Atypical Behavior Identification in Large Scale Network Traffic

Daniel Best {daniel.best@pnnl.gov}
Pacific Northwest National Laboratory

Ryan Hafen, Bryan Olsen, William Pike
Agenda

- Background
- Behavioral algorithm
- Scalable data intensive architectures
- Visualization
- Future directions
**What is large scale network traffic?**

- Most enterprises use some kind of **continuous traffic monitoring**.
  - Captured in either pcap or network flow format
- Network flow is a summarization of network communication
- Network flow is **ubiquitous and voluminous**
  - Groups of computers can easily have thousands of flow records per second
  - Large enterprises generate billions to tens of billions of flow records per day

```
```
Development goals

- Provide situation awareness and event discovery in large data sets
- Facilitate behavioral modeling and anomaly visualization for streaming network traffic
- Be capable of real-time and exploratory mode of investigation
How to find atypical behavior?

► Application concepts paying attention to three areas
  ■ **Algorithm**: Must be efficient to cope with volume of data
  ■ **Data Management**: Must be able supply data quickly
  ■ **Visualization**: Must provide the user the ability to discern atypical behavior and begin investigation process

► Meeting our goals
  ■ Operationally demonstrated on a dataset containing 100B flow records
  ■ Demonstrated capability to stream network flows at ~3 thousand flows per second on a single desktop computer
Atypical behavior algorithm background

- Behavioral model based on temporal patterns
  - Improvement over previous models (SAX: Symbolic Aggregate approxXimation)
- Operates under the assumption that network flow attributes exhibit cyclical behavior of a weekly periodicity
  - Exploration has shown this holds well for most protocols
- Various attributes can be modeled
  - Total bytes, total packets, network flow count
- Aggregation is necessary for statistical robustness
Weekly periodicity

Take median to form baseline
Comparing current activity to historical trends

- Running median calculated for single current series and for m number of historic series

\[ c_t = \text{median} \left( x_j : j \in \left( t - \frac{k-1}{2}, \ldots, t + \frac{k-1}{2} \right) \right) \]
\[ h_t = \text{median} \left( x_{j}^{(i)} : i \in (1, \ldots, m), j \in \left( t - \frac{k-1}{2}, \ldots, t + \frac{k-1}{2} \right) \right) \]

- Median absolute deviation (MAD) calculated based on current and historic running medians

\[ \hat{\sigma}_c = K \text{median}(|x_t - c_t| : t \in (1, \ldots, n)) \]
\[ \hat{\sigma}_h = K \text{median}(|x_{t}^{(i)} - h_t| : i \in (1, \ldots, m), t \in (1, \ldots, n)) \]

- MAD and a configurable deviation number used to set upper and lower bounds for current and historic series

\[ c_t^{\text{lower}} = c_t - \alpha \hat{\sigma}_c \]
\[ c_t^{\text{upper}} = c_t + \alpha \hat{\sigma}_c \]
\[ h_t^{\text{lower}} = h_t - \alpha \hat{\sigma}_h \]
\[ h_t^{\text{upper}} = h_t + \alpha \hat{\sigma}_h \]
Current and historic trend overlap
Visually encoding overlap with saturation

\[ \lambda_t = \begin{cases} 
\max(0, c_t^{upper} - h_t^{lower}) & h_t > c_t \\
\max(0, h_t^{upper} - c_t^{lower}) & h_t \leq c_t 
\end{cases} \]

\[ s_t = \begin{cases} 
0 & h_t = c_t \\
1 - \min(1, \lambda_t / |\delta_t|) & h_t \neq c_t 
\end{cases} \]

Saturation used to color encode the background of plots.
Scalable data intensive architectures

- Client visualization with various database back-ends
  - Postgres, Greenplum, Netezza
  - Needs database driver and appropriate configuration files
- Scalability through aggregation
  - Using summary table (not required), improves performance
- Network traffic grouped into categories
  - Rule based categorization algorithm
  - Based on attributes available in the data
    - port, protocol, payload, etc.
Primary data architecture focus

- Development and research on Netezza
  - Leverages available hardware and closely resembles the target release architecture
  - We still remain database agnostic for other deployments

- DISTRIBUTE ON Clause
  - Determines how data is distributed across database appliance (Netezza specific)
  - Candidate keys should have high cardinality and commonly used in joins
  - We chose IP address
Atypical behavior visualization (Clique)

► Behavior baseline for actors
  ■ Creates statistical model of what is typical for a given actor and category set
  ■ Visualizes the deviation from typical activity

► Actor / group hierarchy
  ■ Groups of IP addresses, a single IP address, or query based on an attribute
    ● Site > Facilities > Buildings > Individuals
  ■ Individually configurable and sharable

► Interactive interface provides semantic zooming (LiveRac)
  ■ Added adaptive bin widths, deviation highlighting, stability, and database independence
Future directions

- Investigate and implement alternative bottom up approach
  - statistical model per IP address and aggregation based on that model
- Improve interface performance
  - Investigate alternate middle tier architectures
- Enhance applicability by developing prototypes in different domains
- Incorporate abrupt outlier identification and visualization
How to get in touch

Daniel Best
@danzvzsec
daniel.best@pnnl.gov