Distributed & Collaborative Worm Detection Experiments

Senthil Cheetancheri, Denys Ma, Jeff Rowe, Karl Levitt
UC Davis
John Mark Agosta, Jaideep Chandrashekar, Denver Dash, Eve Schooler
Intel Corp.

Overview

- Motivation
- Methodology
- Experiments
- Results
- Appendix
  - The Math

Motivation

The Specific Problem

- Often centralized worm defenses are unavailable.
  - Mobile Users
  - Home Offices
  - Small Businesses
  - Network defenses have been bypassed or penetrated
- End-host detectors - last line of defense against large-scale distributed attacks.
- End-host detectors are “weak”
  - Without specific attack signatures, false-positives are high.
  - Local information insufficient to infer a global phenomenon.

- Can ‘weak’ host-based detectors collaborate to produce a ‘strong’ global detector?
- If yes, how such a federation can detect worm attacks?

Our Approach…

- Motivated by
  - Portscan Detection using Sequential Hypothesis Testing

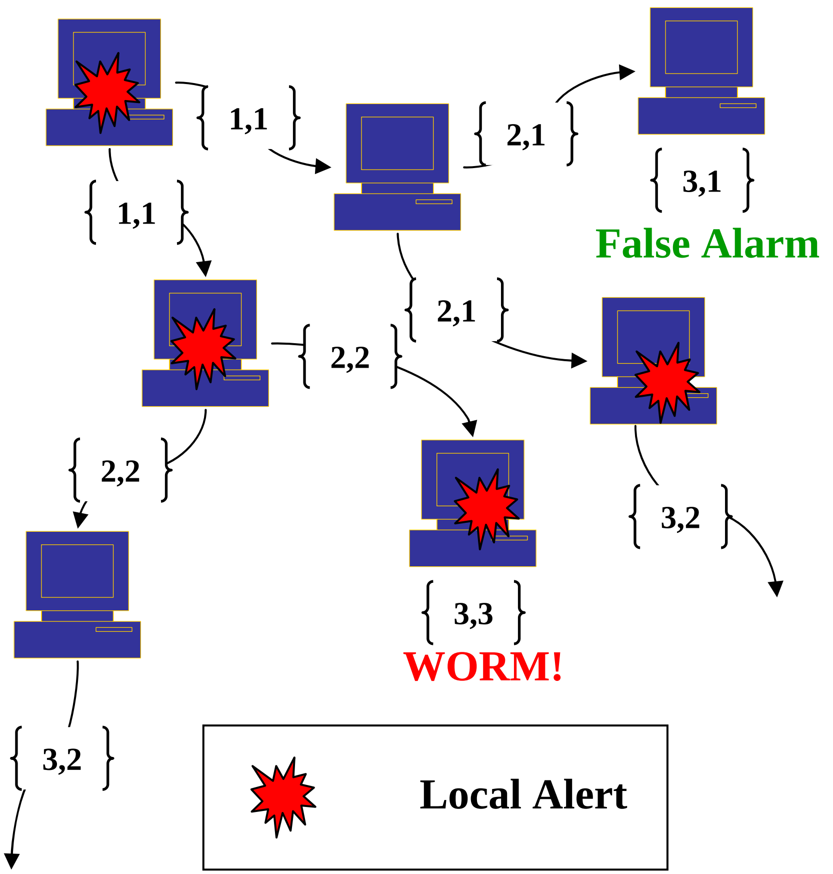
    Jung, J., Paxson, V., Berger, A., Balakrishnan, H., *"Fast Portscan Detection Using Sequential Hypothesis Testing"*, Proceedings of the IEEE Symposium on Security and Privacy, 2004
  - Corraborative Intrusion Detection and Inference

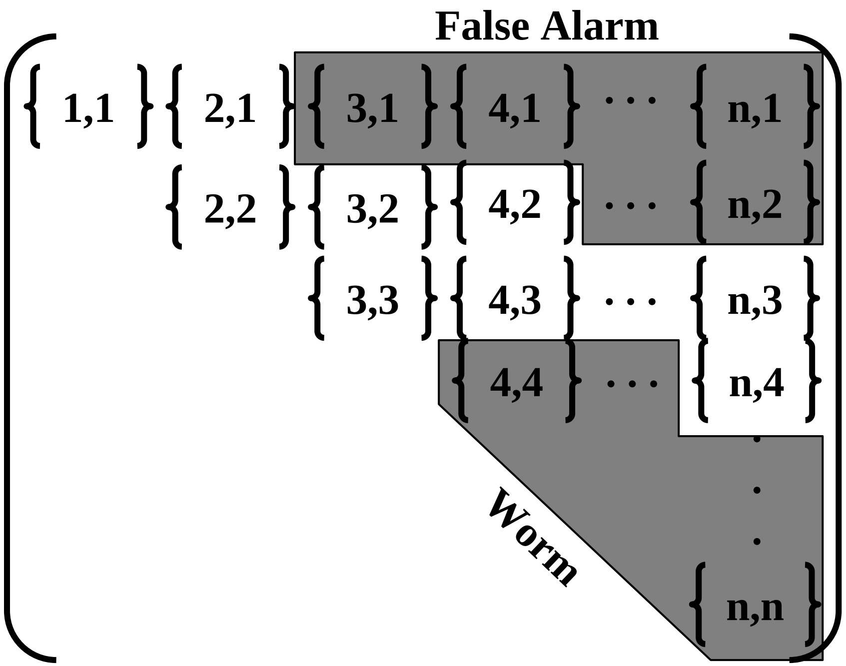
    Agosta, J.M., Dash, D., Schooler, E., Intel Research

- Designed a Protocol for distributing alert information.
- Developed a Probabilistic model for a federation of end-node detectors to infer worm attacks co-operatively.

Methodology

Distributed Decision Chains (TRW)



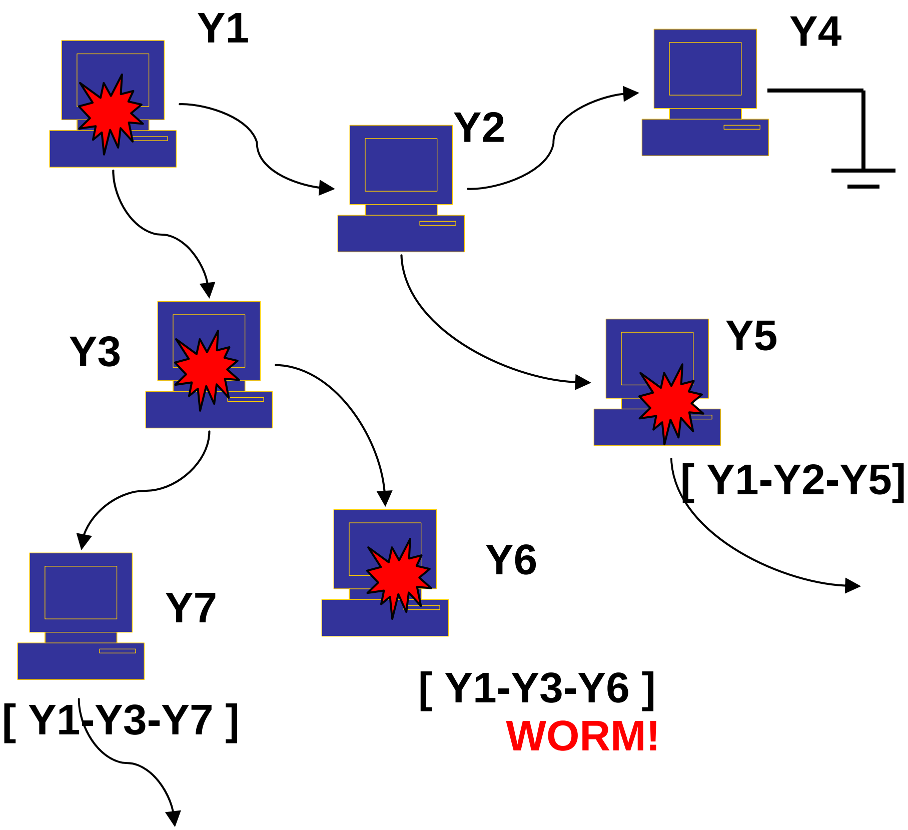


Matrix of Likelihood ratios

Thresholds: T0, T1

Protocol Message: { i, j, A} - *j* local alerts for anomaly *A* seen after *i* steps

Redundant Testing of Hypothesis



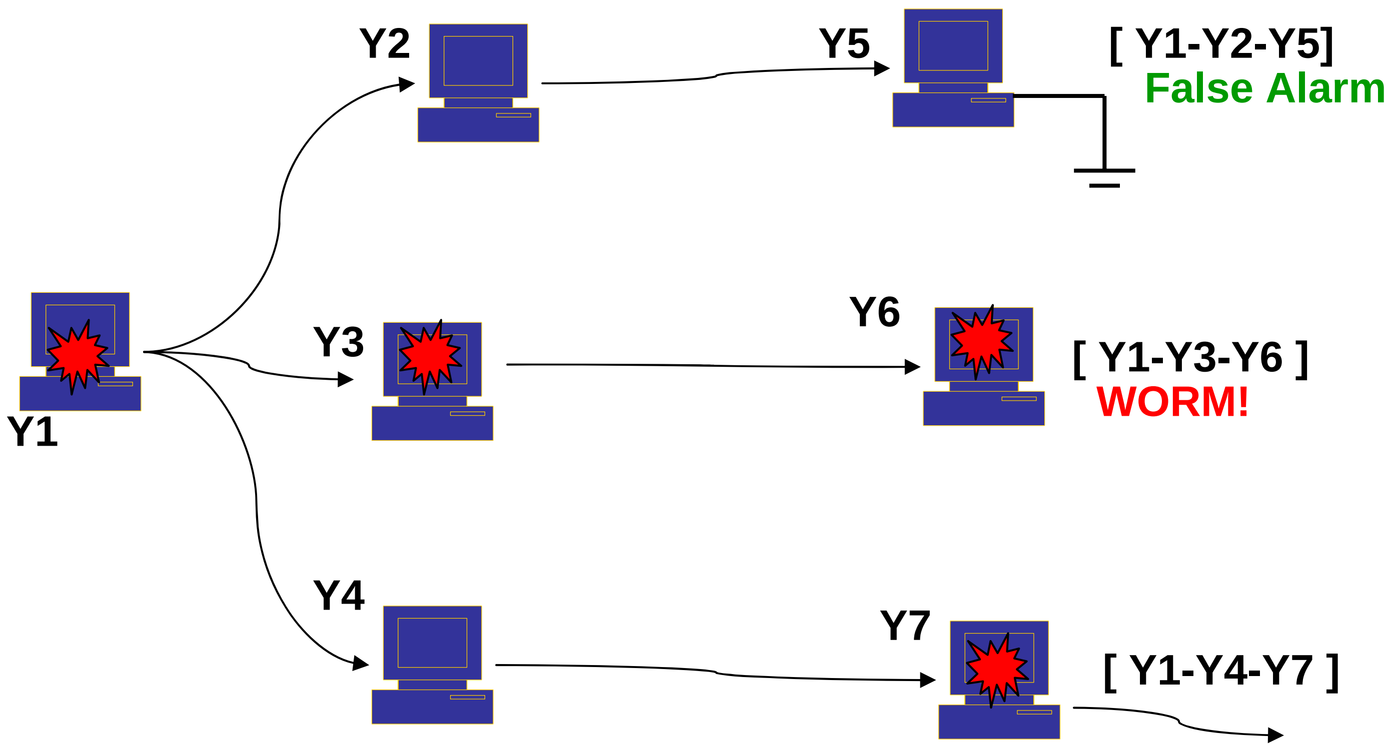
Tree is efficient but…

-Decision heavily biased on the first couple of nodes

-Redundant information

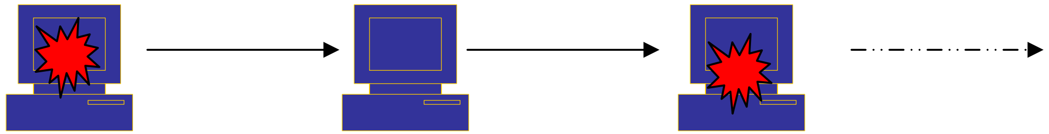
-Maximize information aggregation / generation

Optimal Testing of Hypothesis



- Each node needs to maintain the partition to be used
- This partition needs to be randomized to avoid targeted attacks
- Memory and computation overhead

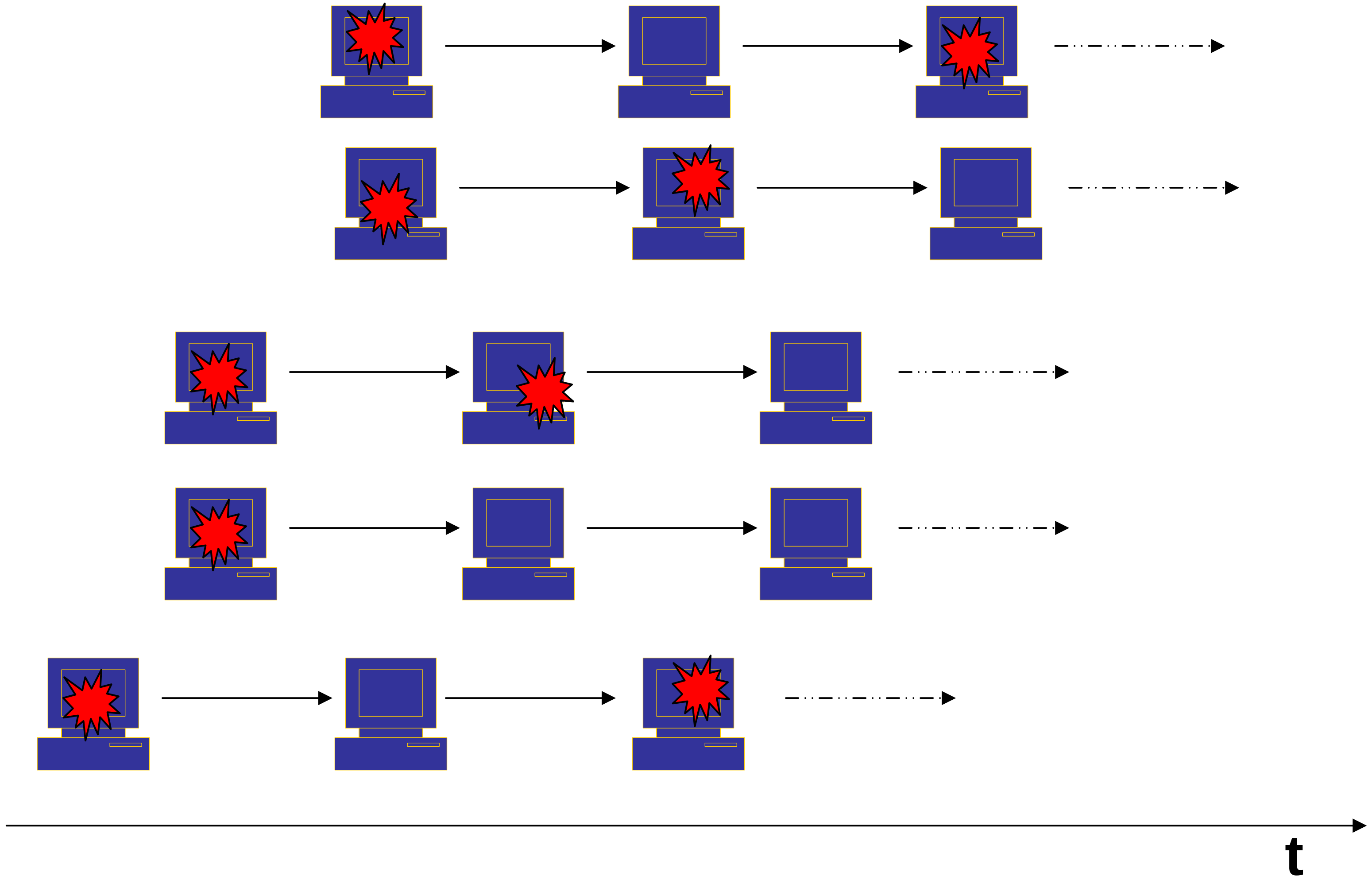
Further Optimization



•Just have one chain

•Randomly choose next node

Rationale



Rationale

- As infection progresses, several parallel chains
- Effect of double counting (redundant info.) is negligible

Experiments & Results

Experimental Set-up

Goal:

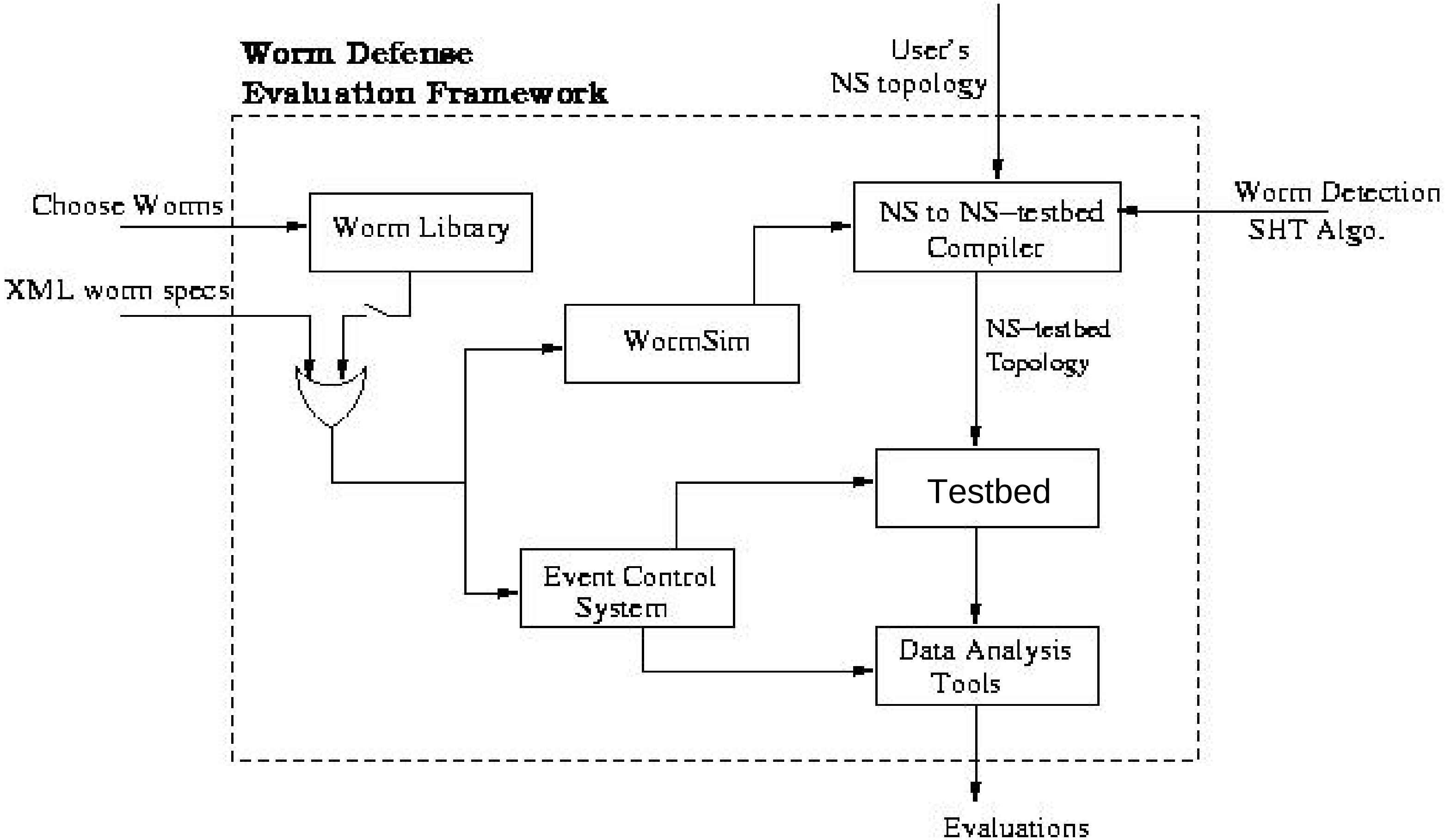
- **Catch atleast 98% of worms**
- **Miss a maximum of 2% of worms**

Resource:

- **Weak end-host IDSs**
  - **raise alarm for every gratituous connection attempt**

- **Miss 1% of anomalies (*fn*)**
- **False positives = variable for each expt. (*fp*)**

Worm-Defense Evaluator

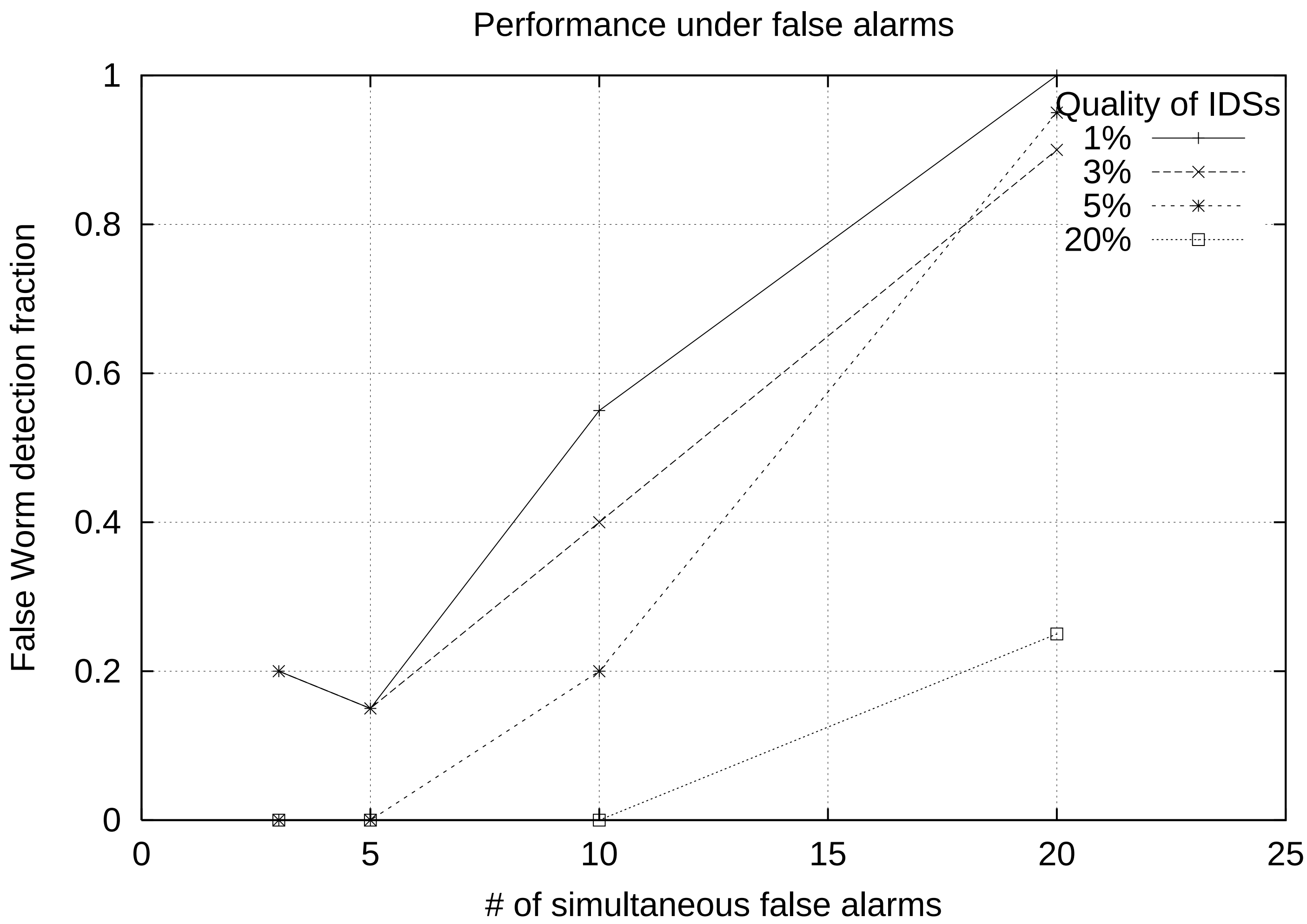


[McAlerney'04], [Malware'06]

Experiment 1- False Alarm Performance

- Traditionally, *fp* rates are used to test fidelity of IDS.
- We use *fp* to denote quality of end-host IDSs.
- Several simultaneous distributed false alarms.
  - How many do we need to trick our system?

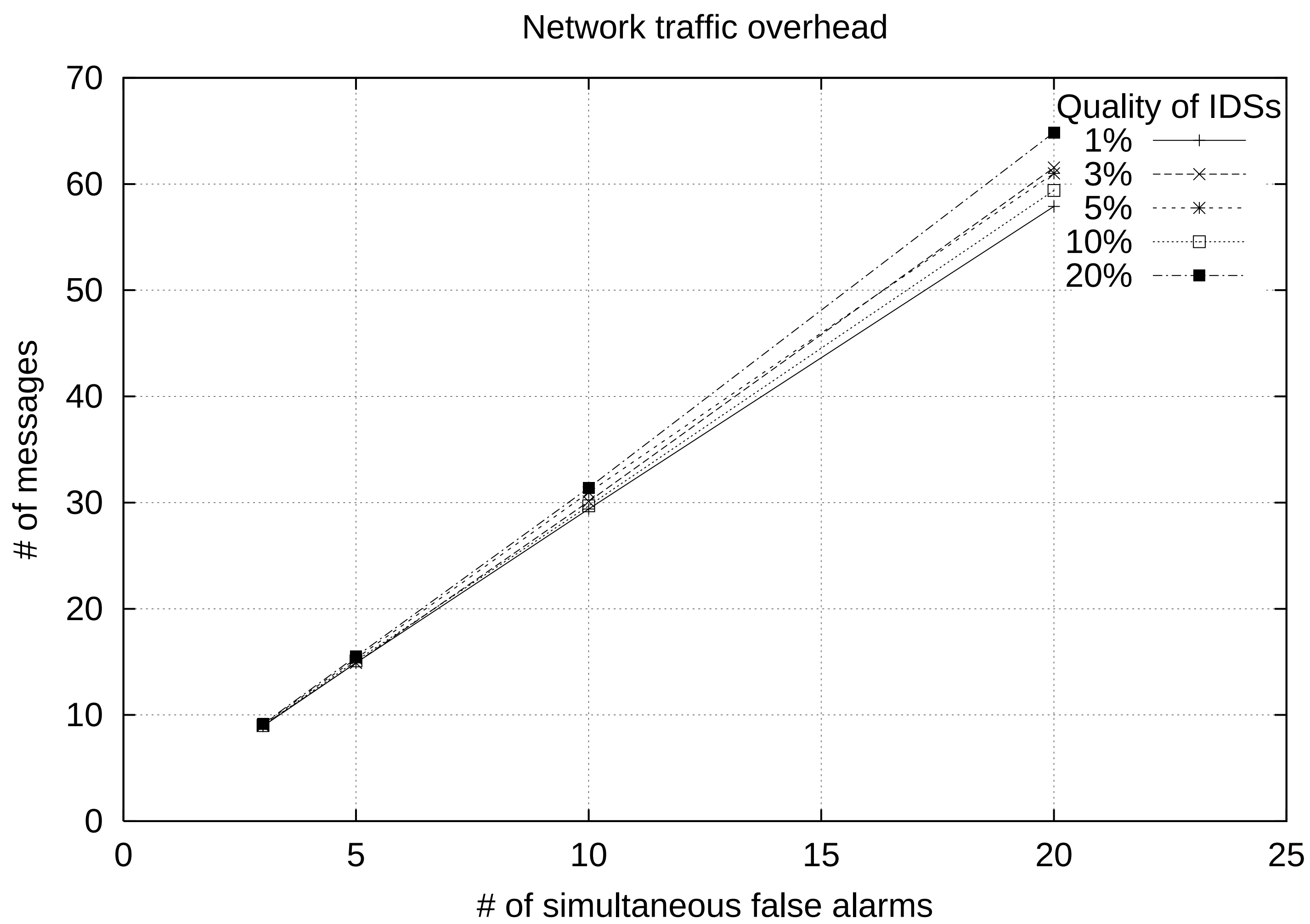
Crying worm!


Performance under false alarms
False Worm detection fraction
# of simultaneous false alarms
Quality of IDSs
1%
3%
5%
20%
0
0.2
0.4
0.6
0.8
1
0
5
10
15
20
25

Network Overhead of Protocol

- Bandwidth needed during normal operation
  - # of messages exchanged due to false alarms
- Doesn’t vary greatly with the quality of the end-node IDS
- Grow linearly with # of simultaneous false alerts
  - Opposed to quadratic or exponential flood

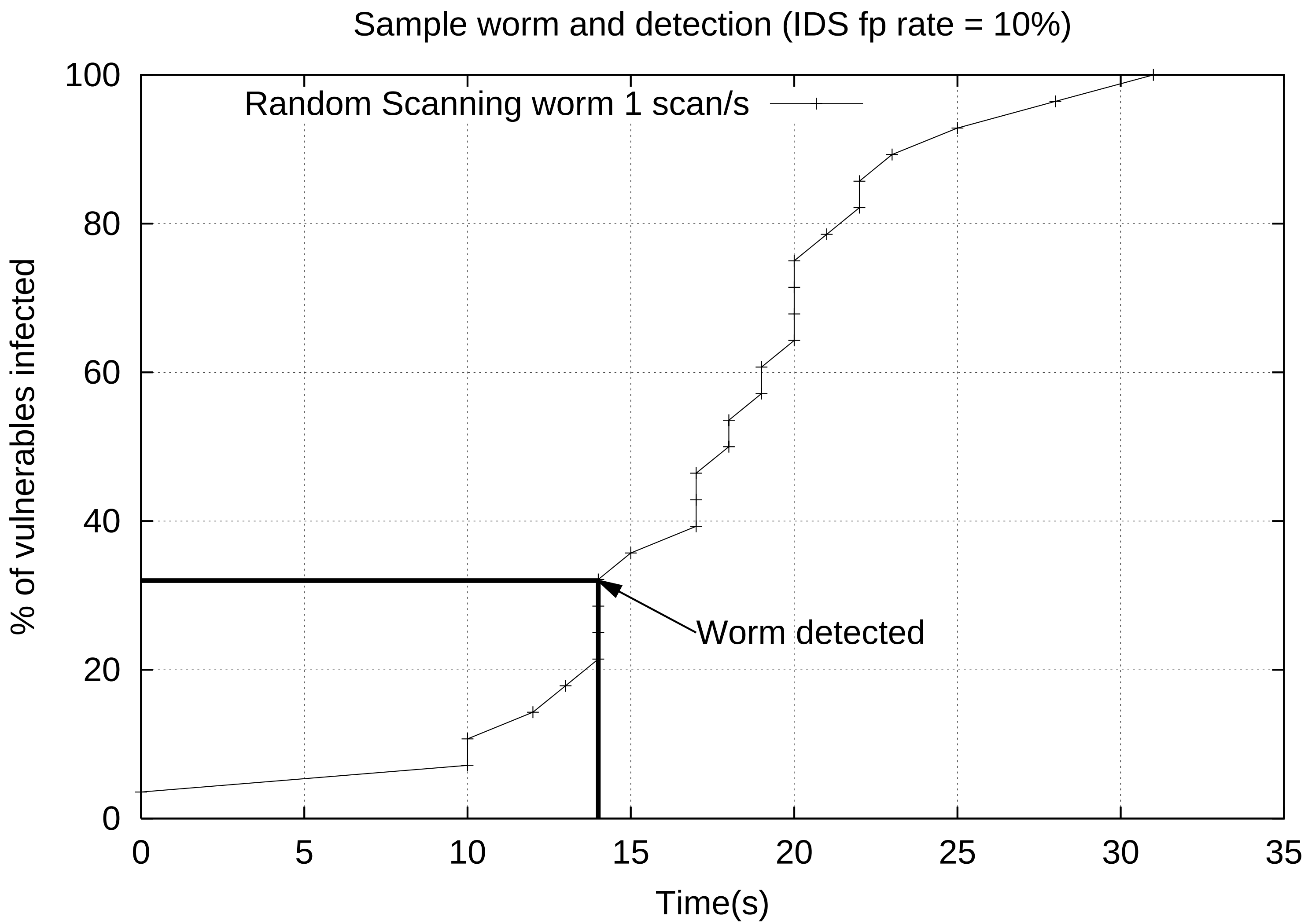
Protocol Overhead (Bandwidth)



Experiment 2 –
Worm Detection

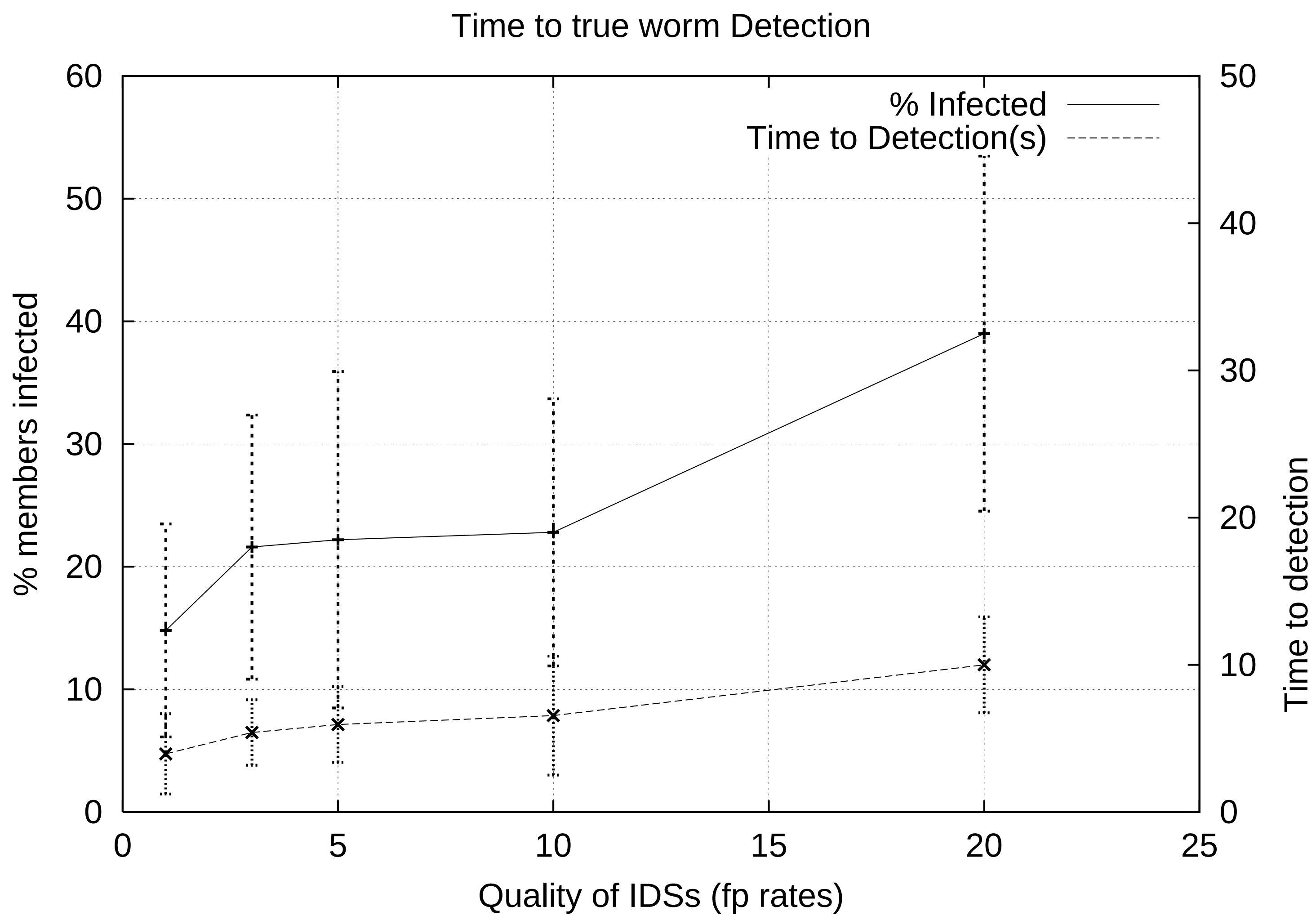
- 25% of participants are vulnerable
- End-node IDS – raise alert for any gratituous cxns.
- Vulnerable nodes' IDS can't detect attacks
- Random Scans – 1scan /second.

Sample Worm



Sample worm and detection (IDS fp rate = 10%)
Random Scanning worm 1 scan/s
Worm detected
% of vulnerables infected
Time(s)
0
20
40
60
80
100
0
5
10
15
20
25
30
35

Effectiveness



Future Work

- Scale-down node
  - To emulate worm traffic from the rest of the Internet
- Background Traffic
- Anomaly vector

Message

- Unity is Strength !

Papers & Theses

- Wormsim [McAlerney] - MS
- Worm Defense Evaluator [Malware '06]
- Current Work [LSAD '06]