BSC RESEARCH PROJECT · CSE3000

Active Search on Post-Compromise Graphs

When does topology-aware planning actually help the analyst?

TU Delft

K. Stefanopoulos

Supervisors: A. Minculescu, F. Oliehoek

Sequential Decision Making Group

MOTIVATION

The Problem

A breach hits. A few machines are infected, most are clean. An analyst can inspect only a handful, one at a time.

Which machines do you check first?

Active search: find the most compromised hosts before the budget runs out, without assuming the layout is known.

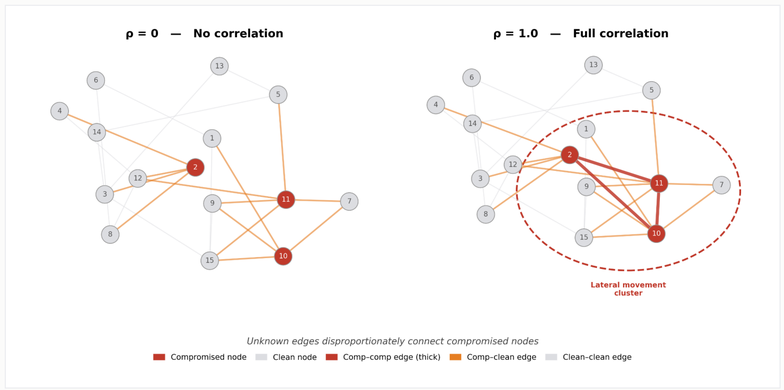
THE SIGNAL

Core Idea: the parameter ρ

Infected machines connect to one another more than to clean ones. The parameter ρ sets how strong that pattern is.

$\rho = 0$
no hidden links

$\rho = 1$
every infected pair linked



As ρ rises, unknown edges concentrate among compromised nodes, forming a lateral-movement cluster.

Methodology EXPERIMENTAL SETUP

≈ 2,300 jobs · DelftBlue HPC · ≈ 46 core-hours

SYNTHETIC · PYTHON SBM

15 nodes · 3 communities · ρ:0→1 · 20 seeds

REAL CAPTURES · TRACES

PCAP·9 | CTU-13·15 | NASim·8

→

Graphs as JSON

nodes = hosts
edges = comms

→

JULIA SOLVER FRAMEWORK

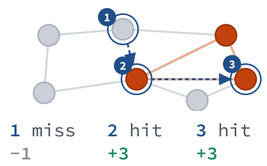
Topology-aware planner
uses the layout to pick the next host

3 baselines
myopic · random · uncertainty

known vs. uncertain

budget 7 · 1 host/step

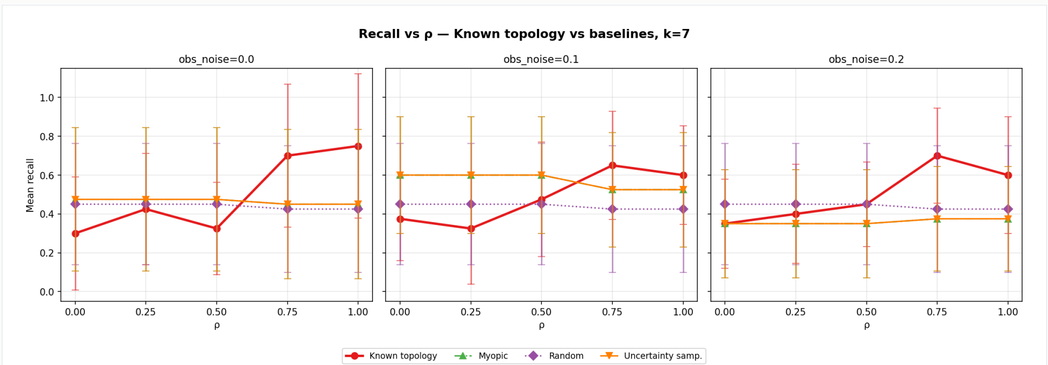
ONE SEARCH EPISODE



Active search: **7 queries** to find as many red nodes as possible.

Each episode records recall · cumulative reward · belief-filter diagnostics.

Main Result FINDINGS



Known topology (red) is the only solver whose recall climbs with ρ; the three baselines stay flat at 0.43–0.48.

2.5× more infected hosts found as the signal strengthens, when topology is known (p = 0.0023).

When the pattern is weak, the planner does ***worse than random***, chasing structure that carries no signal.

BOTTOM LINE

Topology-aware planning wins *only* when all three conditions hold. Miss any one, and random guessing does as well or better.

1 Layout is known
Without it the planner's belief collapses within a few steps.

2 Pattern is strong
If infected hosts are no more connected, there is nothing to find.

3 Network is dense
On sparse graphs even a strong pattern gets lost.

Code & testbed
github.com/KostasStef/CyberSearchPipeline