RSA CONFERENCE CHINA 2011
NOVEMBER 2-3 | CHINA WORLD HOTEL | BEIJING

Big Data Techniques for Faster Critical Incident Response Handling

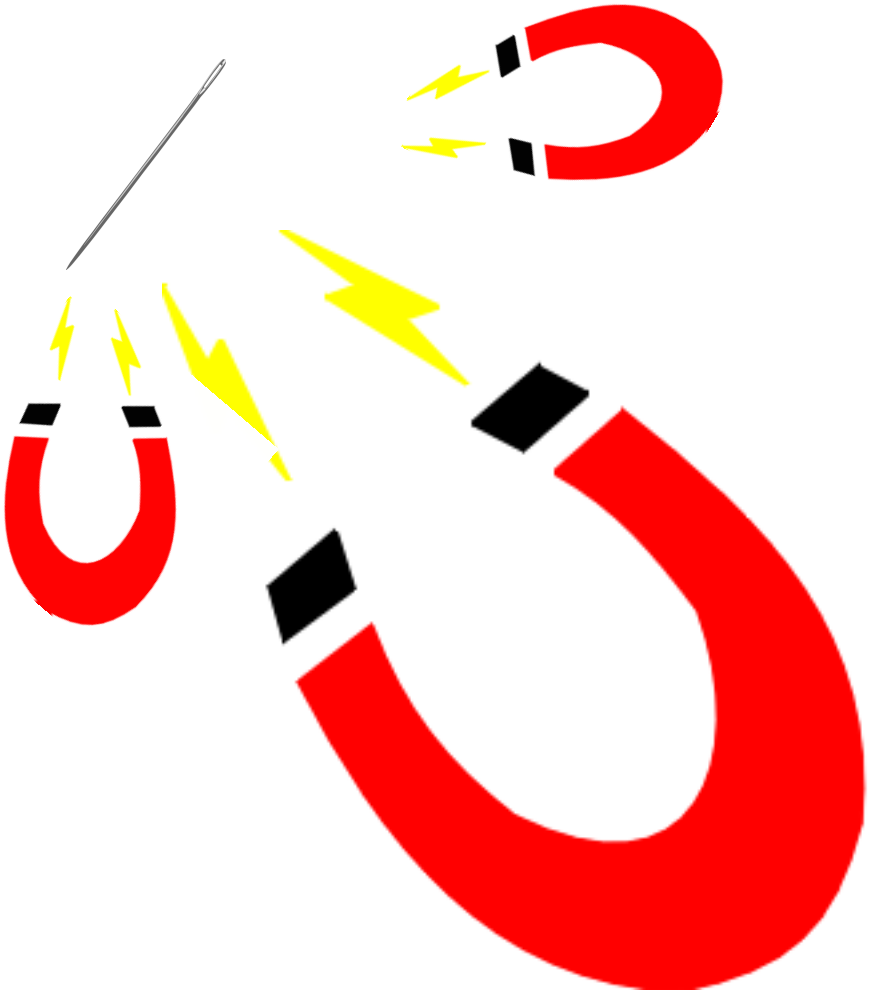
Speaker : Samir Saklikar, Dennis Moreau

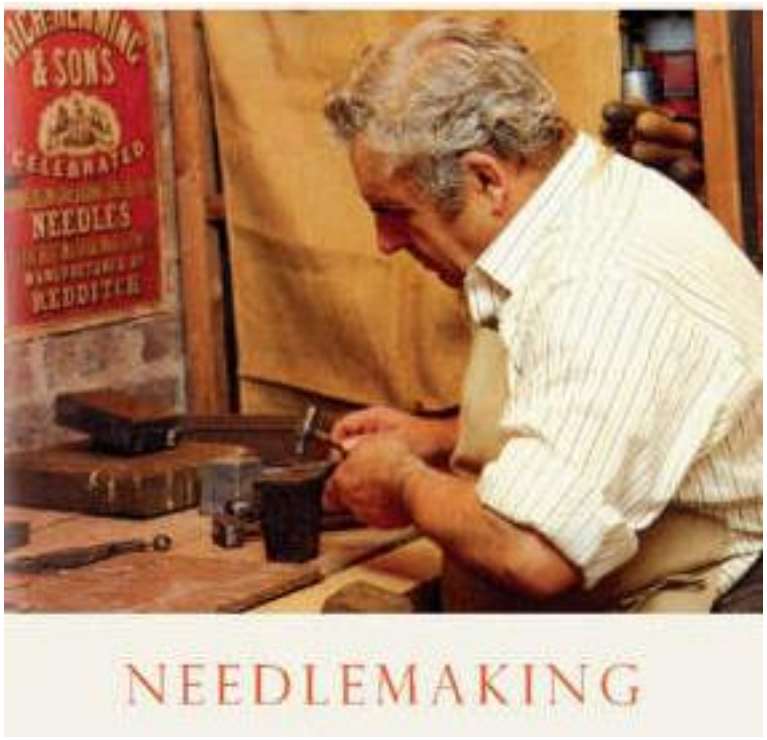
Job Title : Principal Technologist, Senior Technologist

Company Name : RSA, The Security Division of EMC

Data Analytics – Finding Needles in Haystacks

… Some solutions work …





But what if ?

Find  in 

That “blade” of grass – of Height H, Width W, Green-ness G, which grew in Field F and was cut at time T..

And yes… That blade of grass – withers differently over time…from the rest of the stack…
So, at a given time, the blade appears “identical” to the rest of the stack, but differs in “behavior” over time.

Now, That’s a Challenge!

Critical Incident Response

- Organized Approach to addressing and managing the aftermath of a security breach or attack
  - Detection & Correlation (AT, 0-Day Scenario)
  - Limit Damage
  - Reduce Recovery time
  - Learn (to) and Adapt (from) the attack
  - Protect but Monitor
  - Collect Data to Litigate

A Typical Incident Response system

- From ‘Evolution of Incident Response’ Blackhat 2004
  - Pre-Incident Preparation
  - **Detection of Incidents**
  - **Initial Response**
  - Formulate Response Strategy
  - Data Collection
  - Data Analysis
  - Reporting

Upfront Challenges

- Infection point may be way in the past
  - Missing Logs, Newer Configuration..
- Distributed in time
  - Gap of days or even weeks between attack steps
- Distributed in space
  - Use different endpoints for different steps
- Identifying a single step in attack is not enough
  - Need the Attack vector to identify the next step

Hidden Challenge – Evolving IT Landscape

- More Assets in your Infrastructure to be managed
  - Move towards cloud; large interdependent assets
- More layers in the Technology stack to monitor
  - Virtualization (Endpoint/Server virtualization), Dynamic Hosting
  - Mobile Clients
  - More Layers → More Logs
- More Detailed Context required
  - Network Topology, Resource Pools, Asset/Service/Data Classification, Service Stack Provisioning/Provenance …
- More Security Data Sources
  - Netflow, FPC, Sandbox Indicators, Appliances, Mobile
- Less Visibility – Increased Informational Complexity
  - Hidden Dependencies on Resource Coupling (Chipset, BIOS, Cache, DMA Controllers, Firmware, Interfaces (ACPI), …)

Incident Response – The Tools at hand

- Intrusion Detection
  - Host and Network-based tools
- SIEM/Packet Capture Tools
- Vulnerability Scanners
- Memory Analysis
  - WinDD, MDD, Volatility
- Disk Analysis

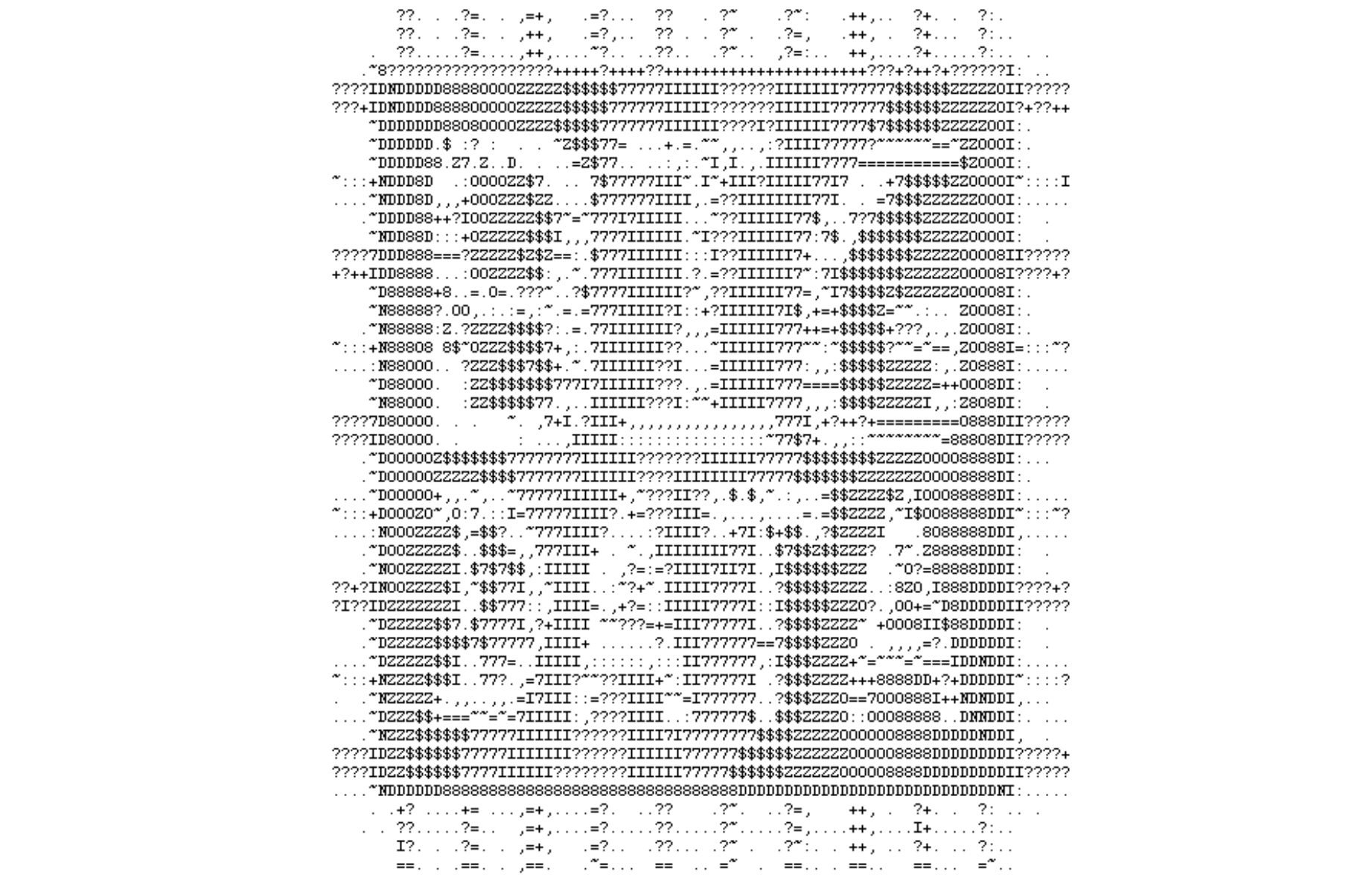
But the Data!

- Massive Aggregations= Logs for all endpoints in enterprise/cloud
  - Logs across all stacks (HW, VMM, OS, APP, Service, …)
- Loosely Structured = Log formats (developer-defined strings), Packet Captures
- Distributed = Multiple sensors
- Multiple Log consumers => multiple analytics and representations (Analysts, Auditors, Ops Problem Resolution, Optimization …)

And some more Data..

- Asset Configuration/Vulnerability Scores/Criticality
- Regulatory Controls/Security Architecture
- Topology
- Attacks
- Threats
- Disk/Memory Images
- Provisioning Provenance
- Hosting Stack Configuration
- …

Incident Response – The Challenge!

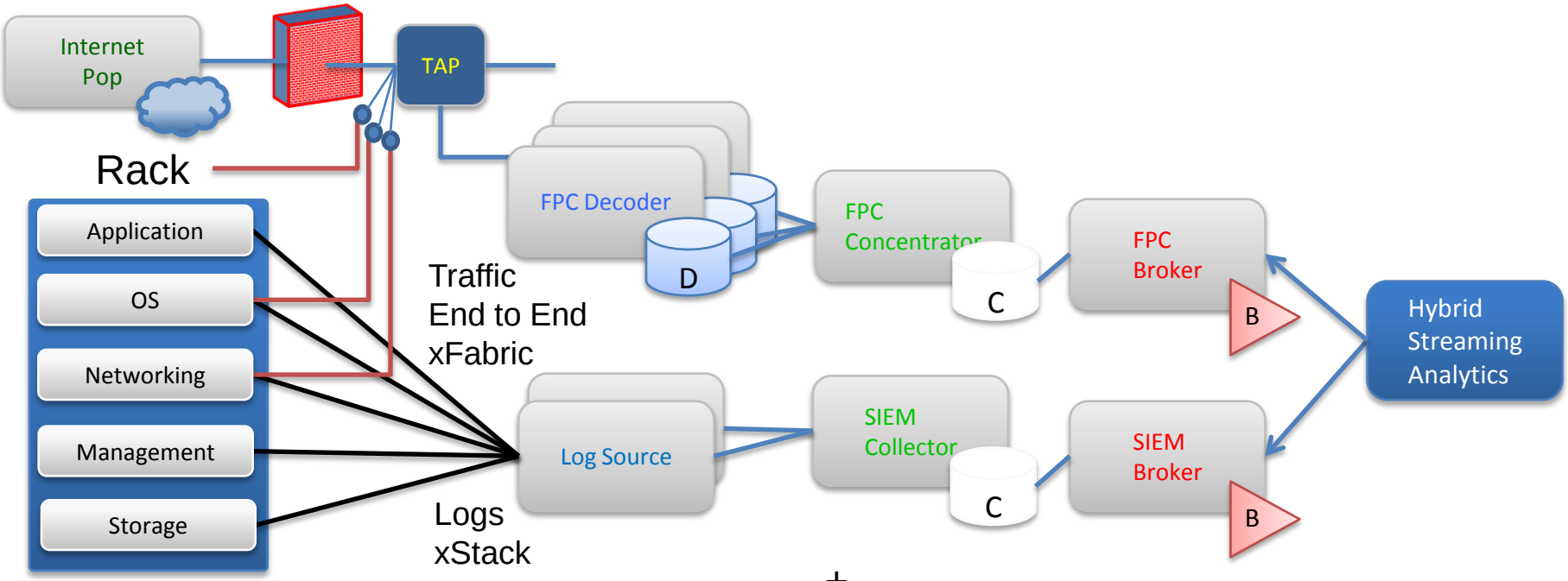


Starting easy – A definition

451 Group

- Big Data is a term applied to Data sets
  - That are Large, Complex and Dynamic, with a need to
  - Capture, Manage and Process the Data Set in its entirety,
  - Giving Results in Tolerable Time Frames, such that
  - Existing & Traditional Software Tools and Analytic techniques fall short
- Seems Made-to-Order for Incident Response!

Why Big Data for Incident Response?



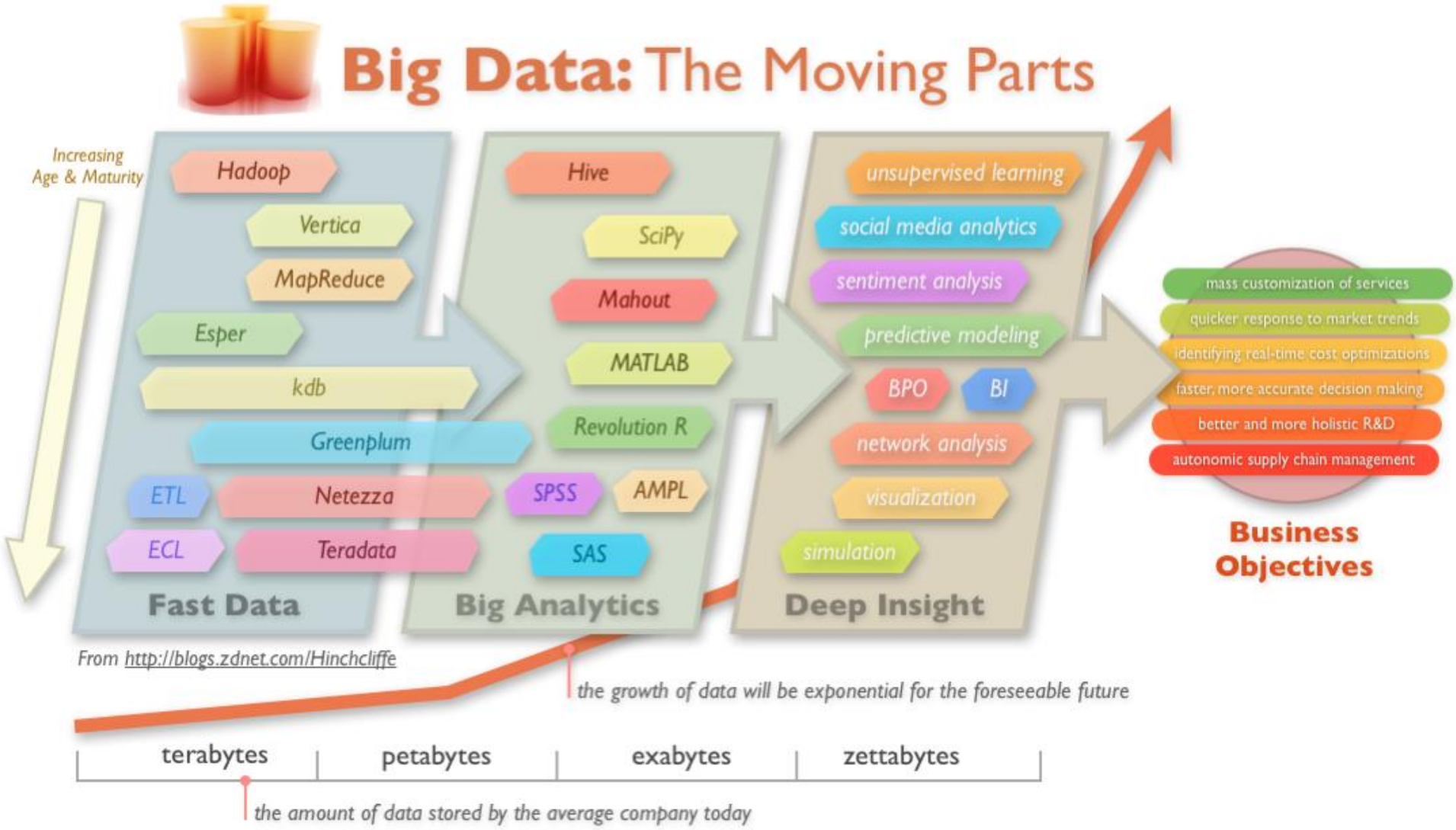
+

Net-flow Behavioral Analysis
Automated Sandbox Execution Behavior learning

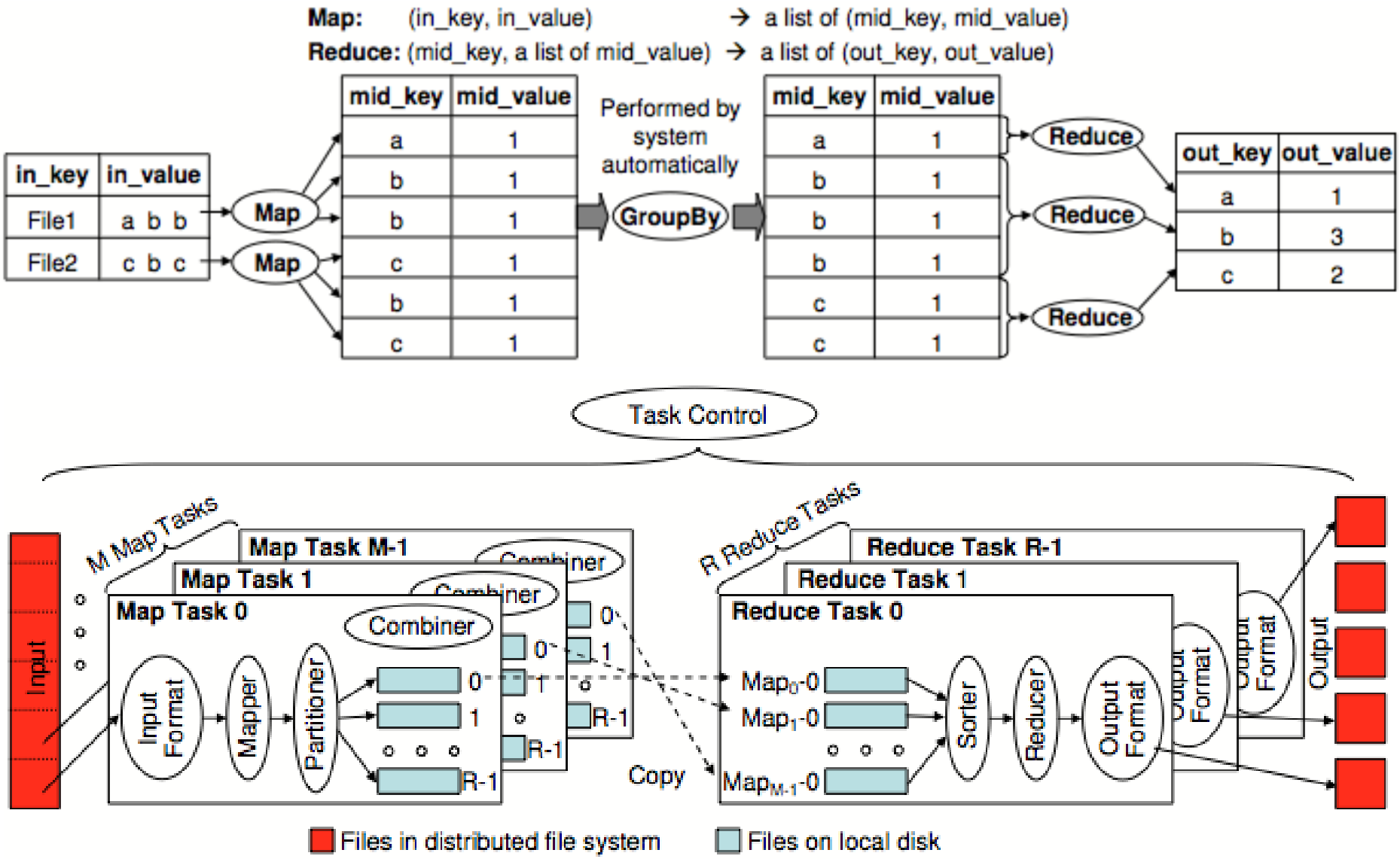
+

Asset Configuration
External Information - CVEs

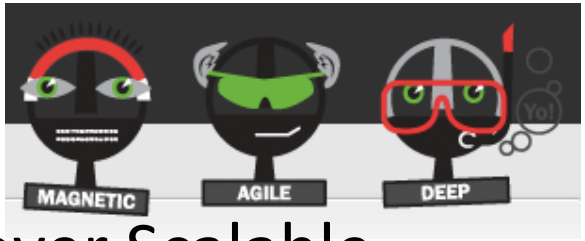
Big Data Technologies



Map Reduce (A Quick Primer)



Scalable Machine Learning over Big Data



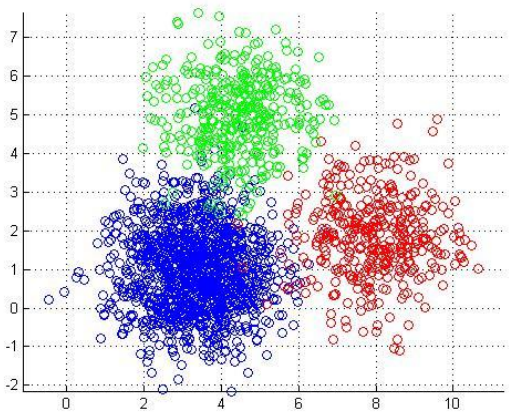
- OOTB support for Machine Learning over Scalable, Highly Parallel Infrastructure
- Clustering
  - Canopy + K-Means Clustering
  - Fuzzy K-Means Clustering
- Classification
  - Logistic Regression
  - Bayesian
  - Support Vector Machines
  - Neural Networks

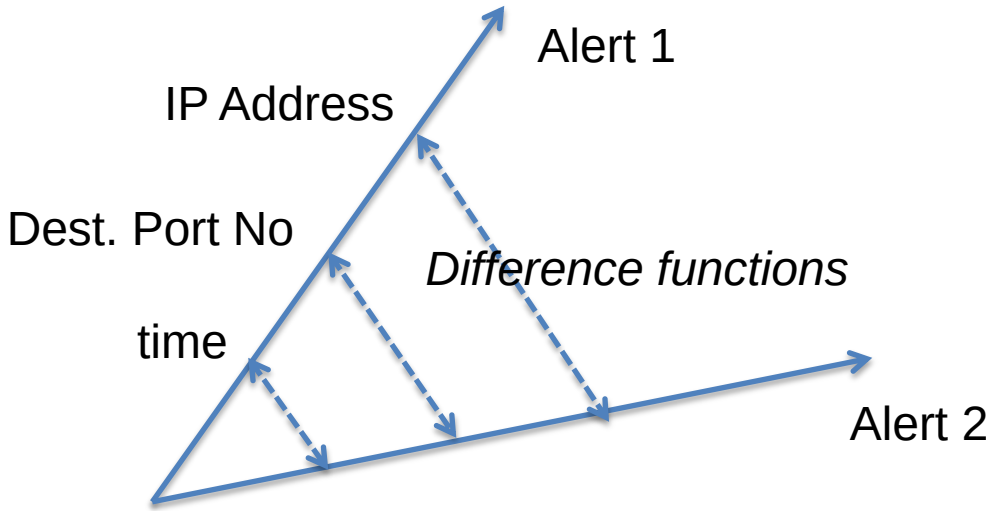
Big Data Processing of Logs

- Various Industry and Academia examples
  - **CoHadoop: Flexible Data Placement and Its Exploitation in Hadoop** *(Proc. VLDB 2011)* - Empirically and analytically demonstrates scalable log processing
  - **In-situ MapReduce for Log Processing** *(Usenix ATC 2011)* - Demonstrates log processing adaptively balancing latency and fidelity.
  - **Parallel Data Mining Platform in Telecom Industry** – Demonstrates production use of Hadoop for Telecom data mining, including log processing.
  - **Taming Orbitz Logs with Hadoop** – Demonstrates production log processing using Hadoop.

IR Challenge # 1 – Large No. of Alerts at any given time

- Problem
  - Large number of Alerts >>>> CIRT Capacity
  - Not all Alerts are ever handled by the CIRT team.
- K-Means Clustering (Mahout over Hadoop Or MadLib over SQL)
  - Alerts represented as n-dimensional vectors
  - Needs Difference Function
- *Clustering Intrusion Detection Alarms to support root cause analysis – K. Julisch*
- *Intrusion detection with unlabeled data – L Portnoy*





IR Challenge # 2 – Large No. of Similar Alerts seen over time

- Problem
  - Large number of Alerts >>>> CIRT Capacity
  - Similar Alerts seen over time. Needs re-disposition from CIRT Team member

- Decision Tree Classification (Mahout over Hadoop)
  - Supervised Learning Mechanism based on Alert Classification by CIRT
  - A tree-structure with '*decision nodes*' containing test attributes, *branches* as possible attribute values, and *leaf nodes* as classification answers

- Naïve Bayes Classification
  - Supervised Learning Mechanism
  - Stochastic Model wherein Input 'Independent' Variables contribute 'Independently' towards probability of a data belonging to class C
  - 'Independent' – makes it manageable;

Code for Naïve Bayes (Using MadLib over PostgreSQL or Greenplum)

```
sql> SELECT madlib.create_nb_prepared_data_tables(
'training-table','class-col','attributes-col',num-attr,
'nb_feature_probs', 'nb_class_priors');
```

```
sql> SELECT madlib.create_nb_classify_view (
'nb_feature_probs', 'nb_class_priors','to-classify-
table', 'id-col', 'attributes-col', num-attr, 'output-
class-table');
```

Out of the Box parallelization across multiple appliances

IR Challenge # 3 – Low Frequency, Low Visibility Small Number of Alerts

- Problem
  - Low Frequency, Low Visibility Alerts (always lesser than the CIRT Radar threshold)
  - Characteristic of APT Behavior – Low and Slow, Distributed in time/space
- Solution
  - Alert Scavenging! Parallel Execution for High Seed trial-and-error searches

- Multi-Level Clustering?
  - 1st Pass on IP Address & varying time-intervals
    - Identify cluster of Alerts, related to same IP
    - Identify Multiple sets of Clusters over different time-intervals over the data-set
  - 2nd Pass on Alert Types
    - Needs Input on "closeness" of Alert Types, based on which Alert types may follow another
  - Desired Output
    - A cluster of Alerts, related to same IP address within certain time-intervals, wherein Alerts seem to convey a pattern

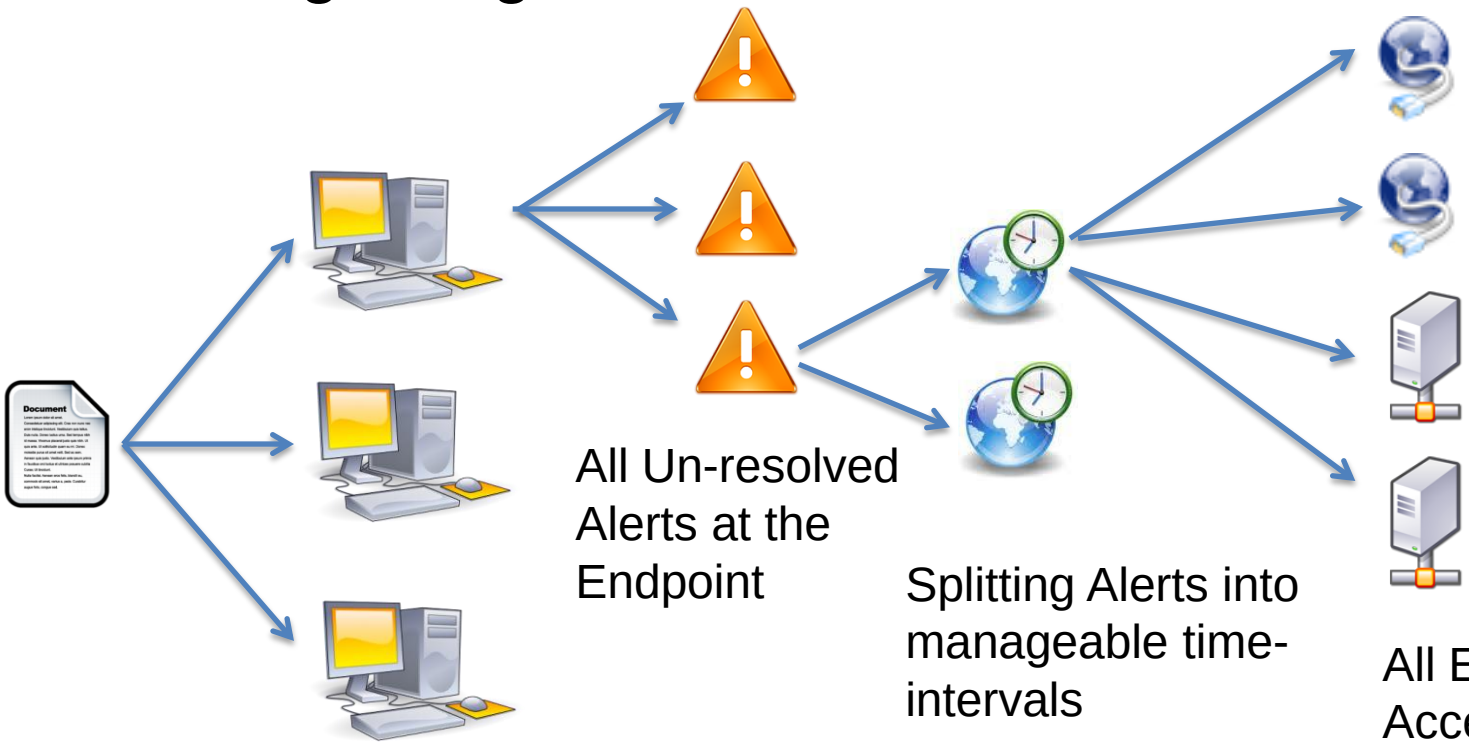
IR Challenge # 4 – Initial Response. Going from Knowledge to Source of Infection

- Problem
  - Going from
    - Information about a possible incident
      - a leaked document found on underground sites
      - strong evidence of a compromise
  - To Source of Infection
    - A small set of potential suspected sources of infection on listed endpoints
- Challenges?
  - Large number of Endpoints
  - Larger Number of Alerts, Logs, Packet Capture
  - Needs to deal with past Data
  - …

Proposal – Big Data for Initial Response

- An Iterative Feedback-based Diverge-and-Converge Approach for Identifying Source of Infection
- Diverge
  - Go from knowledge of leaked document to increasingly bigger sets of Endpoints, Alerts and Server Access Logs
- Converge
  - Cluster within Server Access Logs, Correlate across Alerts and Endpoints into a small set
- Feedback-based
  - Get feedback from Human CIRT Operator for suspected set of Infection points
- Iterative
  - Rinse-and-Repeat using Parallel Execution for speedy responses

The Diverge Stage



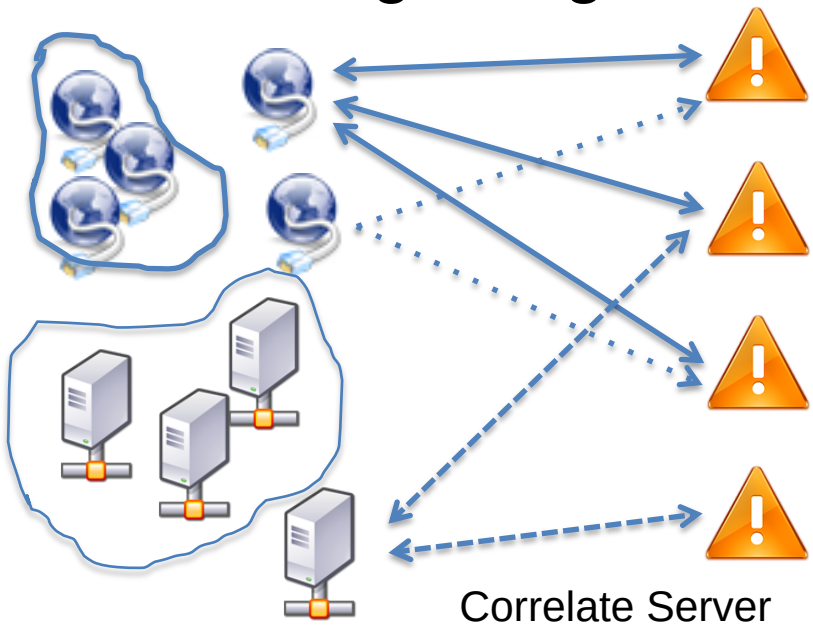
All Endpoints which accessed the document

All Un-resolved Alerts at the Endpoint

Splitting Alerts into manageable time-intervals

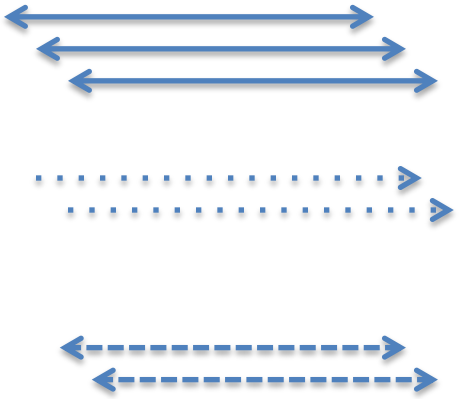
All External Accesses from that endpoint in that time-interval. To both internal and external servers

The Converge Stage



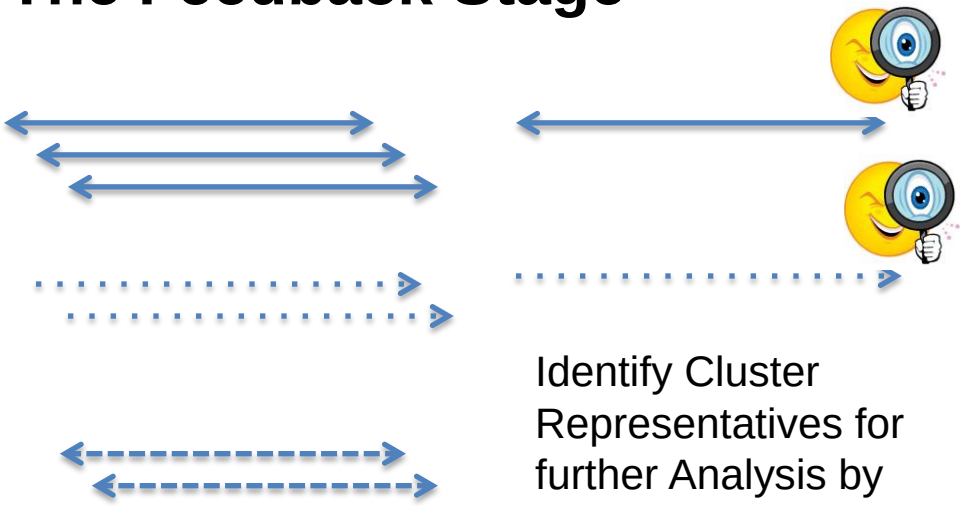
Cluster Server Accesses to remove well-known dominant Servers. Focus on the outliers of the clusters

Correlate Server Accesses with Alerts, to identify patterns of Access between them (one followed by other etc)



Cluster across Server-Access-Alert Correlations to find dominant type of such activities across Endpoints.
Select Cluster Representative for Human Analysis

The Feedback Stage



Identify Cluster Representatives for further Analysis by CIRT Team

CIRT team analyzes the short-listed Alerts, and gives feedback, which is applied to the entire cluster.

If high number of Alerts, then feedback is used to tighten the Converge Stage

If low number of Alerts or bad results, feedback is used to widen the Diverge Stage

The Iterative Phase

Pros and Cons

Pros

- Generalized Algorithm
  - No details of specific incident required
  - Can be tweaked
- Brute Force approach
  - Can be exhaustive, instead of specific search criteria
- Data-driven
- Fast
  - Takes advantage of Parallel hardware
- Can be checked for convergence

Cons

- Not “Intelligent”
  - May waste CPU cycles

Futures

- Need to bring Intention into the Picture
  - Security Architecture Models - What did I intend?
  - Impact: GRC –> Service Catalog -> Hosting Stack Provisioning -> Logs -> Forensics – What's important? (triage)
  - Operational Constraint/Policy Boundaries – What are plausible response options? (next desired state, re-provisioning, resilience, …)
- Need to Leverage Standards
  - Semantic: As glue across larger information bases
  - Community: Dissemination/consumption feeds
  - Automation: Integration of tools/data/processes

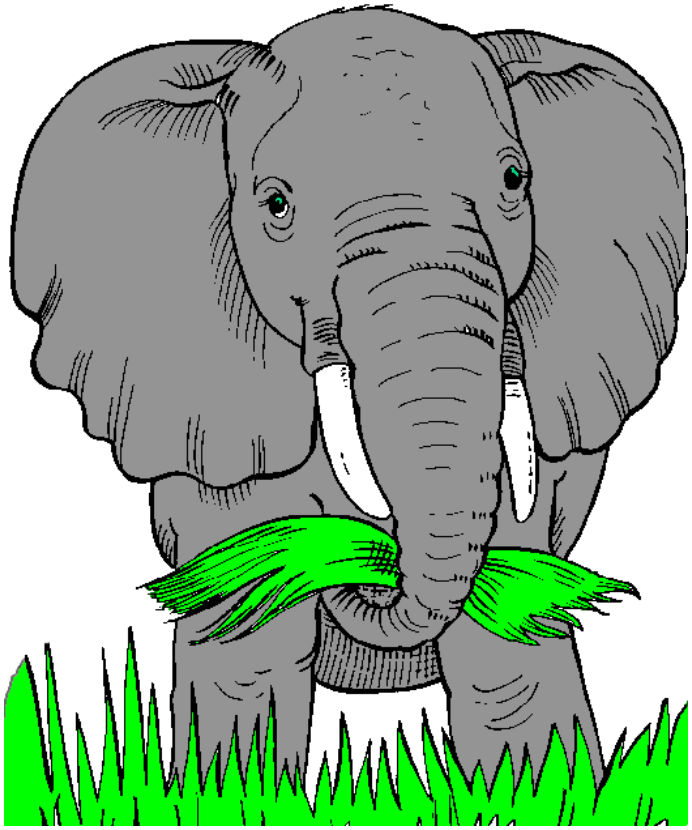
Futures

- Need to exploit “community” effects:
  - Peer Benchmarking, Peer Cooperation, Shared Intelligence
- Need to deliver Situation Awareness across Hosting Relationships
  - Iaas, Paas, Saas, Cross Cloud
- Need to Accommodate “Security as a Service”
  - Analyst Collaboration
  - Vendor Guidance
  - IR Augmentation as a Service

Conclusion

- Big Data seems *made-to-order* for Security Analysis
  - Incident Response is the Killer App
- Big Data-based Machine Learning for Security
  - Moving from Academia to Industry setting
  - Accessible set of OS and Commercial tools
- Big Data for Security – Scope for Innovations
  - Security specific languages, tools
  - OS Frameworks built using existing tools
- Big Data for Security.. too Big
  - to be ignored by vendors and practitioners

How to get to the "needle"?



Guess what? Elephants can eat Grass!

So, Eat your way to the Needle!

Thank You!

Samir Saklikar
(samir.saklikar@rsa.com)

Dennis Moreau
(dennis.moreau@rsa.com)

Backup

Limitations of InDB Scale-Out Processing

- **SAS High Performance Computing: The Future is Not What it Used to Be** – Documents SAS's experience and documented limitations of shared nothing scale out processing. Description of the necessary extensions for addressing projection and optimization analytics are also identified.

Futures

- One Size does not fit all … even if it is “Big”
  - Optimization
  - Projection
  - Deep Dependency (data coupling)
- Emerging Complexity implies need for Simulation to inform plausible response (actionability)
- “Big Data” plumbing $\neq$ “Retention” Plumbing $\neq$ “Simulation” plumbing $\Rightarrow$ Need for Information Coherence (across analytic/forensic bases)
  - Root Cause Analysis
  - Drill Through