Lateral Movement Detection Using Distributed Data Fusion

Atul Bohara

PI: William H. Sanders

ACC Seminar, Sep. 28, 2016

Citation: *Lateral Movement Detection Using Distributed Data Fusion.* Ahmed Fawaz, Atul Bohara, Carmen Cheh, William H. Sanders. In *Proceedings of 35th Symposium on Reliable Distributed Systems* (SRDS 2016).

Slide Credits: some slides are taken from the presentation made at SRDS 2016 by Ahmed Fawaz

Introduction

Intrusion resilience

- Monitor the operation of system
- Detect intrusions
- Take response actions

Volume of information that is required to construct a system-wide state can grow rapidly

Long-lasting targeted attacks pose more scalability challenges

Contributions

- A **distributed data fusion framework** for system resiliency.
- An agent-based monitoring and fusion mechanisms to **detect lateral movement behavior** in an enterprise system.
- A host-level monitor to infer **connection causation relations.**

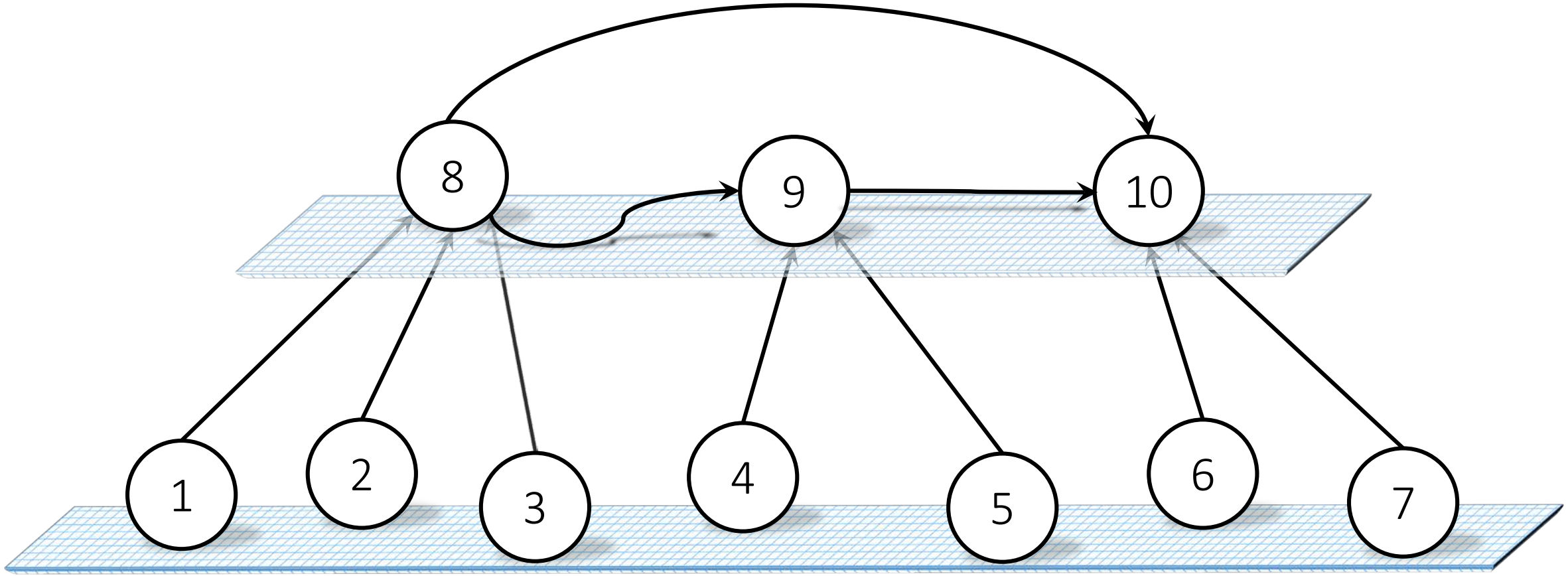
Distributed Data Fusion Framework

We propose a fusion framework, $\mathcal{F} = (G, f, g, \mathbb{T})$, where:

$G = (V, E)$	A graph of agents
$f(.)$	Local transformation function
$g(.)$	Fusion transformation function
$\mathbb{T}$	A set of temporal propositions

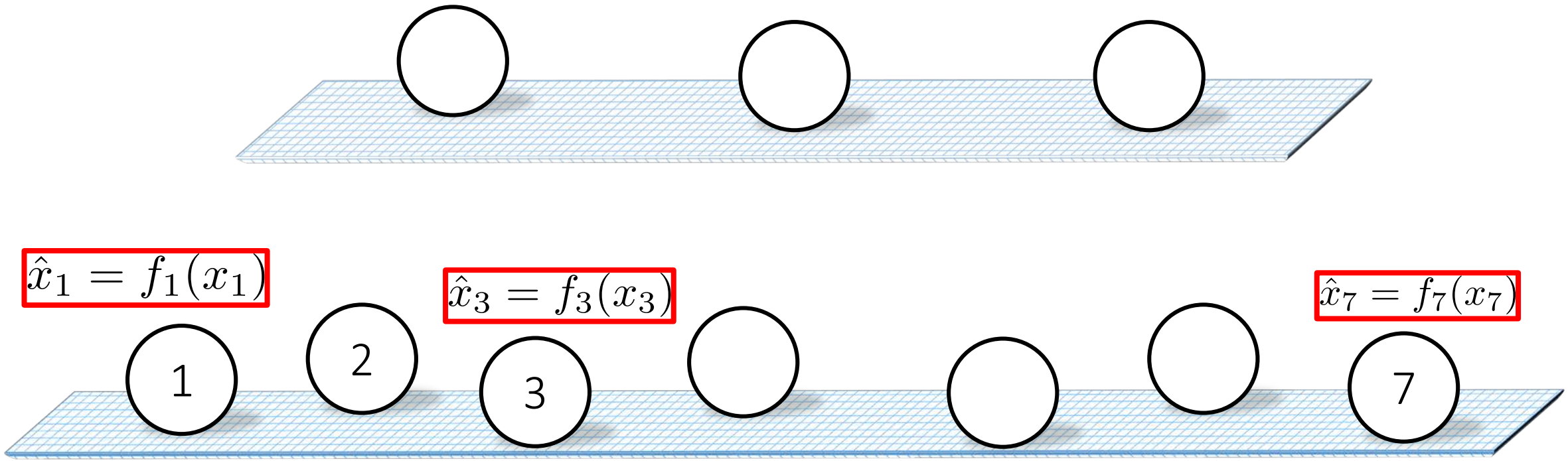
Fusion Graph

A graph where the edges between the **agents** represent communication **channels**



Local Transformation

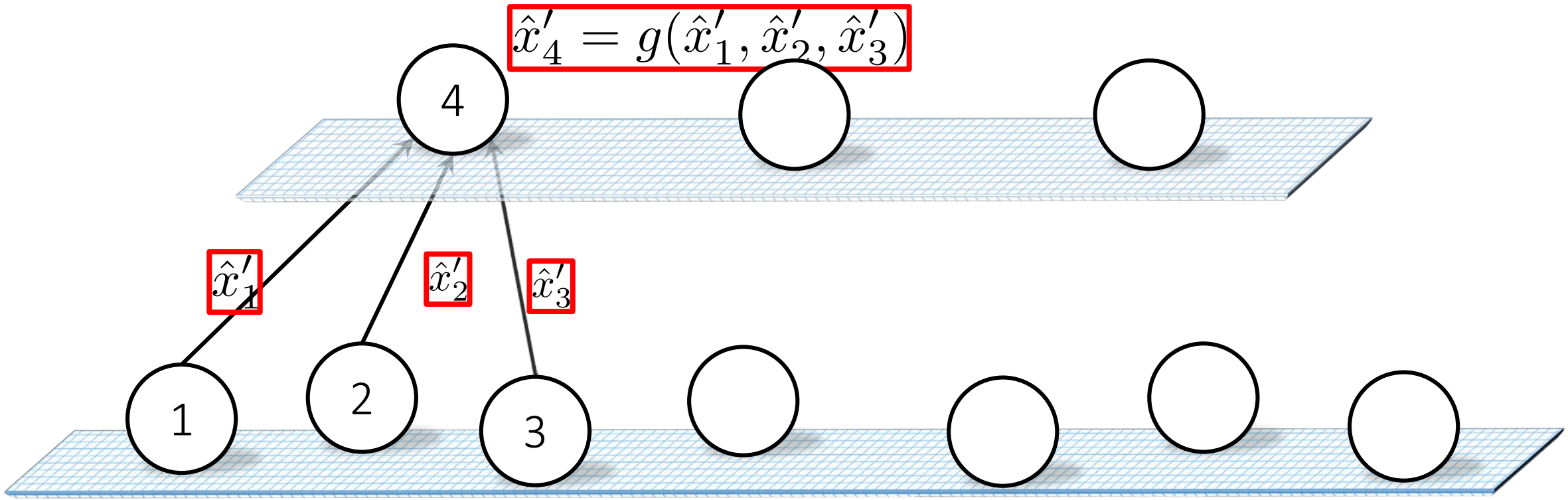
A function f to estimate **local state**



State in node 1 is x_1

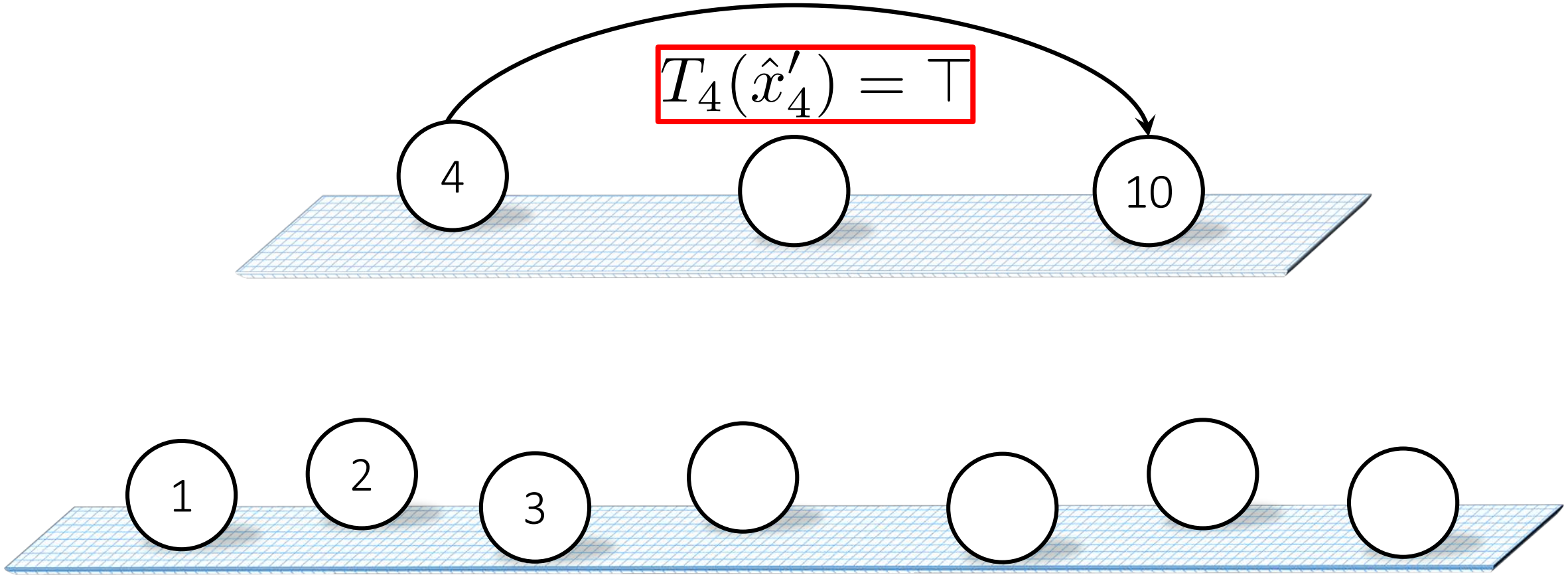
Fusion Transformation

A function g that **fuses and abstracts** local data and received data



Temporal Propositions

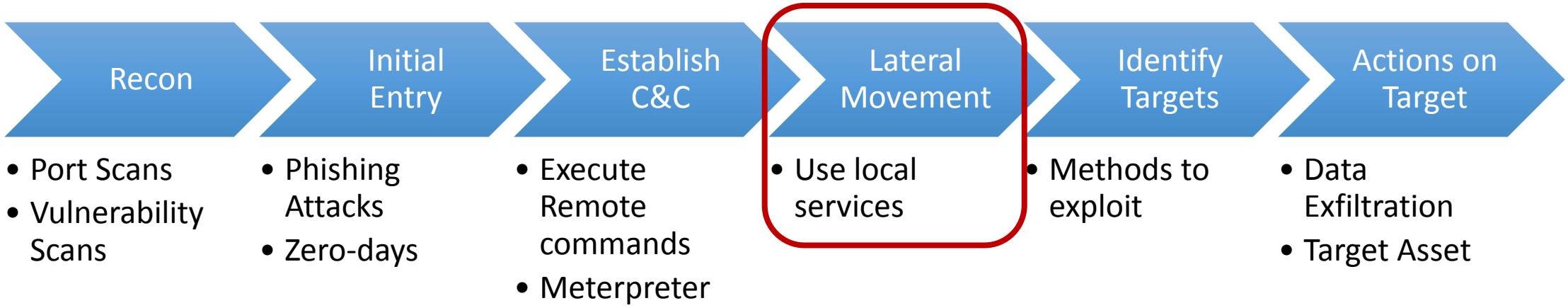
A temporal proposition defines **trigger events**



Lateral Movement Detection

A Case Study

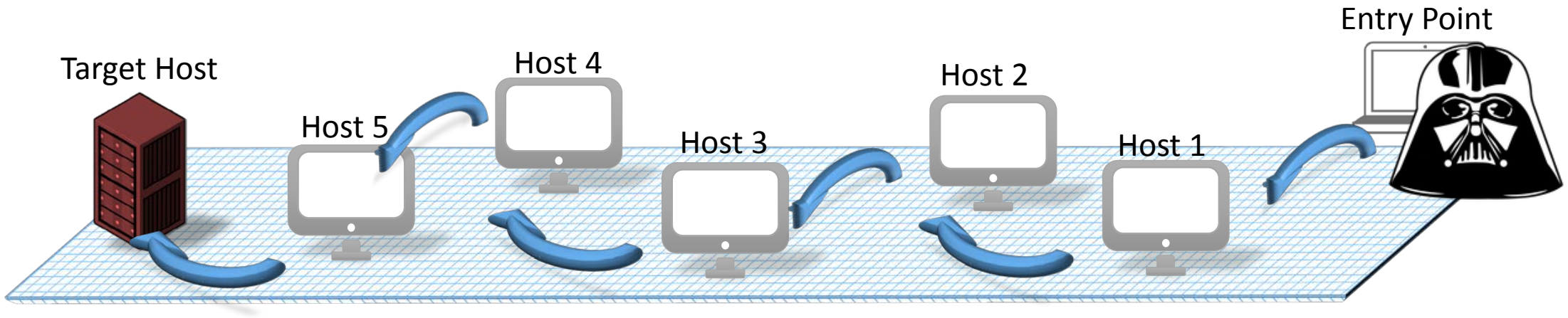
Stages of Advanced Persistent Threat (APT) Attacks



Source: http://www.trendmicro.com/vinfo/us/threat-encyclopedia/web-attack/3142/carbanak-targeted-attack-campaign-hits-banks-and-financial-institutions

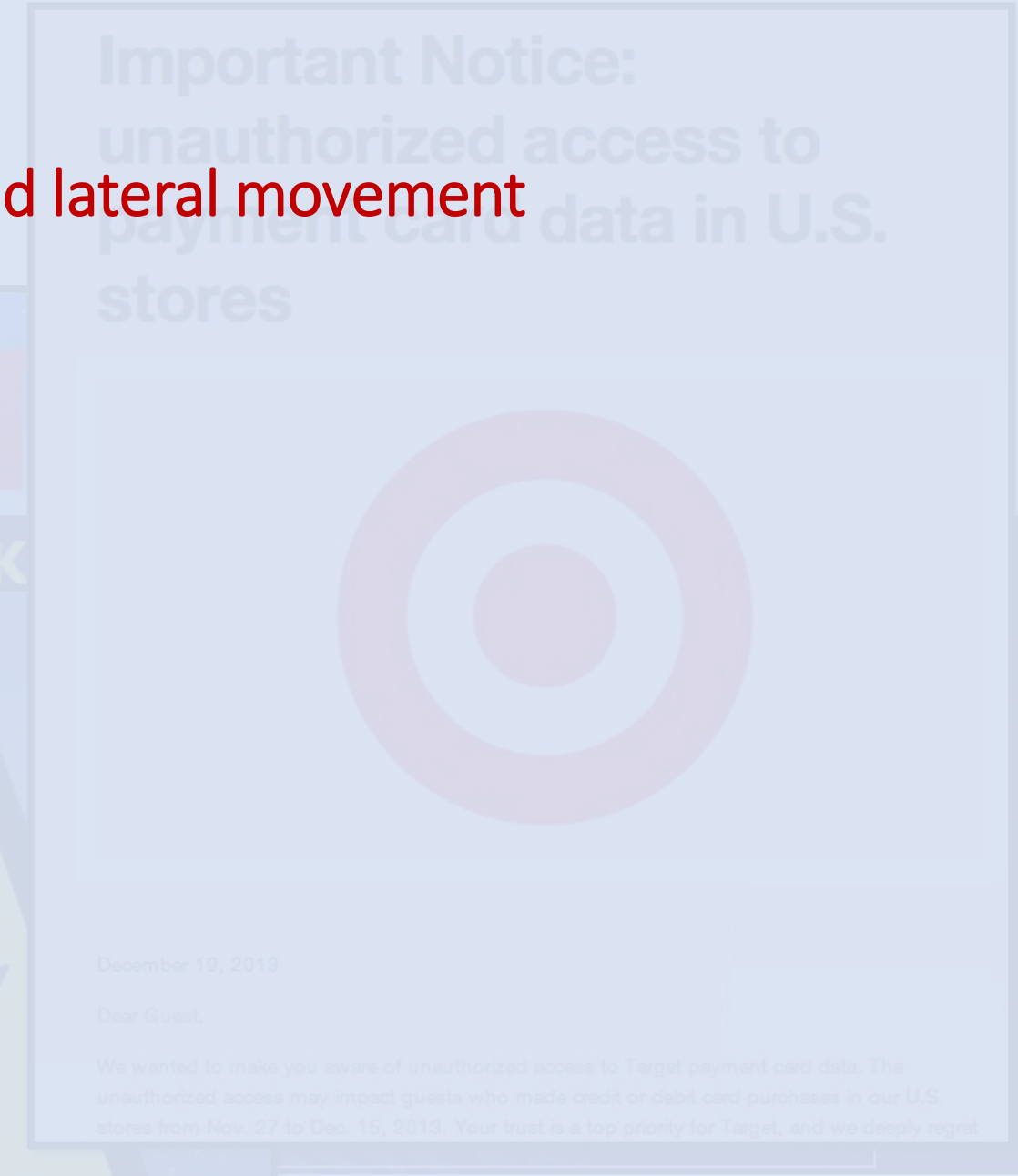
Lateral Movement Explained

- Starting from the entry point attacker moves to target host
- Uses system services or custom tools



In the News

Persistence, stealthiness, and lateral movement



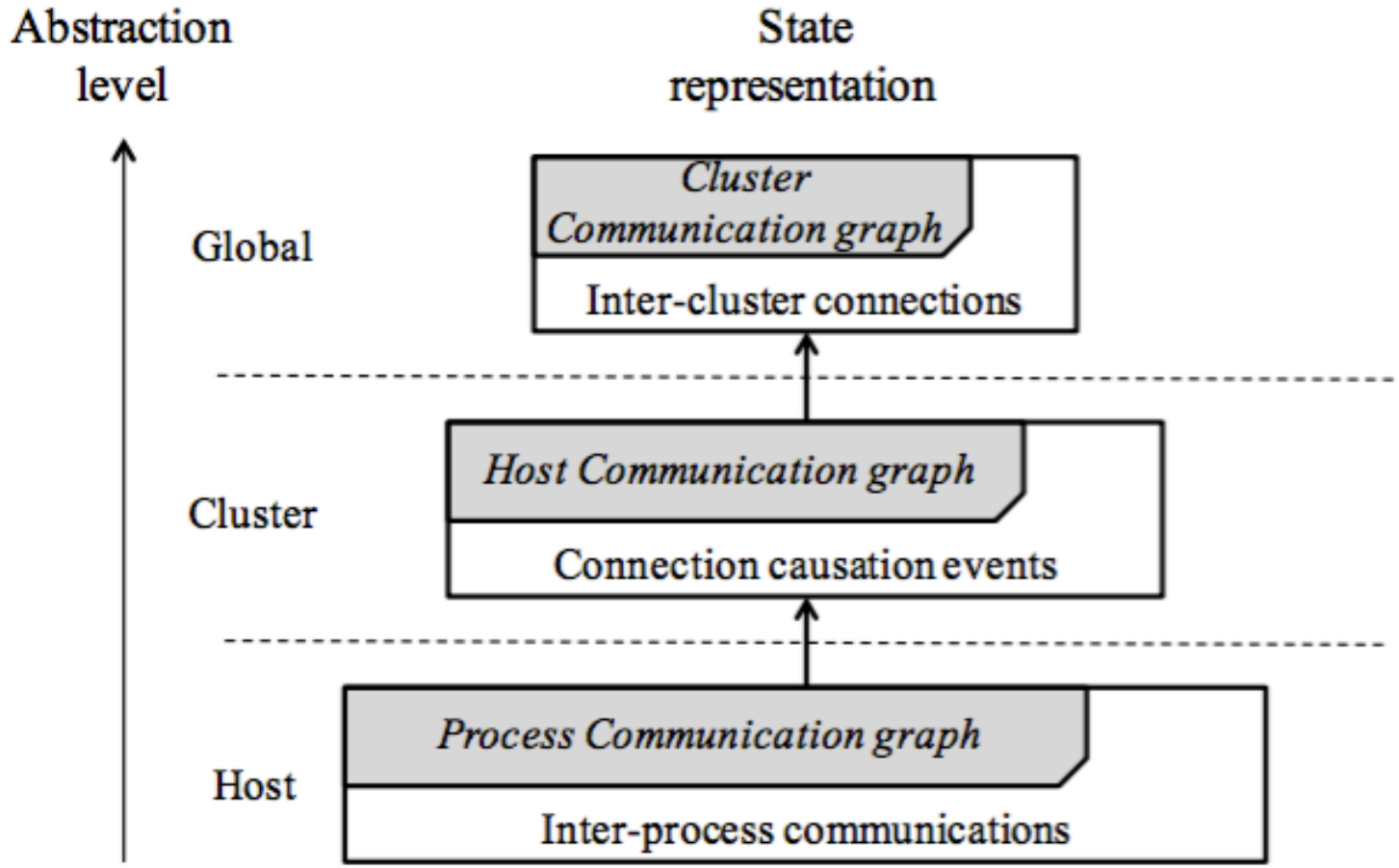
Motivation

Lateral movement detection is challenging

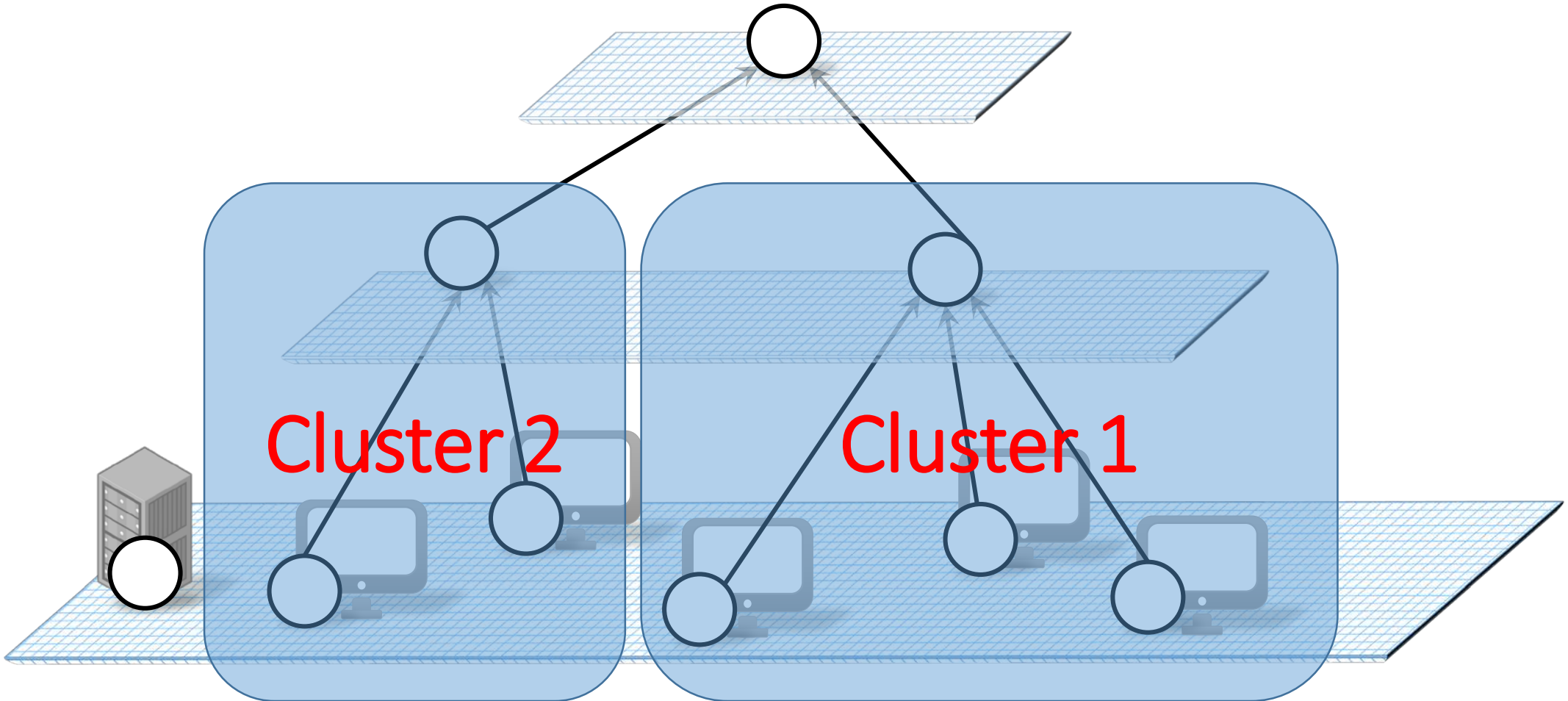
- Need to estimate system-wide state
- Information overhead
- Attacker uses legitimate network services
- Requires a global clock

Lateral movement detection enables proactive prevention and response before the actual damage (e.g., data exfiltration)

Lateral Movement Detection Overview

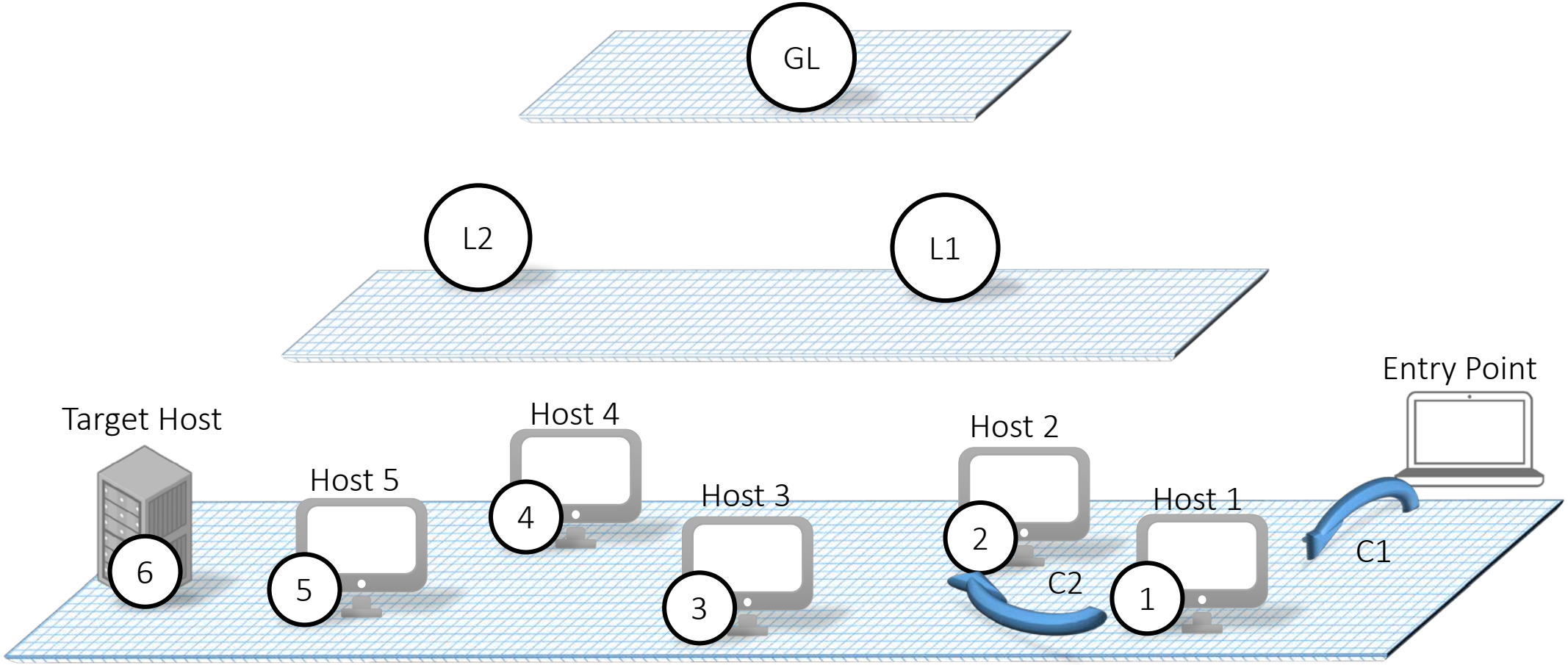


System Model



Lateral Movement

A critical step during APT to move from the entry point to target host



Inside Host 1

Local agent infers connection causation using the **Process Communication Graph**

Collect timestamped events of:

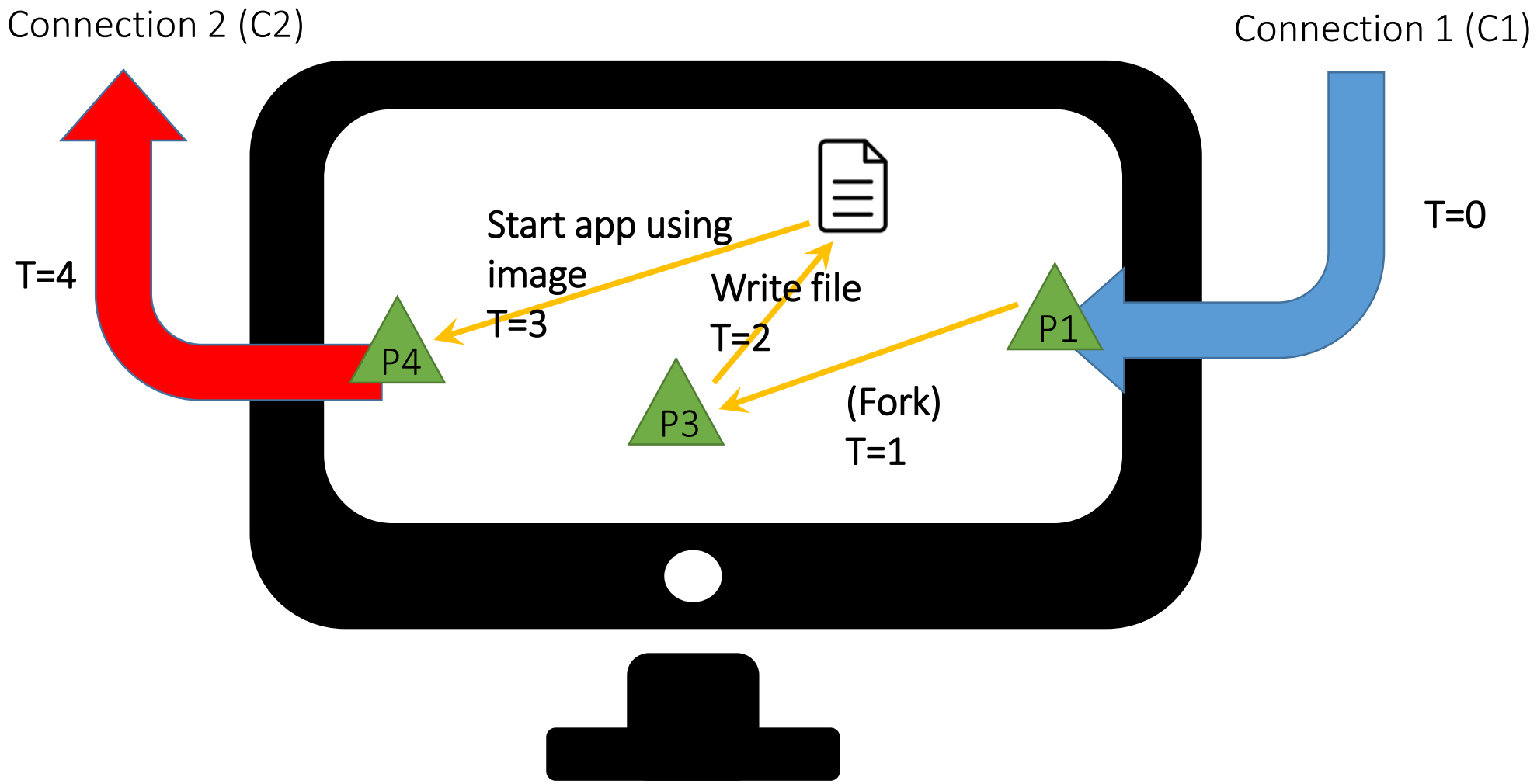
- Processes running
- Process communication (pipes, messages,…)
- Network connections
- File access

The agent creates a timed directed graph of communication between processes

Causation is inferred via a path between incoming and outgoing connections

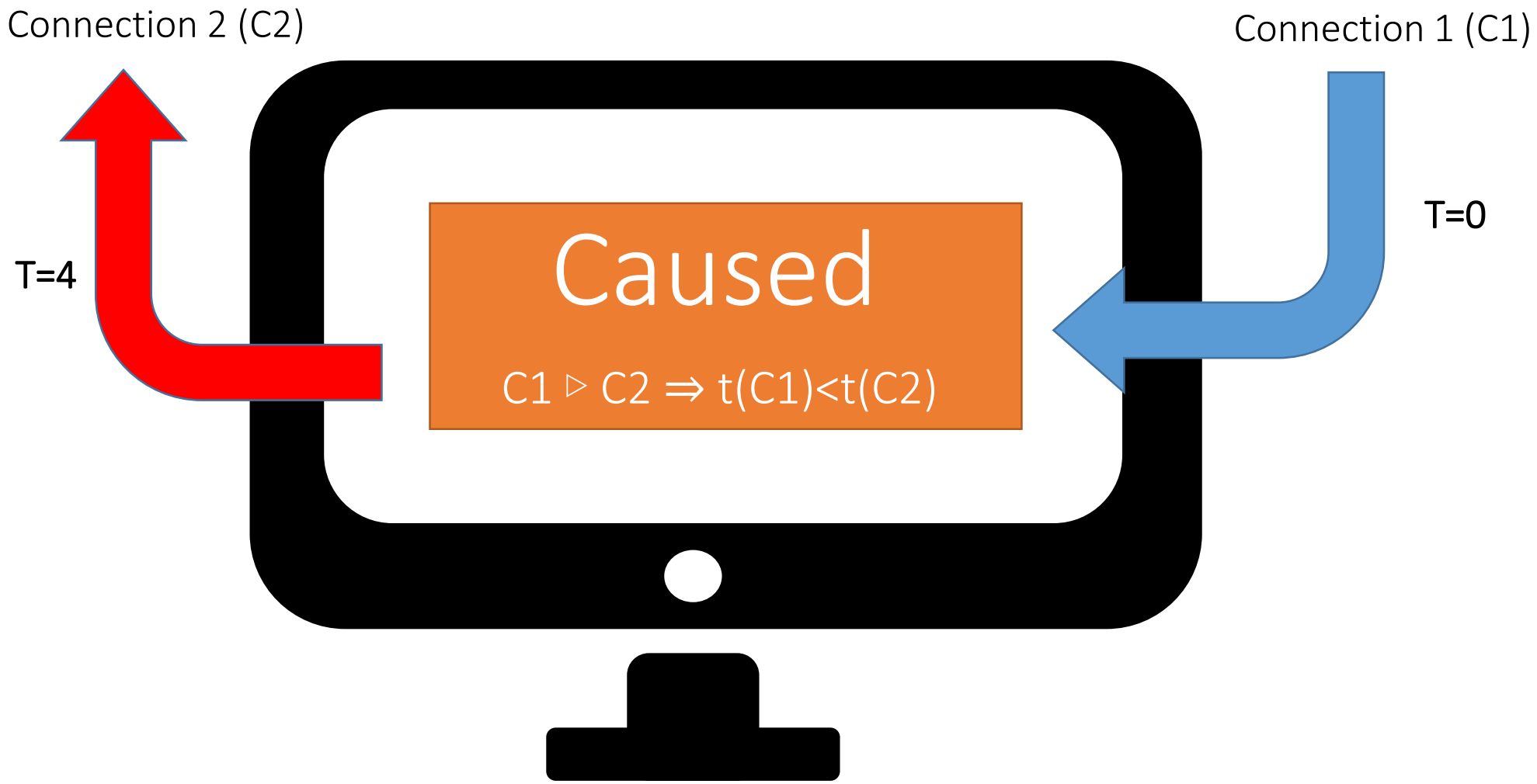
Inside Host 1

Local agent infers connection causation using the **Process Communication Graph**



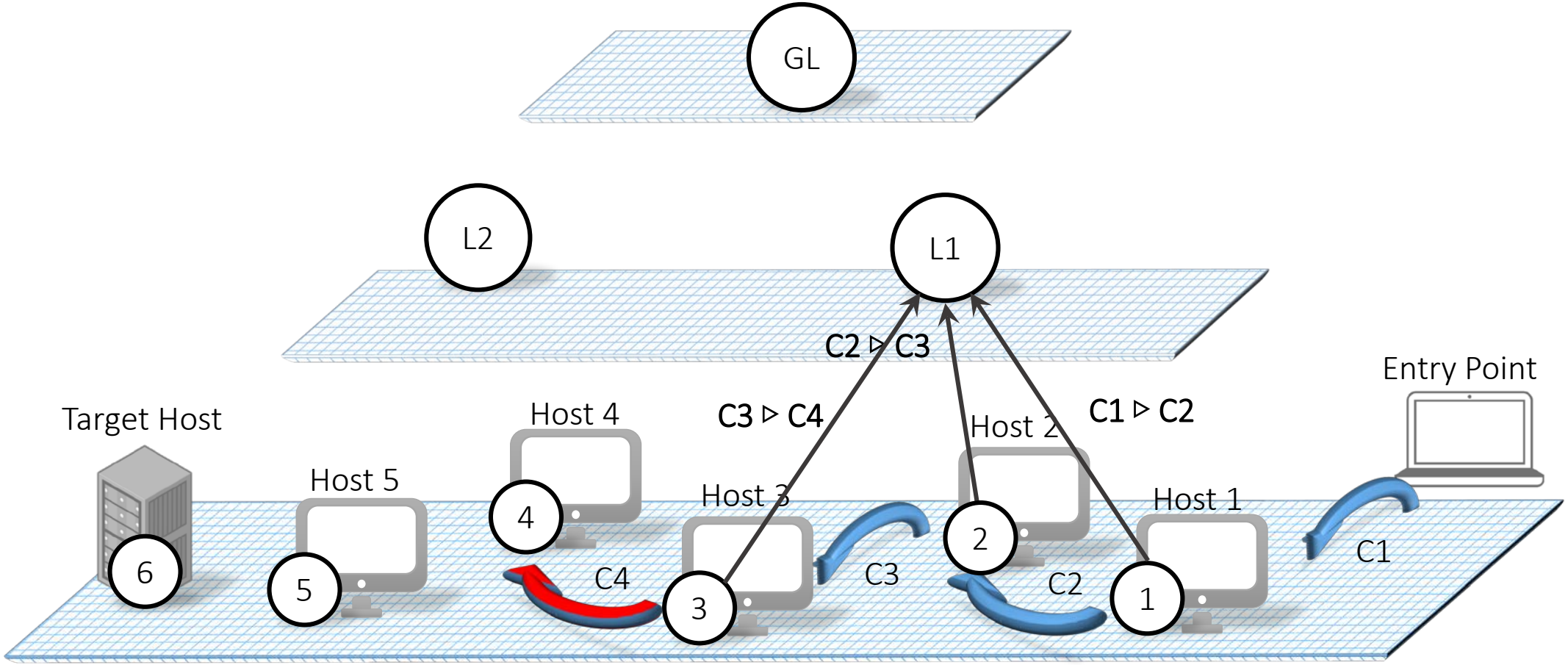
Inside Host 1

Local agent infers connection causation using the **Process Communication** Graph



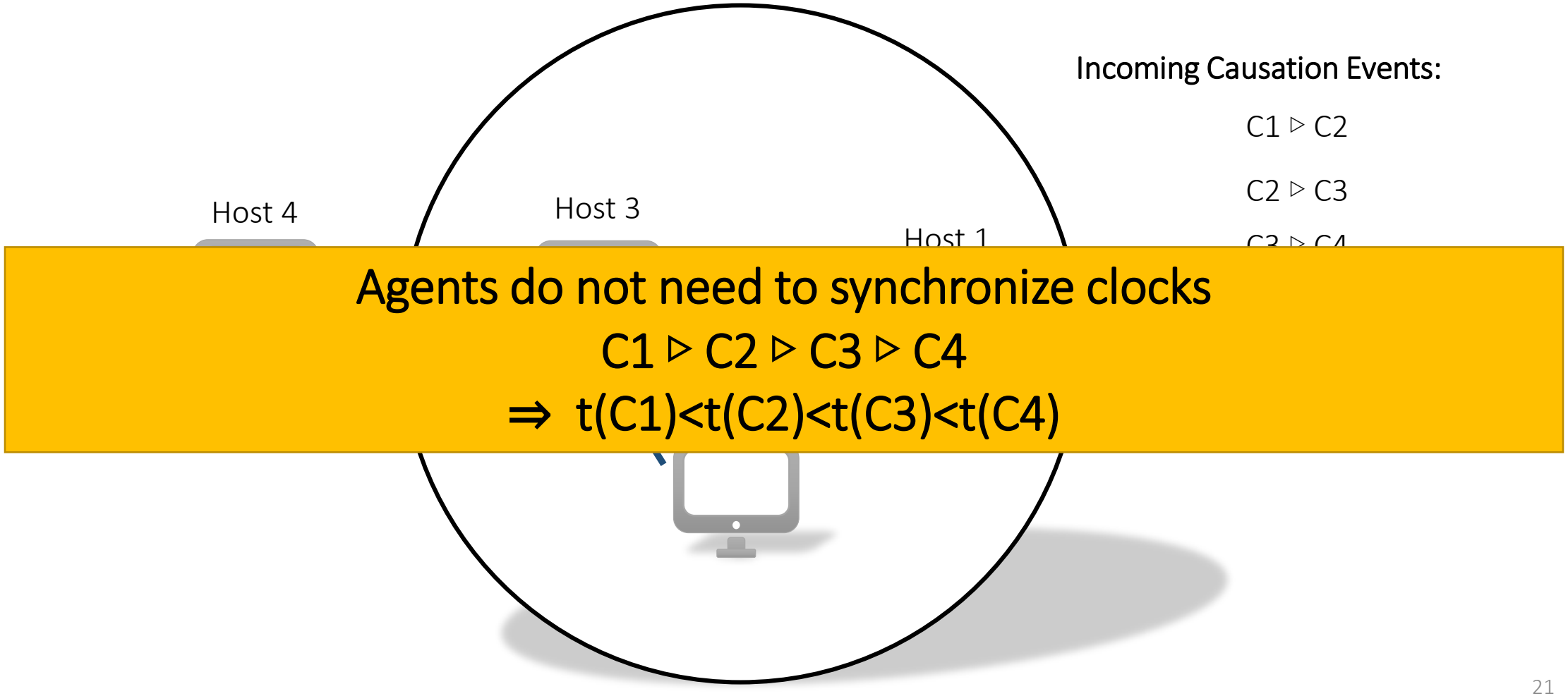
Lateral Movement

A critical step during APT to move from the entry point to target host



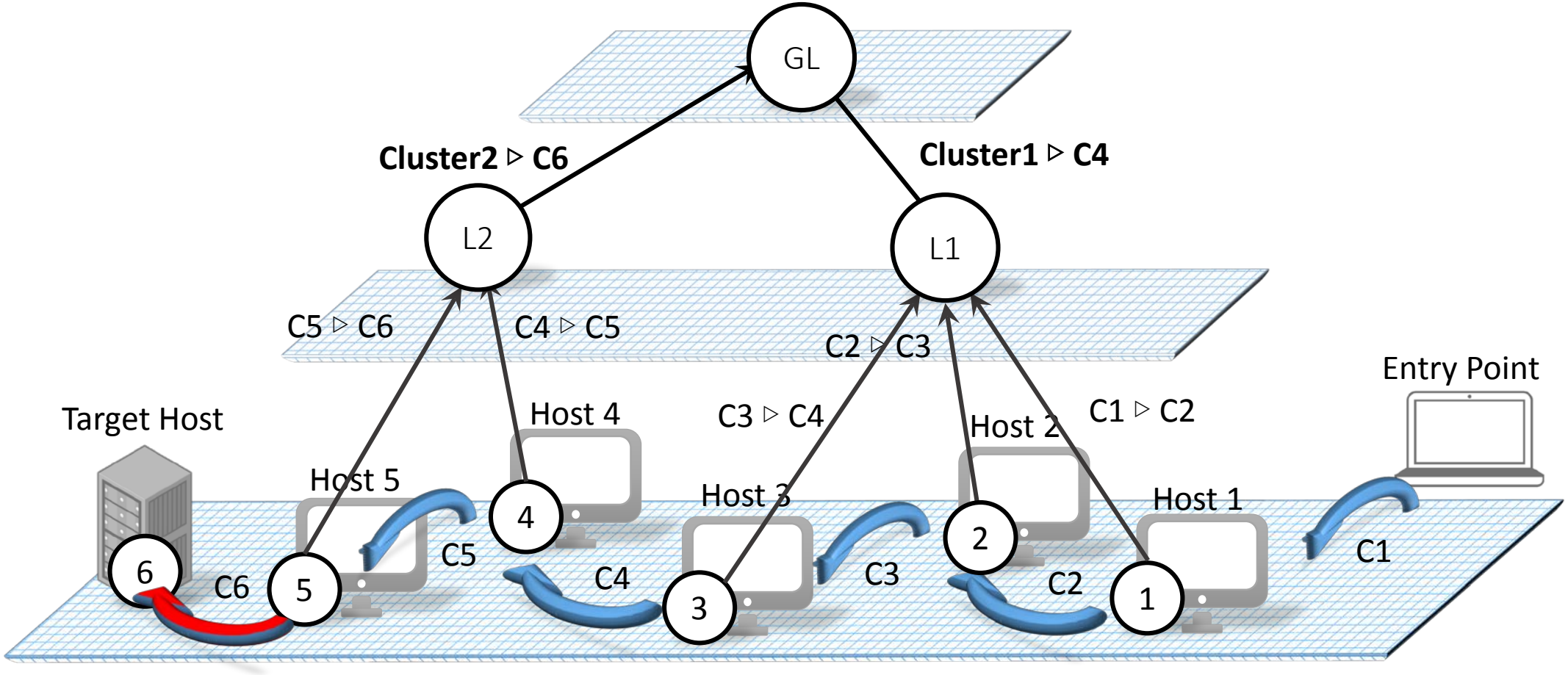
Inside Cluster Leader 1

Cluster head maintains **Host Communication** Graph



Lateral Movement

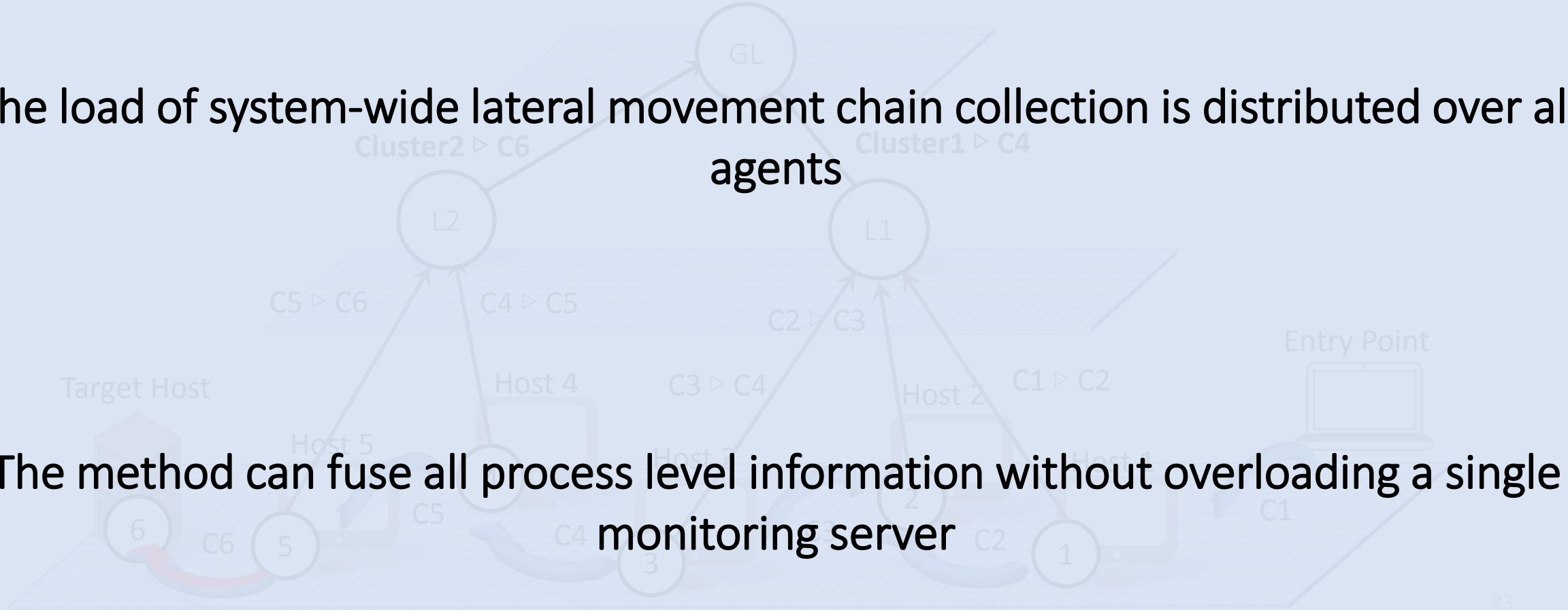
A critical step during APT to move from the entry point to target host



Discussion

A critical step during APT to move from the entry point to target host

The load of system-wide lateral movement chain collection is distributed over all agents



The method can fuse all process level information without overloading a single monitoring server

Results

- Simulation-based evaluation
- Evaluated storage and processing overhead, fairness of resource consumption, and quality of local state
  - Clustering improves the scalability
  - Better fairness and quality can be achieved through topology-aware clustering of hosts
- Implemented a prototype of host-level process monitor
  - Using DTrace on OS X
  - Overhead is manageable

Conclusion

The data fusion framework is a generalized method for fusing monitoring information

Hierarchical fusion framework distributes the fusion loads across the network

Process communication at the host-level infers connection relations

Detection of malicious chains is not investigated

- Work provides a needed step towards the goal

Lateral Movement Detection Using Distributed Data Fusion. Ahmed Fawaz, Atul Bohara, Carmen Cheh, William H. Sanders.
In *Proceedings of 35th Symposium on Reliable Distributed Systems* (SRDS 2016).

BACKUP SLIDES

Image Sources

- Ukrain: http://thehackernews.com/2016/01/Ukraine-power-system-hacked.html
- Target: https://securityledger.com/2015/12/target-agrees-to-pay-39m-to-banks-for-data-breach-reuters/
- OPM: http://thehackernews.com/2015/09/opm-hack-fingerprint.html
- Stuxnet: http://www.mapsofworld.com/around-the-world/recent-hacking-incidents.html