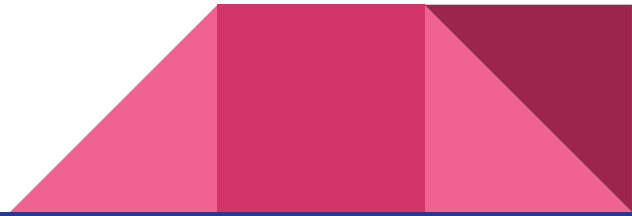
Data Driven Data Center Network Security

Speaker: Omid Madani
With contributions from: Vimal Jeyakumar, Ali ParandehGheibi, Navindra Yadav, and Cisco Tetration Analytics Team

Overview

- Motivation: Securing and managing a complex datacenter network
- Approach: Fine grained, flow-based and process-based monitoring & control
- A sampling of problems encountered, and findings
- Future directions



Motivation: Attacks and Defenses

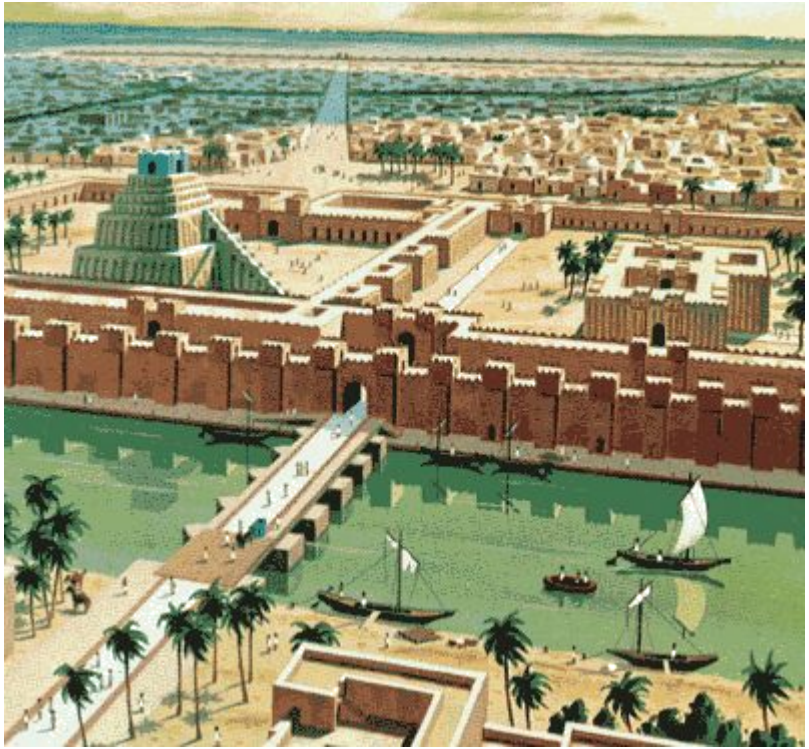
- Datacenters host a variety of applications, and *valuable data*
- Currently, little visibility into what's running: legacy apps
- From security-architect.blogspot.com:

> ***"Once inside an enterprise account, hackers*** *don't need domain administrator access. They can still use Active Directory to gather intelligence and* ***move laterally through the IT administrative hierarchy to access their target."***

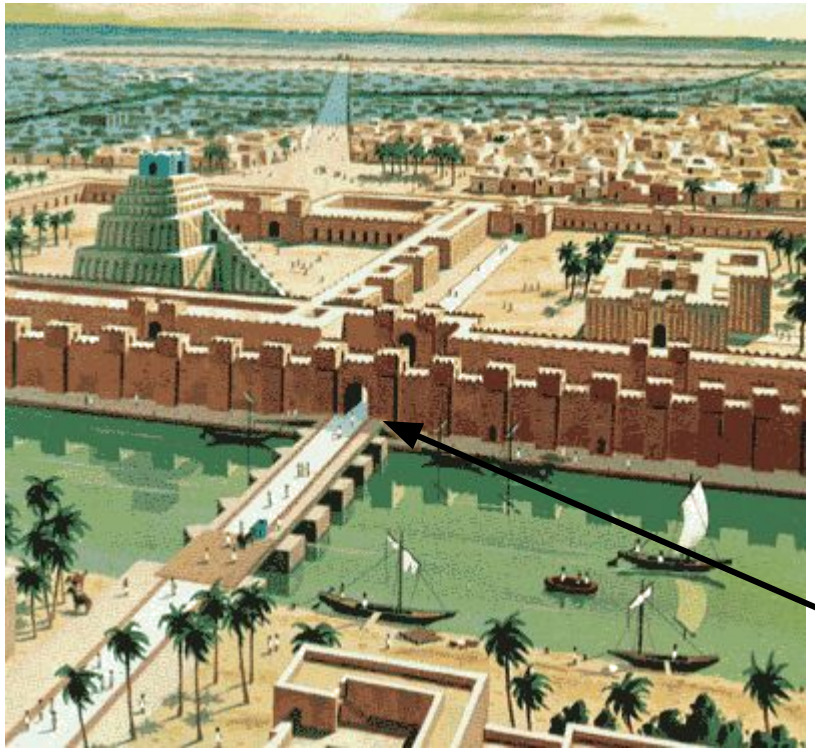
Rob Joyce (decades at NSA, theregister.co.uk, 2016):

> *"If you really want to protect your network you have to know your network, including all the devices and technology in it… In many cases we know networks better than the people who designed and run them."*

Walls/Barriers Not Enough!



Walls/Barriers Not Enough!

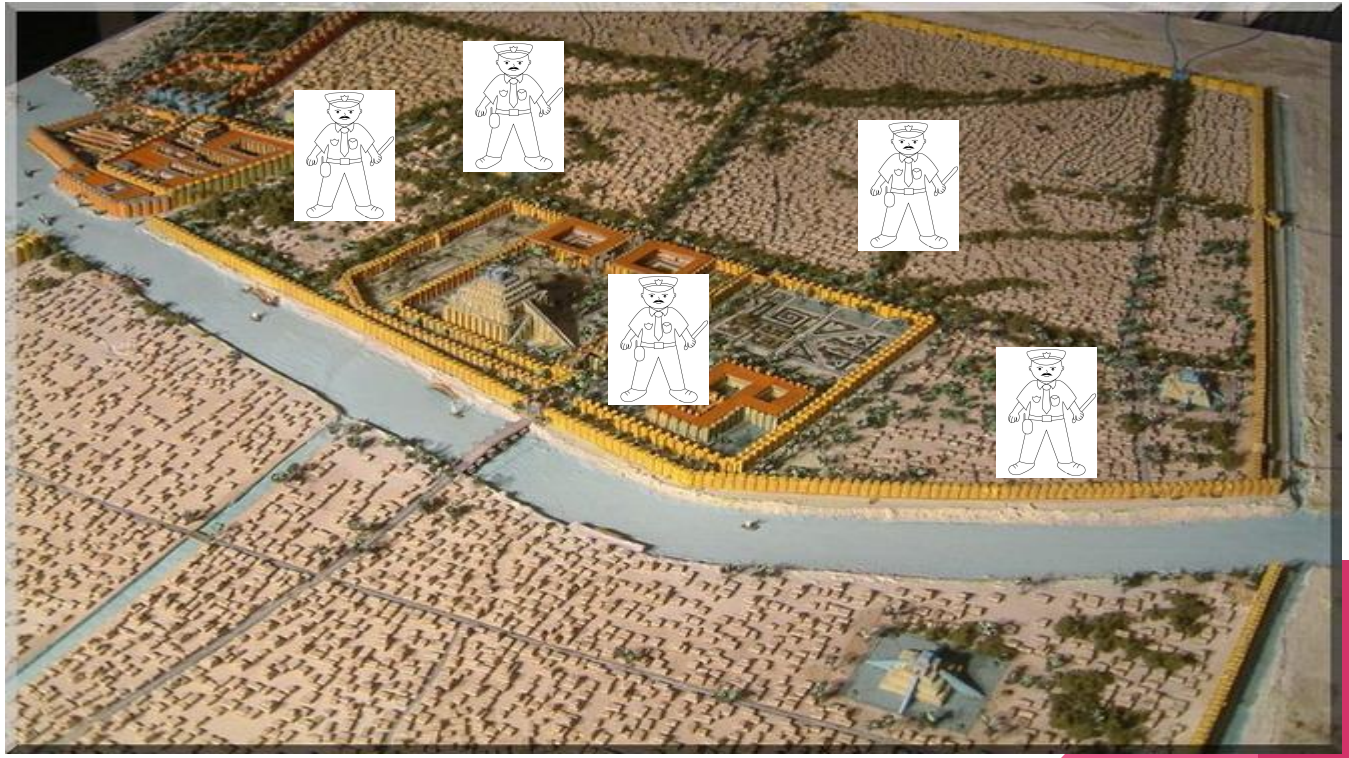


back door!

sides

front door!

We also Need Policing.. Inside!

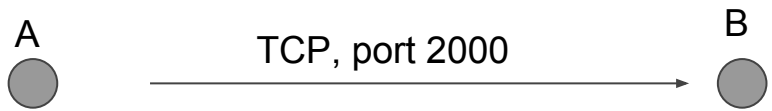


Overview

- Fine grained monitoring & control
- Approach:
  - Pervasive sensors
  - Data analytics: smart summarization, policy derivation and exploration
  - Other benefits: visibility & monitoring, performance
- The network setting, and a sampling of problems and findings
- Future directions

Whitelist Policy

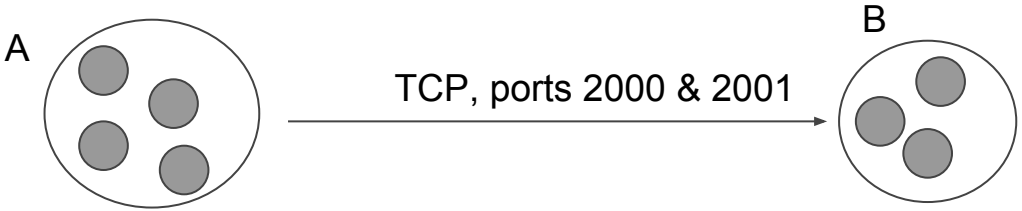
- Whitelist policies = Permitted communications
- Or: Who can talk to whom, on which port, protocol (TCP, UDP, ...)
  - No other communications allowed



- Host A can talk to host B using protocol TCP on port 2000.
  - Or, B serves A on port 2000 (there is an application/process on B that listen on 2000)

Whitelist Policy

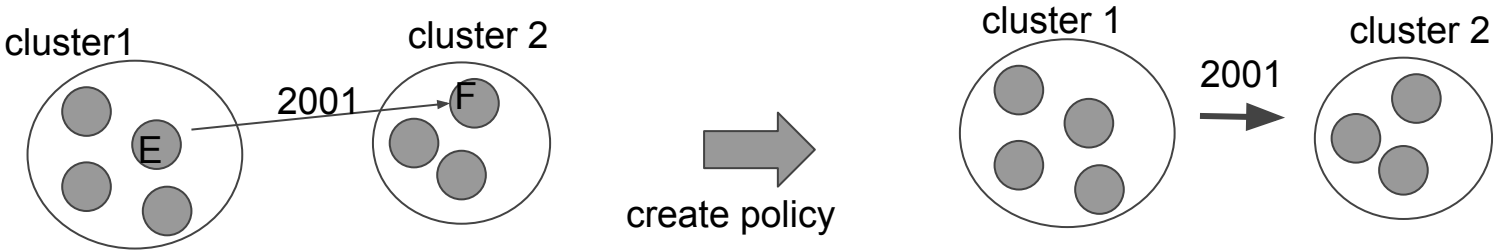
- Whitelist policies = Permitted communications
- Or: Who can talk to whom, on which port, protocol (TCP, UDP, ..)
  - No other communications allowed



The endpoint can be a group (cluster) of nodes/hosts

- Cluster A can talk to cluster B using protocol TCP on port 2000 and 2001
  - Or, B serves A on ports 2000 and 2001 (there is an application/process on B that listen on 2000)

From Clustering to Policies



The implication of clustering: If E is observed to be client of F on 2001, then all elements of cluster 1 are allowed to talk to cluster 2 on 2001 (even if that explicit communication wasn't observed among other pairs).

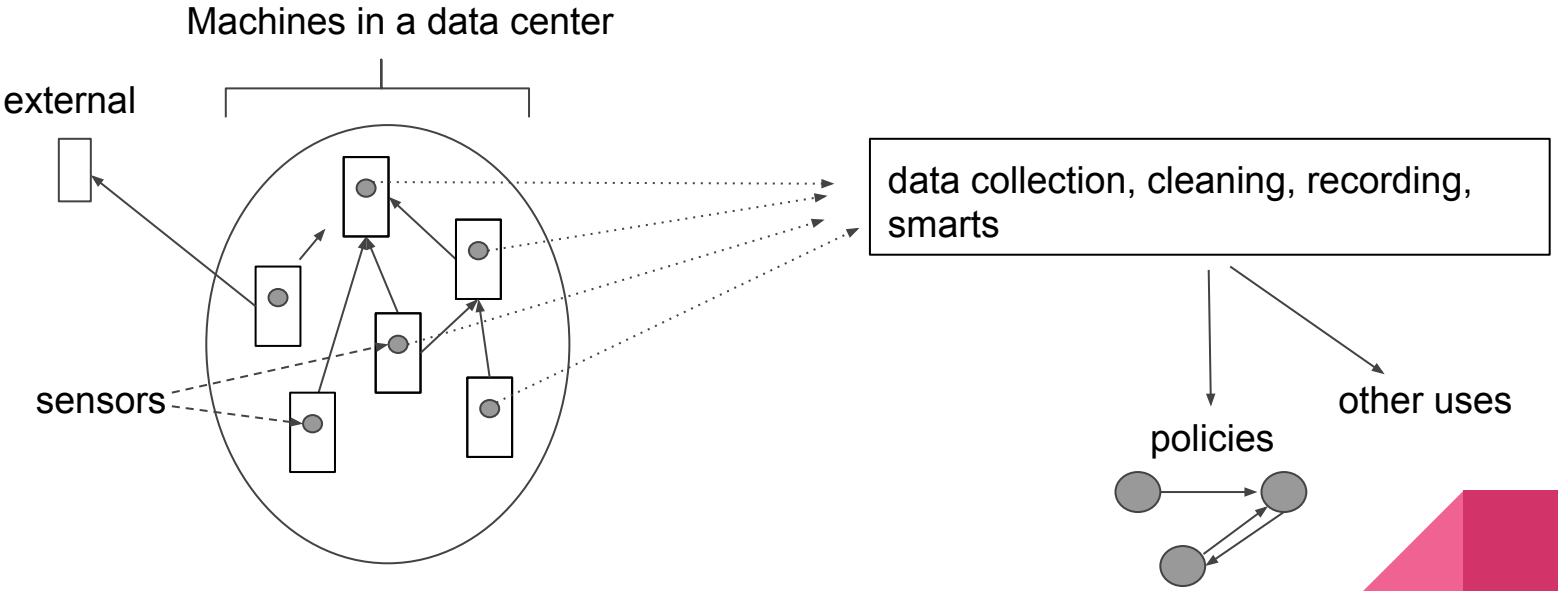
Inducing and Maintaining Policies

- Blacklist = What's *NOT* allowed
  - *Whitelists yield tighter control/security, but more expensive to create/maintain*

- Human knowledge is useful, but not sufficient:
  - Knowledge is distributed among multiple people, and noisy
  - Non-stationarity issues: New apps, updated apps, migrations
- Propose a complementary approach: With sensors and continuous data analysis, we aim to *render feasible the building and maintaining of whitelists*
  - Why not fully automated??
- Other benefits of smart sensing: visibility, anomaly/threat detection, troubleshooting, etc.

Big Picture

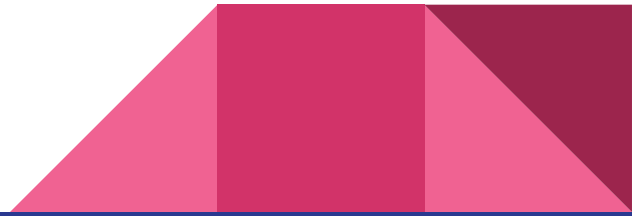
- Sensors everywhere, with smart/efficient data gathering and analytics



Data gathered: who talks to whom (network edges/flows), processes, ...
the host ‘attributes’ needed for computing pairwise similarity & clustering

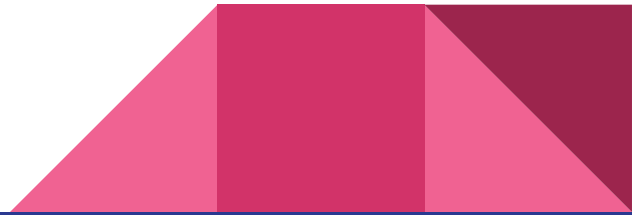
Input & Output for Clustering/Policy Induction

- Repeat:
  - User selects segments of their network, time period
  - Data analysis of the communications yields clusterings and policies
    - Data analysis = feature encoding, vector construction, clustering
  - Explore & make changes, rerun algorithms
  - Experiment/Enforce policy: observe violations/drops, make changes



Aspects of the Clustering Problem

- Model hosts as (text) documents (bag of words)
  - many similarities: frequent connections and distinctive connections
- But *structured* features:
  - ports & protocols, directions (client-server)
  - process information
  - multiple views?
- A graph clustering problem
- Side information: some partitioning via subnets/routes, naming cues, IP structure.
- User feedback, so '*Interactive Clustering*'?
- Evaluation challenges



Hosts as Documents

feature encoding & vector construction

[four: 1, score: 2, …]

pairwise similarity, clustering

Hosts as Documents

feature encoding & vector construction

[four: 1, score: 2, …]

pairwise similarity, clustering

A host

client of 137.3.3.3 on 4100, server to 10.1.1.1 on 6000, …..

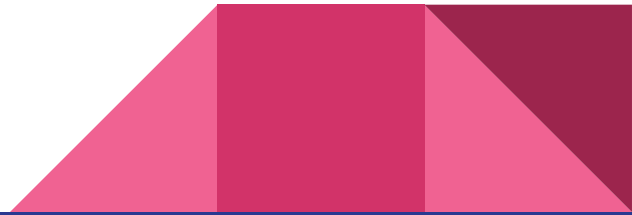
feature encoding & vector construction

[137.3.3.3 on 4100, 10.1.1.1 on 6000, …]

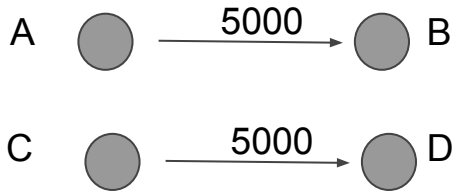
pairwise similarity, clustering

Aspects of the Clustering Problem

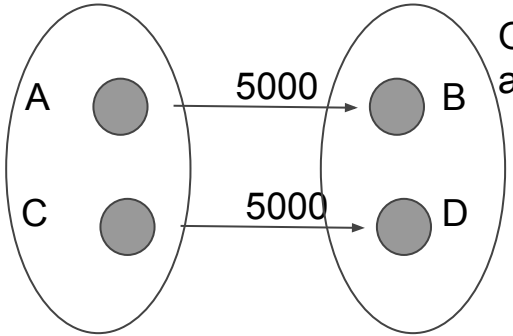
- Model hosts as documents
  - many similarities: frequent connections and distinctive connections
- But *structured* features:
  - ports & protocols, directions (client-server)
  - process information
  - multiple views?
- A *graph* clustering problem
- Side information: some partitioning via subnets/routes, naming cues, IP structure, ..
- User feedback, so '*Interactive Clustering*'?
- Evaluation challenges



Similarity



Similarity



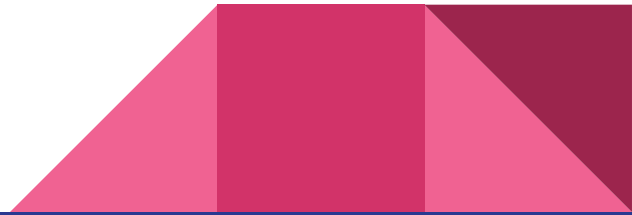
One cluster (or B and are equivalent)

Findings

- ‘Hosts as documents’ is promising
  - Encode port, destination, etc into one string!
  - tfidf better than Jaccard!
- Explored a number of clustering techniques
  - Extended kmeans++ and affinity propagation among best
- Choice of k (number of clusters) is another challenge
  - MDL and Silhouette methods to determine a good k
  - User can influence that choice
- Accuracy tests on several different datasets, with partial groundtruth
- Performance has ranged in the 70s to 90s (precision and recall)
- Sources of inaccuracy/challenges

A few other (sub)Problems

- Client-Server determination
- Port-interval detection & estimation (variant of the *German tank* problem!)
- Determining legit traffic!
- Autonaming clusters
- Feature extraction from process info
- How best to visualize/present? (UI problems)
- Policy tightness/simplicity/accuracy tradeoffs
- ….



Future directions

- Overall challenge/goal: reduce human workload
  - Keep it engaging
- Gather customer feedback
  - Assess impact on network security
- Analytics/ML for related problems
  - Attack detection/classification
  - Normalcy modeling
  - Performance troubleshooting