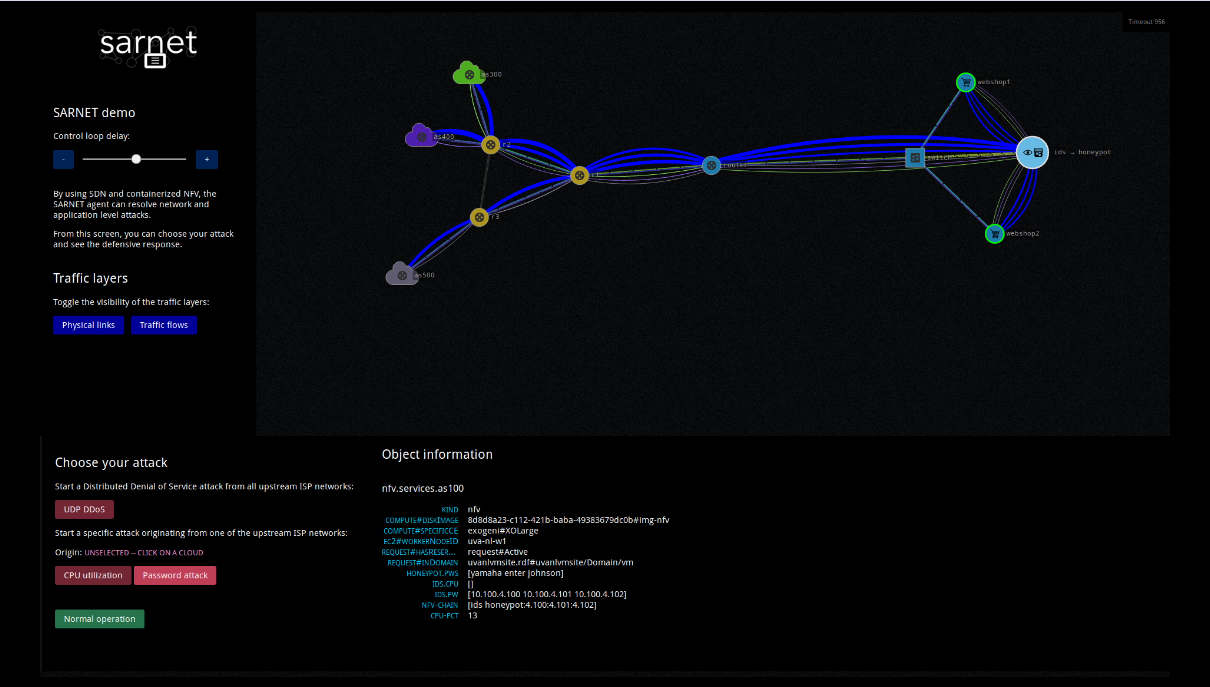
DECIDE

Deploy IDS to gather additional data
Deploy honeypot to divert and capture attack

RESPOND

Deployed NFV chain:
* ids
* honeypot:4.100:4.101:4.102

SARNET 2017



SARNET demo

Control loop delay:

By using SDN and containerized NFV, the SARNET agent can resolve network and application level attacks.

From this screen, you can choose your attack and see the defensive response.

Traffic layers

Toggle the visibility of the traffic layers:

Choose your attack

Start a Distributed Denial of Service attack from all upstream ISP networks:

Start a specific attack originating from one of the upstream ISP networks:

Origin: UNSELECTED -- CLICK ON A CLOUD

Object information

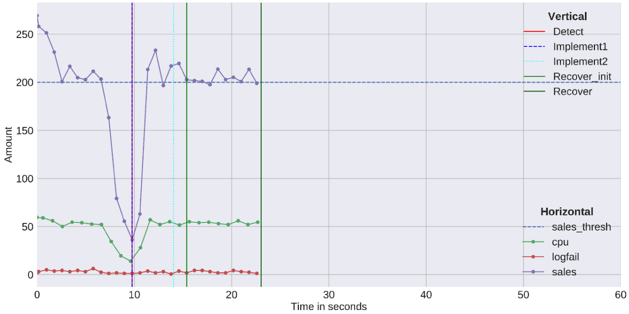
nfv.services.as100

KIND	nfv
COMPUTE#DISKIMAGE	8d8d8a23-c112-421b-baba-49383679dc0b#img-nfv
COMPUTE#SPECIFICCE	exogeni#XOLarge
EC2#WORKERNODEID	uva-nl-w1
REQUEST#HASRESER...	request#Active
REQUEST#INDOMAIN	uvanlvmsite.rdf#uvanlvmsite/Domain/vm
HONEYPOT.PWS	[yamaha enter johnson]
IDS.CPU	[]
IDS.PW	[10.100.4.100 10.100.4.101 10.100.4.102]
NFV-CHAIN	[ids honeypot:4.100:4.101:4.102]
CPU-PCT	13

Response selection

How do we pick the best response to an attack in the **decide phase**?

- Risk evaluation
- **Response selection**



Efficiency based response selection[1][2]

We can use metric *efficiency* to learn the best defense.

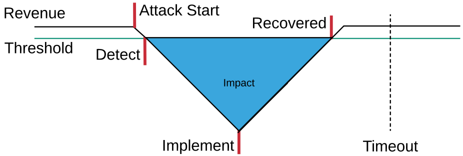


Figure 1: Efficiency requires the impact of an attack; impact is the blue area under the graph

$$\text{E(isRecovered?, I, Ct)} \overset{?}{=} \begin{cases} \beta + \alpha \frac{B \cdot T - I}{B \cdot T} + (1 - \beta - \alpha) \frac{C \cdot T - Ct}{C \cdot T} & \text{Recovered,} \\ \alpha(\frac{\beta}{1-\beta}) \frac{B \cdot T - I}{B \cdot T} + (1 - \beta - \alpha)(\frac{\beta}{1-\beta}) \frac{C \cdot T - Ct}{C \cdot T} & \text{otherwise,} \end{cases}$$

Figure 2: Equation for efficiency

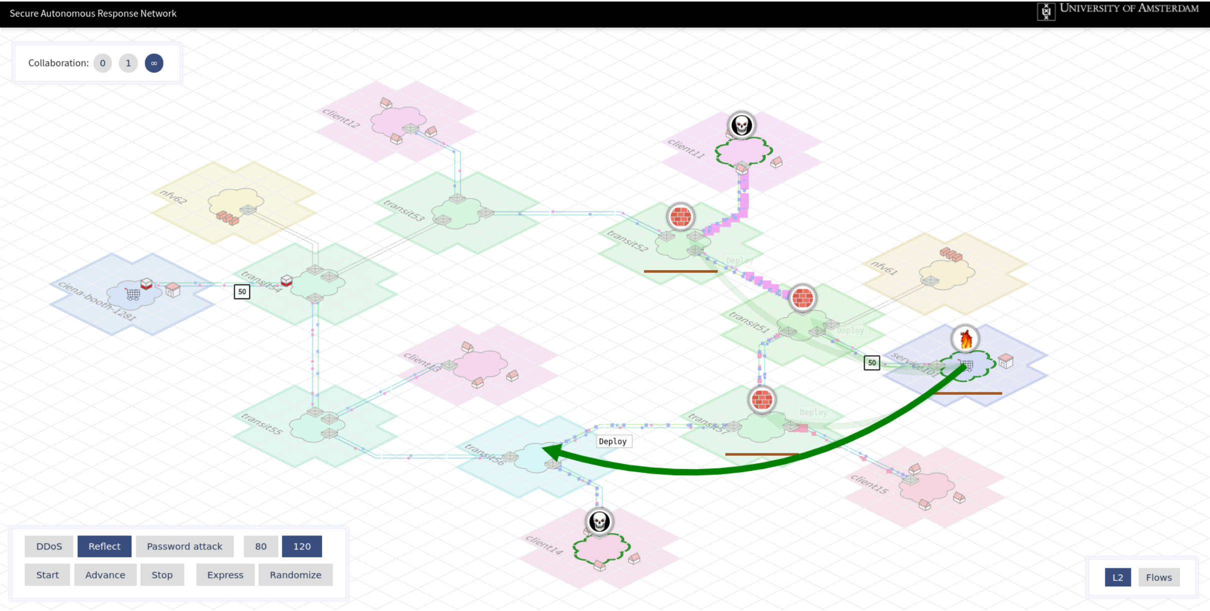
Attack	First choice	Second Choice
cpu_attack	captcha	honeypot
pwd_bf_attack	honeypot/captcha	-
ddos_attack	udp-filter	-
ddos_attack(light)	udp-filter	udp-rateup

Table 1: Defence options per attack ranked by efficiency

[1] **koning2017netsoft.**

[2] **koning2018fgcs.**

Multi-Domain SARNET

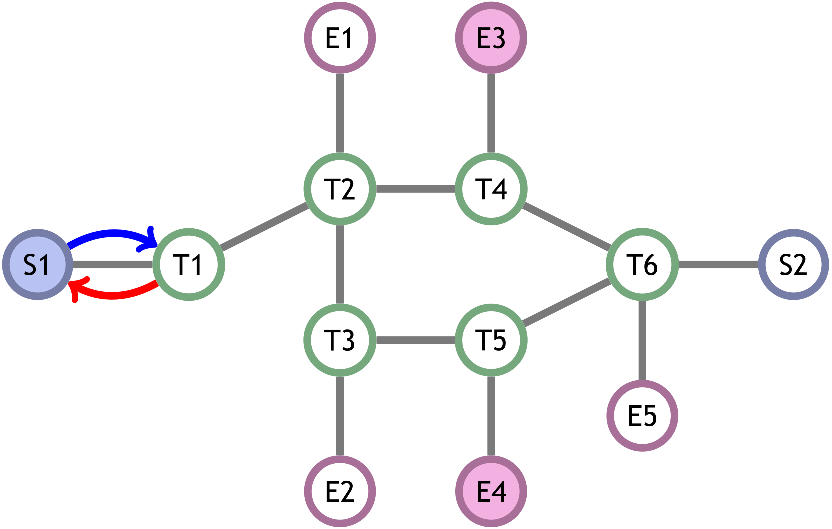


Multi-domain defense: block immediately

Time: 1

Cost: **0**

Impact: **10**

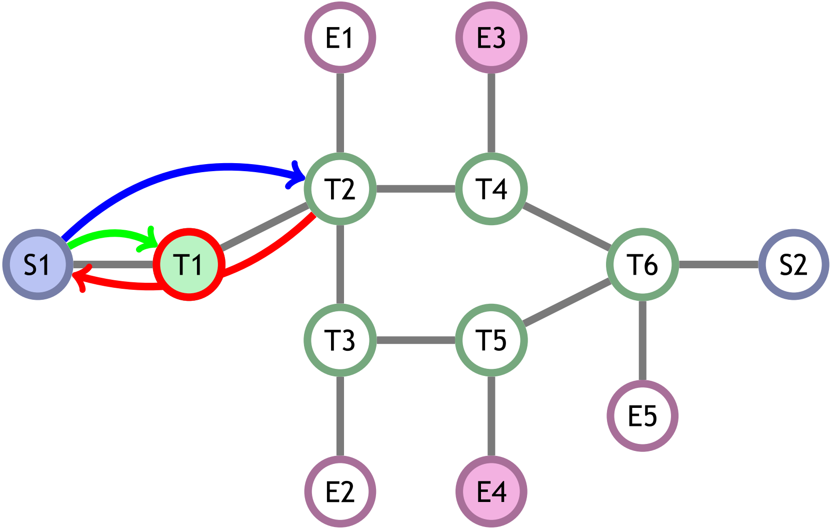


Multi-domain defense: block immediately

Time: 2

Cost: **10**

Impact: **10**

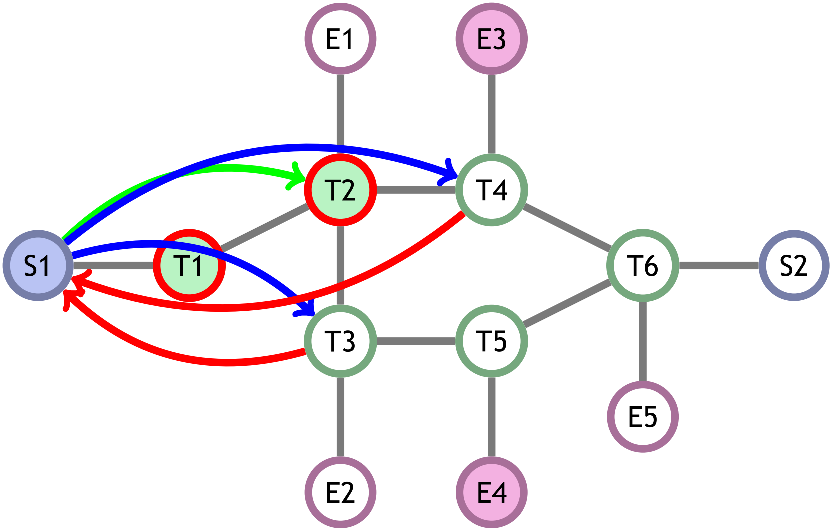


Multi-domain defense: block immediately

Time: 3

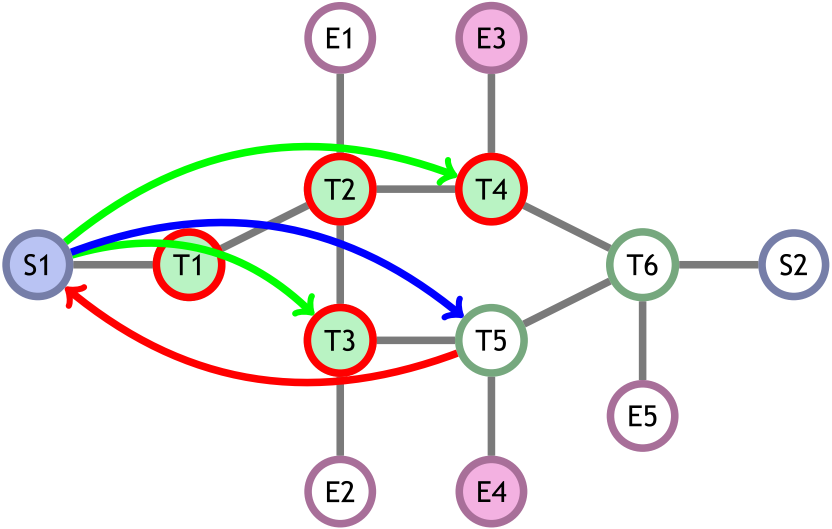
Cost: **20**

Impact: **10**



Multi-domain defense: block immediately

Time:	4
Cost:	**40**
Impact:	**10**

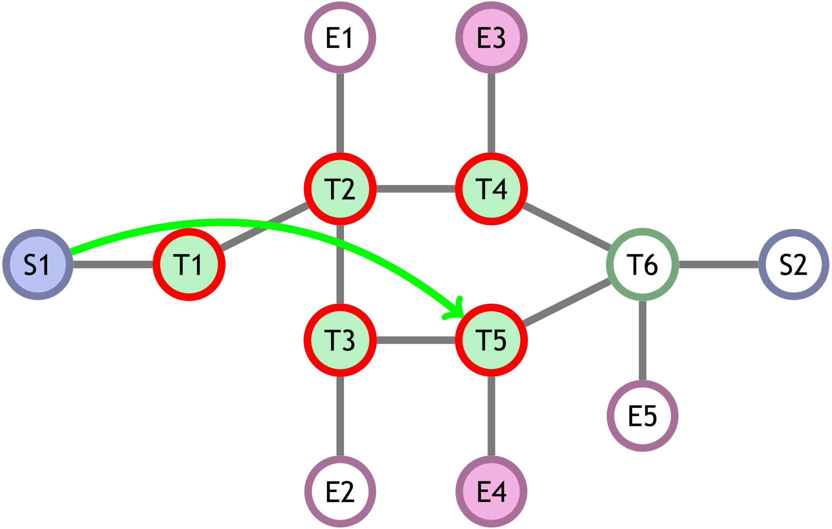


Multi-domain defense: block immediately

Time: 5

Cost: **50**

Impact: **10**



Defense approaches

Invoking a multi domain defense can be done in multiple ways.
How do these approaches perform in terms of efficiency?

We look at three of them:

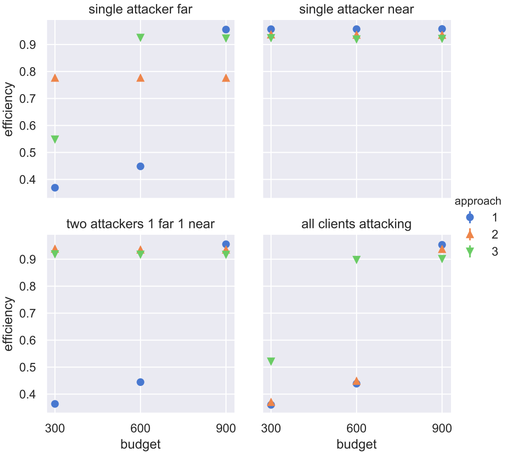
- Approach 1: Block everywhere (starting at victim).
- Approach 2: Minimise amount of countermeasures.
  (or defend close to attacker).
- Approach 3: Minimise defense propagation.

The effect of budget on approach efficiency

- Approach 1 is not so efficient; it always consumes the complete budget.
- For single attacker far situations Approach 2 scores higher than 3.

As a general purpose approach we reccommend Approach 3.
However, Approach 3 is not very alliance ’friendly’ as it only removes traffic from the target.

Figure 3: approach performance for different budget sizes



From metrics to tasks

Defences can be comprehensive, tasks are basic and take few parameters.

Each task can be fulfilled by any (capable) member in the alliance.

Metric	Observable	Classification	Defence	Task
bandwith	>80%	DDoS	Wait it out	start scrubbing
tcp/udp ratio	>0.9		Filter locally	redirect clean
transactions	<0.8		Filter remotely	redirect dirty
			remote scrubbing	

Computational Trust based algorithm

A computational Trust Model allows us to:

- Identify and isolate untrustworthy members
- Estimate the interaction risk
- **Deciding whether and with whom to interact**

Trustworiness' Factors[3]

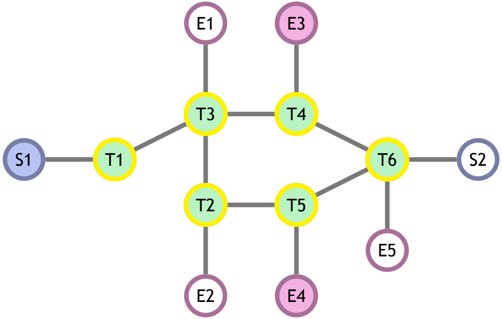
- Competence: The potential ability of the member.
- Integrity: Whether the member fulfills commitments (assumed for now).
- Benevolence: Whether the member acts good and out of kindness.

[3] **deljoo2018sctm.**

Remote help selection based on social trust

Benevolence based algorithm.

Assume integrity of alliance members (for now)

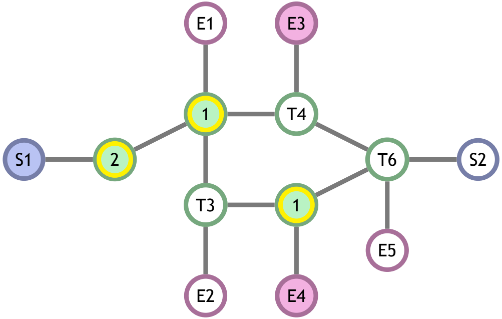


Remote help selection based on social trust

Benevolence based algorithm.

Assume integrity of alliance members (for now)

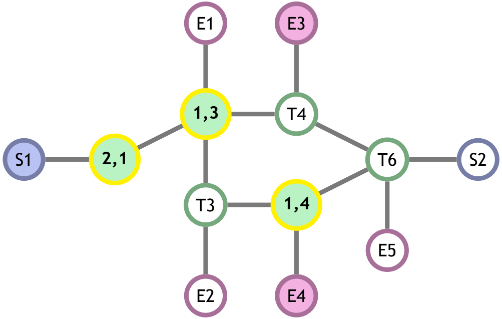
Rank nodes on competence to perform task ‘t‘



Remote help selection based on social trust

Benevolence based algorithm.

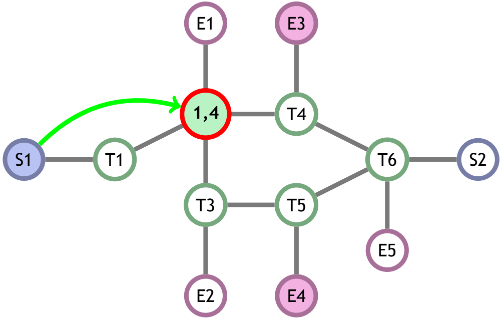
- Assume integrity of alliance members (for now)
- Rank nodes on competence to perform task ‘t‘
- Resolve ties using on benevolence



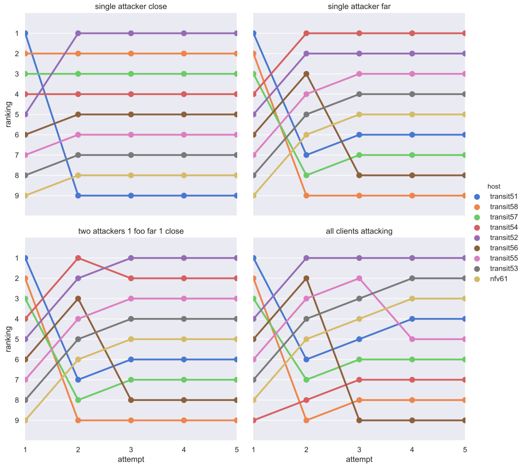
Remote help selection based on social trust

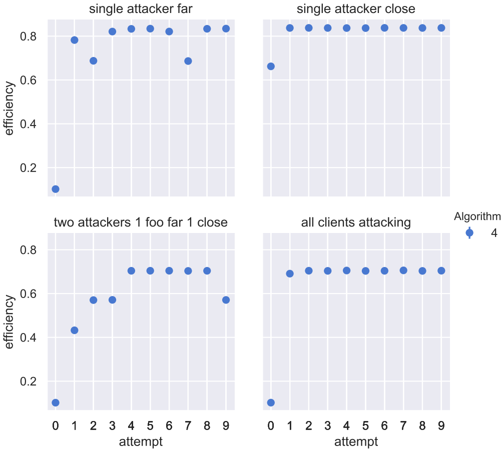
Benevolence based algorithm.

- Assume integrity of alliance members (for now)
- Rank nodes on competence to perform task ‘t‘
- Resolve ties using on benevolence
- Ask node with highest ranking



Computational trust in practice





Conclusion

Main contributions:

- A framework for evaluating defenses in different topologies.
- A method to compare and evaluate countermeasure performance.
- Insights in how to defend collaboratively.

New questions:

- How to resolve conflicting requests?
- How do we optimize for the alliance globally (with limited data)?

Thank you!

For more information (slides, papers, demos):
`https://sarnet.uvalight.net`