SOARTECH

DUDE:
Dynamic Understanding of DNS Events

H. Van Dyke Parunak, Alex Nickels, Rich Frederiksen
Soar Technology, Inc.

SOARTECH

Big Idea: Find a transformation of the data in which the answer is obvious.

We focus on two transformations:

- Data → bipartite graph: keeps all analyses **local**
- Data → dynamics: distinguish beaconing from non-beaconing activity

Preparation

Objectives

- Reduce data size
- Avoid artifacts

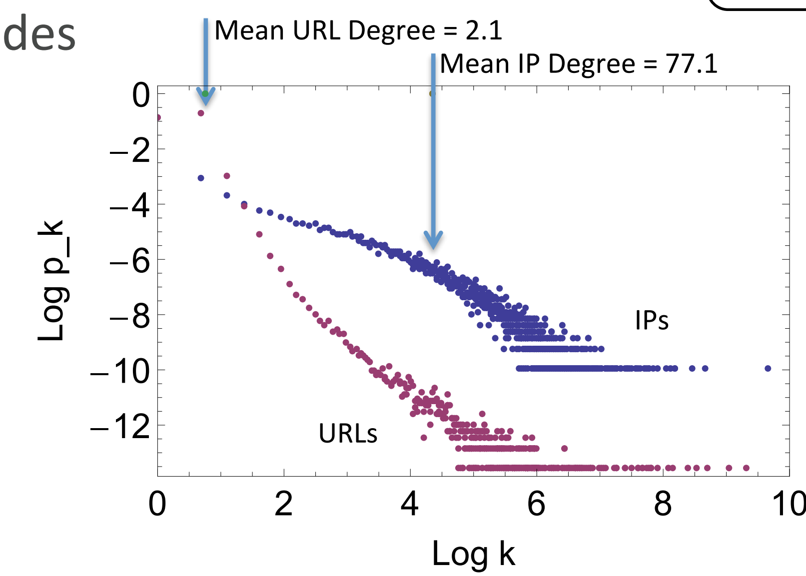
Actions

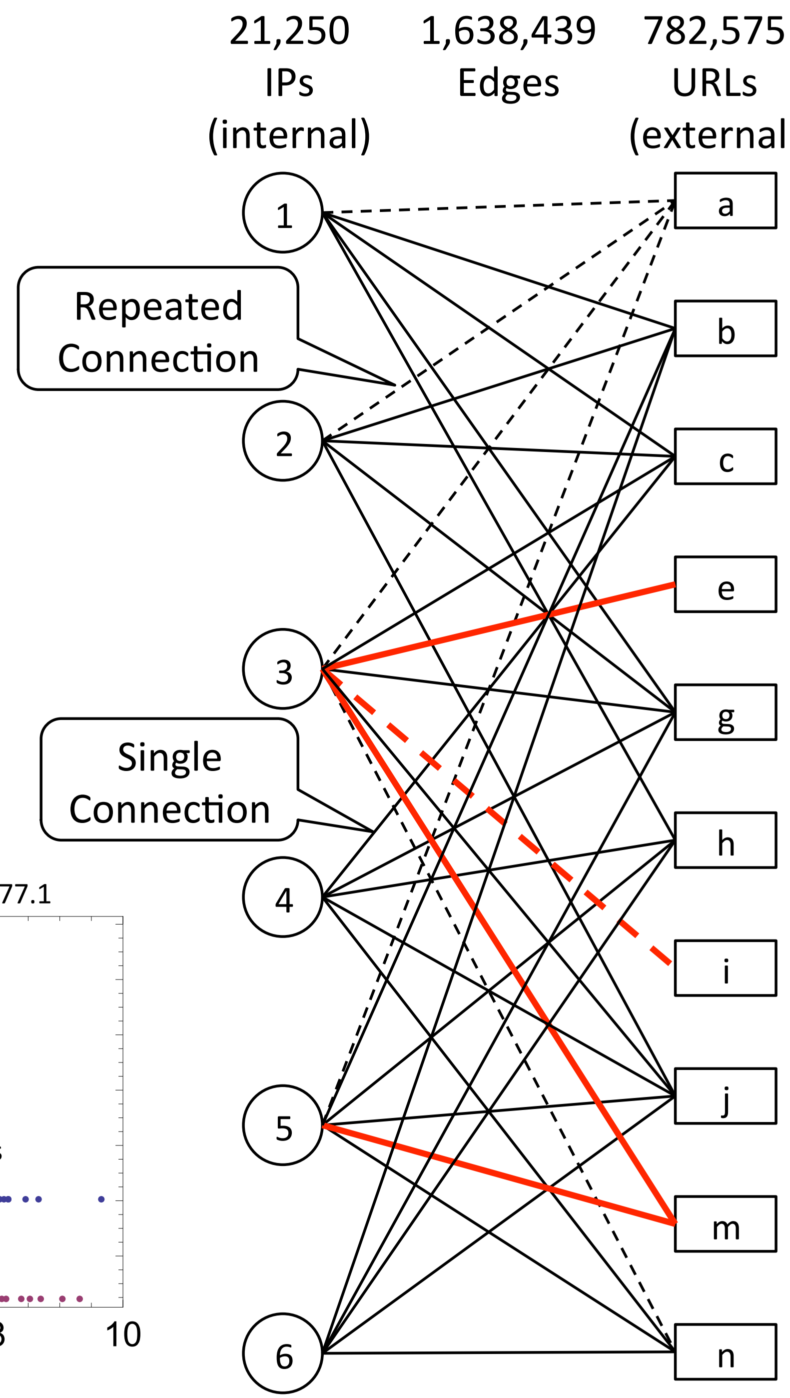
- Round all times to nearest second
- Count each response to the same IP for the same URL only once, regardless of # of IPs returned
- Collapse DNS resolution chains: retain records only from IPs that never respond to a DNS query
- Whitelist URLs from Month 1

The Graph

- **All relevant data is local** in the graph
  - Connection times (on edge)
  - Degree of URL
  - Propagation
- **Scale by distributing** across CPUs
- Agents on each processor
  - Implement different heuristics
  - Interact by annotating the graph

→ Giant component has 800,472 nodes; 1500 smaller components total 3353 nodes

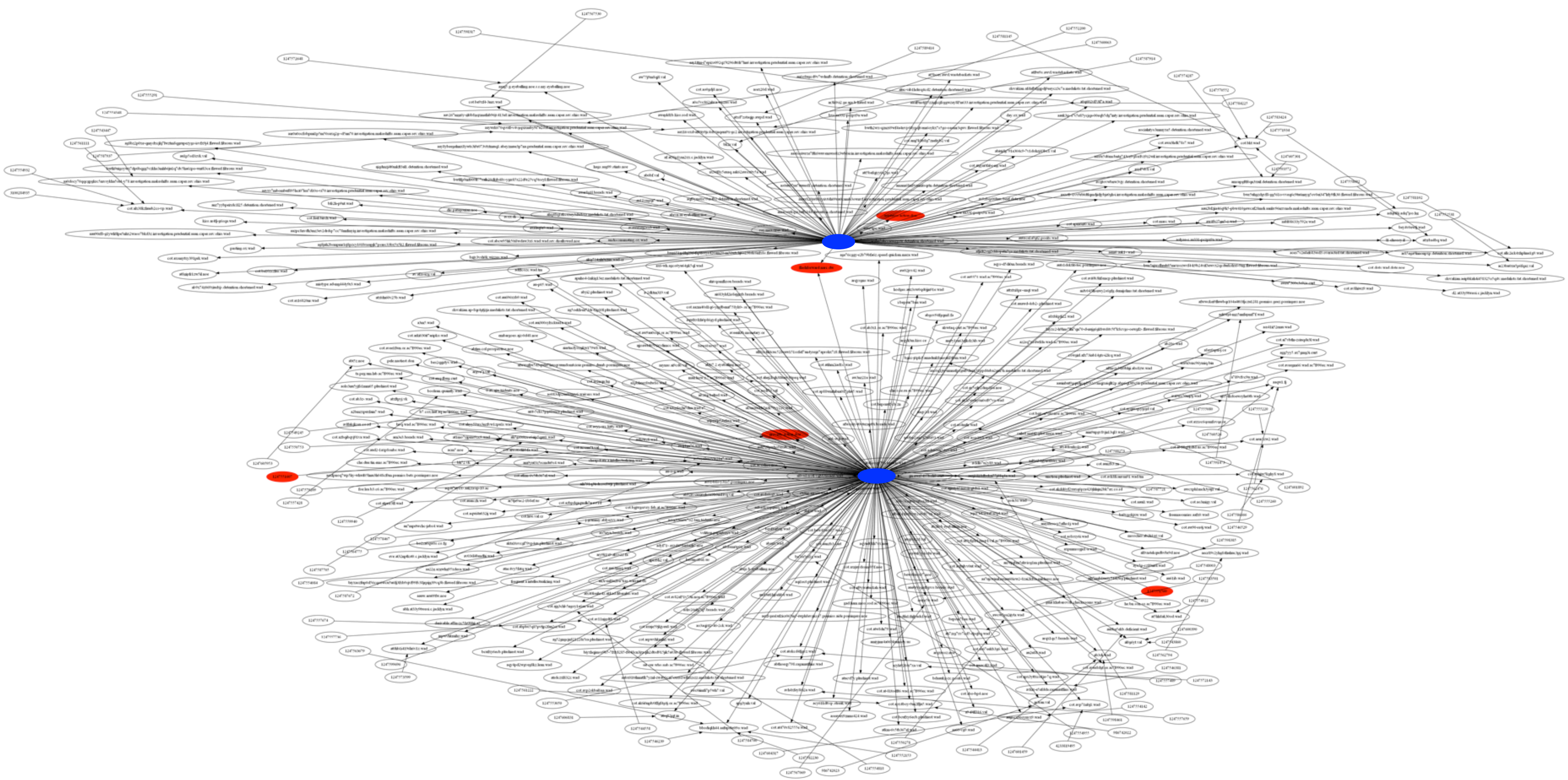




Detecting Connections

Look for

- **IPs** adjacent to **rare URLs** (< 5 IPs)
- **Rare URLs** adjacent to suspect **IPs**



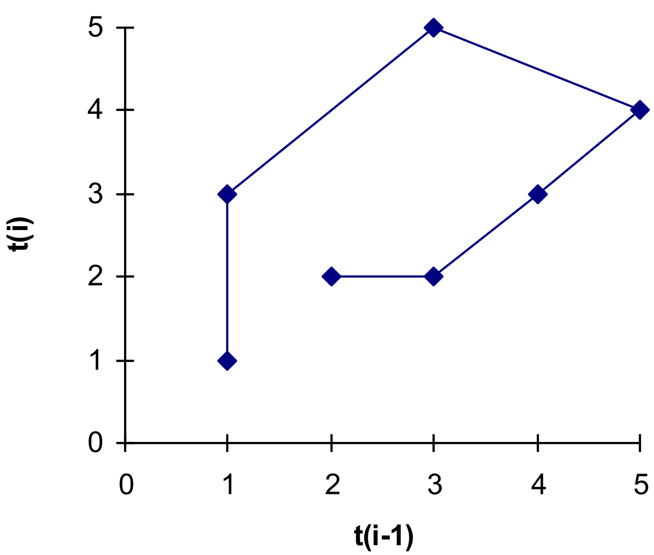
Example (Day 12)

Time-Delay Plots

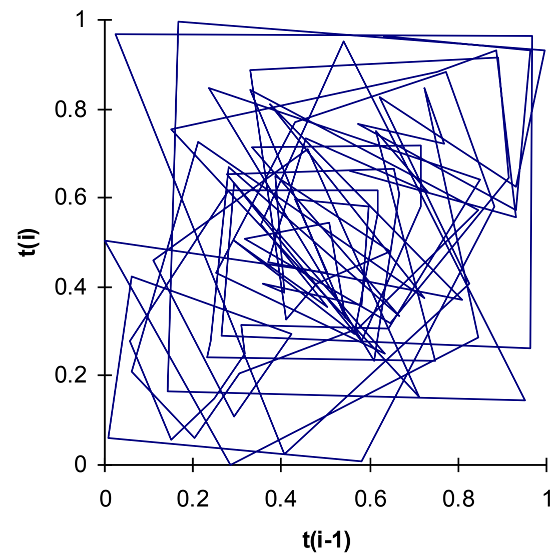
x=t(i-1)	y=t(i)
1	1
1	3
3	5
5	4
4	3
3	2
2	2

Takens' Theorem (1981): Such plots capture the complete topology of the system trajectory in the underlying (unknown) state space.

Key hunch: "nature writes straight with crooked lines." Dynamics that emerge from legitimate software used by many interacting machines should be qualitatively more complex than beaconing from a trojan.



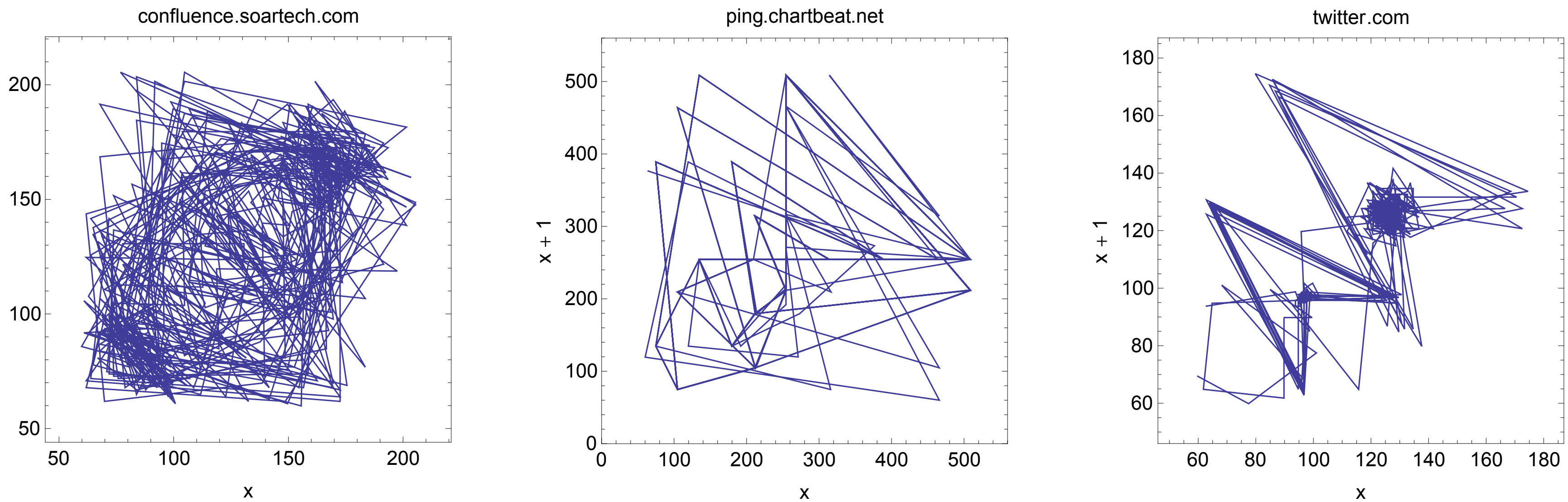
(100 Random Points)



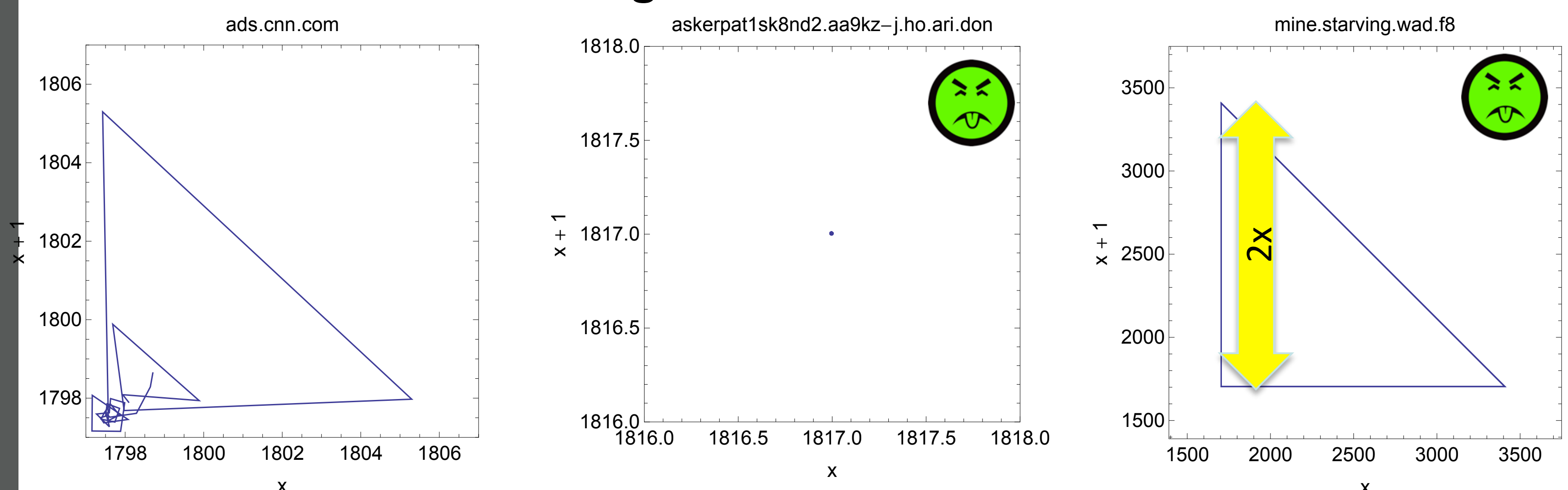
SOARTECH

Time-Delay Plots on Repetitive Edges

Known, Benign Examples



Regular Connections



Type A: perfectly regular

Type B: missed connection

Detecting Regular Beaconing

On a link with 10 or more connections to a rare URL (< 5 IPs)

- Compute successive differences x_i between connection times
- Subtract their mean
- Compute score:

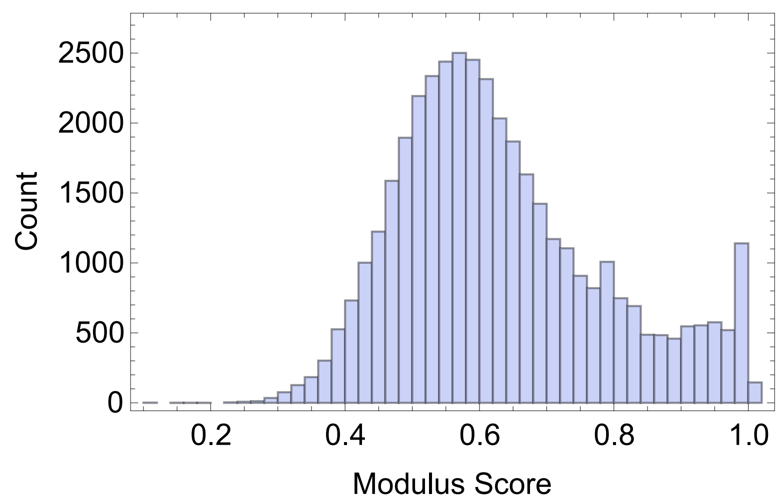
$$J' = 1 - \min_{\alpha} \frac{2}{N\alpha} \sum_{i=1}^{N} x_i \; smod \; \alpha$$

where

$$a \; smod \; b \; \equiv Min(a \bmod b, b - a \bmod b)$$

(Minimization over α can be focused around median of the series of differences)

- Select high scores (in our tallies, 1)



This score allows for

- Noise (small excursions around nominal period)
- Skipping (small integer multiples of the nominal period)

Performance

			Documented Attacks?							
			Yes				No			
			IP	Prede-cessors	Beacons	Suc-cessors	IP	Prede-cessors	Beacons	Suc-cessors
DUDE	Hit	Without Hints	15	10	11	2	19	1155[1]	33	536[2]
		With Hints	14	10			0	0		
	Miss		4	25			?			
Total			33	58			?			

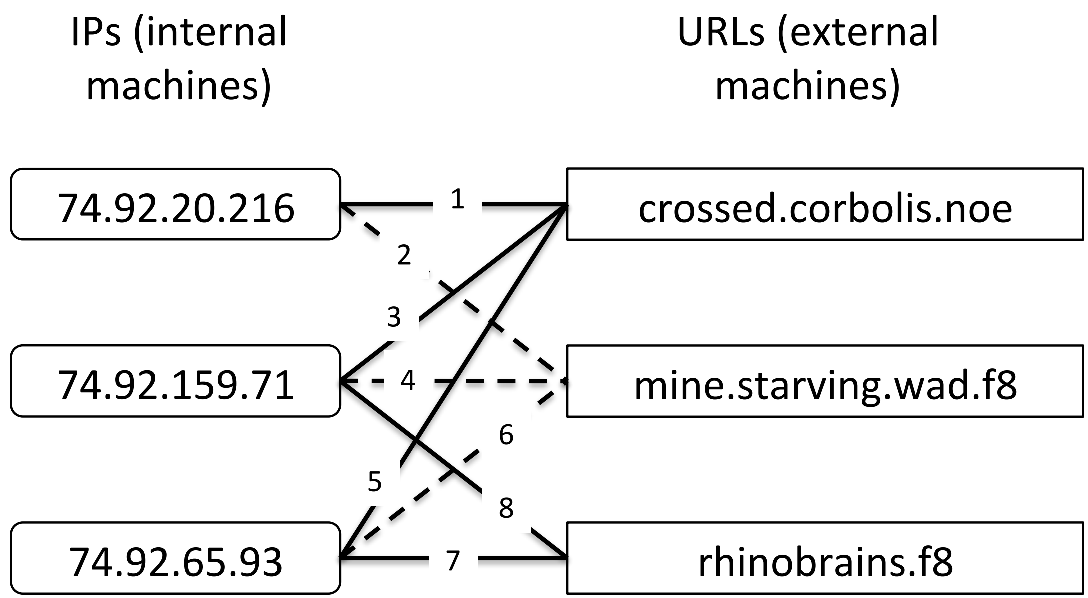
[1] These are associated with only 14 beacon links. # of predecessors/link = {**588, 278, 222**, 26, 21, 5, 2, 2, 2, 2, 2, 2, 2, 1}

[2] These are associated with only 9 beaconing links. # of successors/links = {**262, 218, 44**, 5, 2, 2, 1, 1, 1}.

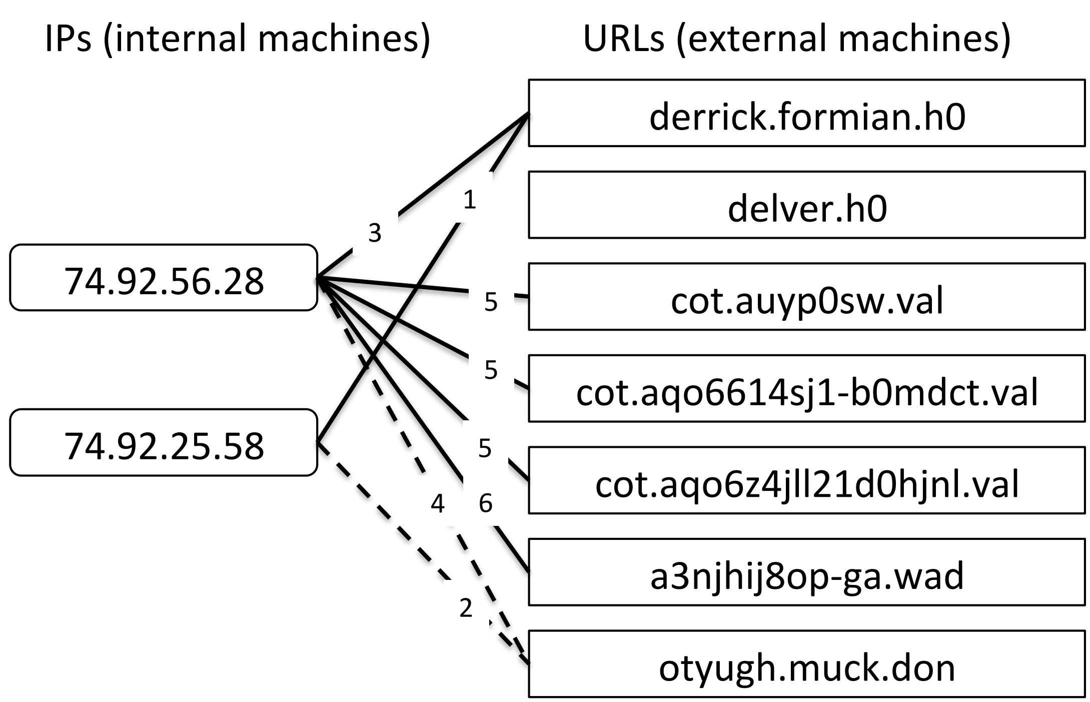
A single IP address is responsible for all of the bolded predecessors and successors.

Examples

Example (Day 11)



Example (Day 22)



SOARTECH

Next Steps

- Implement distributed processing of graph, to demonstrate scalability
- Multi-URL beaconing (at each beacon time, randomly selecting a different URL; requires looking for periodicity across multiple URLs linked to a single IP, which requires managing time as a first-class topology)
- Work on attacks that target DNS itself (e.g., cache poisoning, hijacking, DDoS amplification, …)
- Take into account time of day information
- Develop interactive diagnostics from the two detectors we currently have. E.g., use the beaconing detector to nominate suspect IPs, then use the shared URL detector to find other suspect rare URLs.

Questions and Discussion