Network Threat Detection

Blueprint's X-Challenge March 2022

Forward

Why this topic?

- Not using existing “DNS Threat Detection” accelerator...
  - That approach was aimed at phishing scams and name mangling
  - Targeted Limited Class of Problems
    - *Relied on External Threat Definitions*

- This Approach Is More Fundamental
  - Capture Network Traffic (pcap and netflow data)
  - Identify Anomalies in Traffic
  - Classify Threats Among Anomalies
    - *Allow Network Admins To Refine Definition of “Threat”*

Network Threat Detection

What do end users care about?

End Users Care About Threats

Threats vs. Anomalies

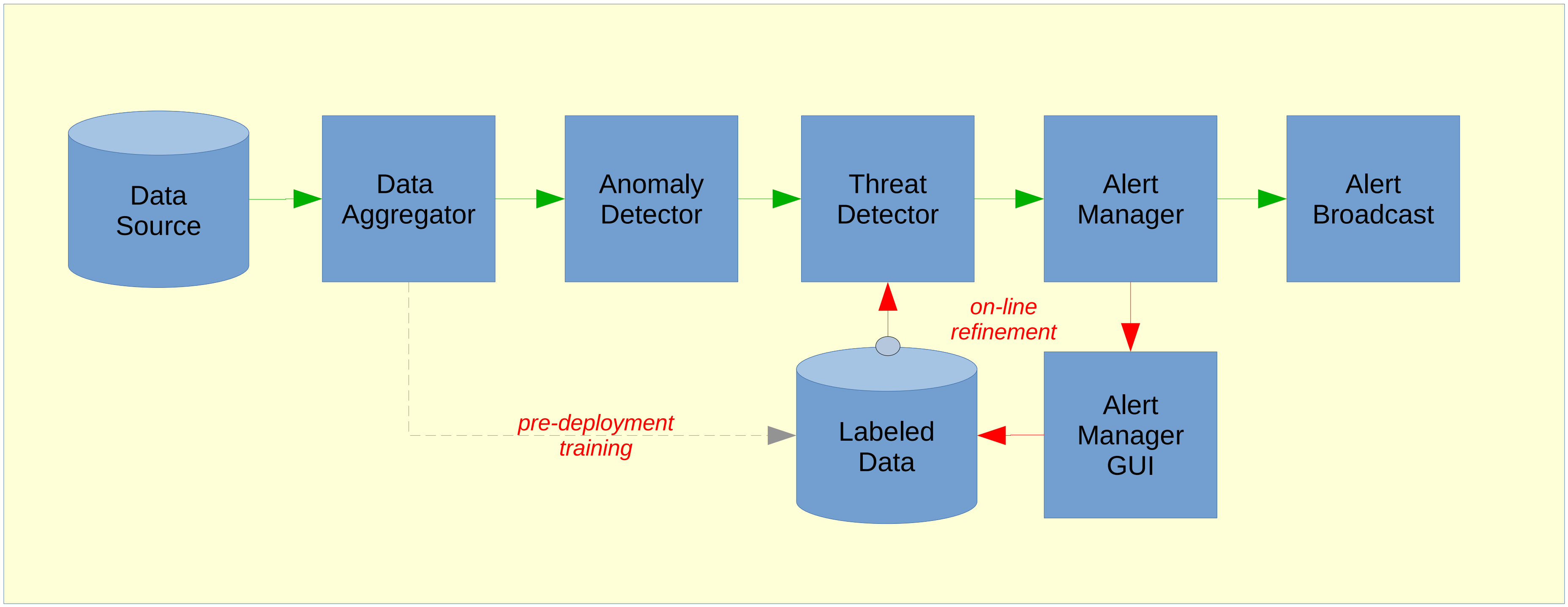
Threats Depend on User Context

Threats are Relevant Anomalies

A Threat Detection System

What makes up a Threat Detection System?

A threat-detection system must recognize anomalies in a data source and selectively report those anomalies that align with the end-user's notion of threat.



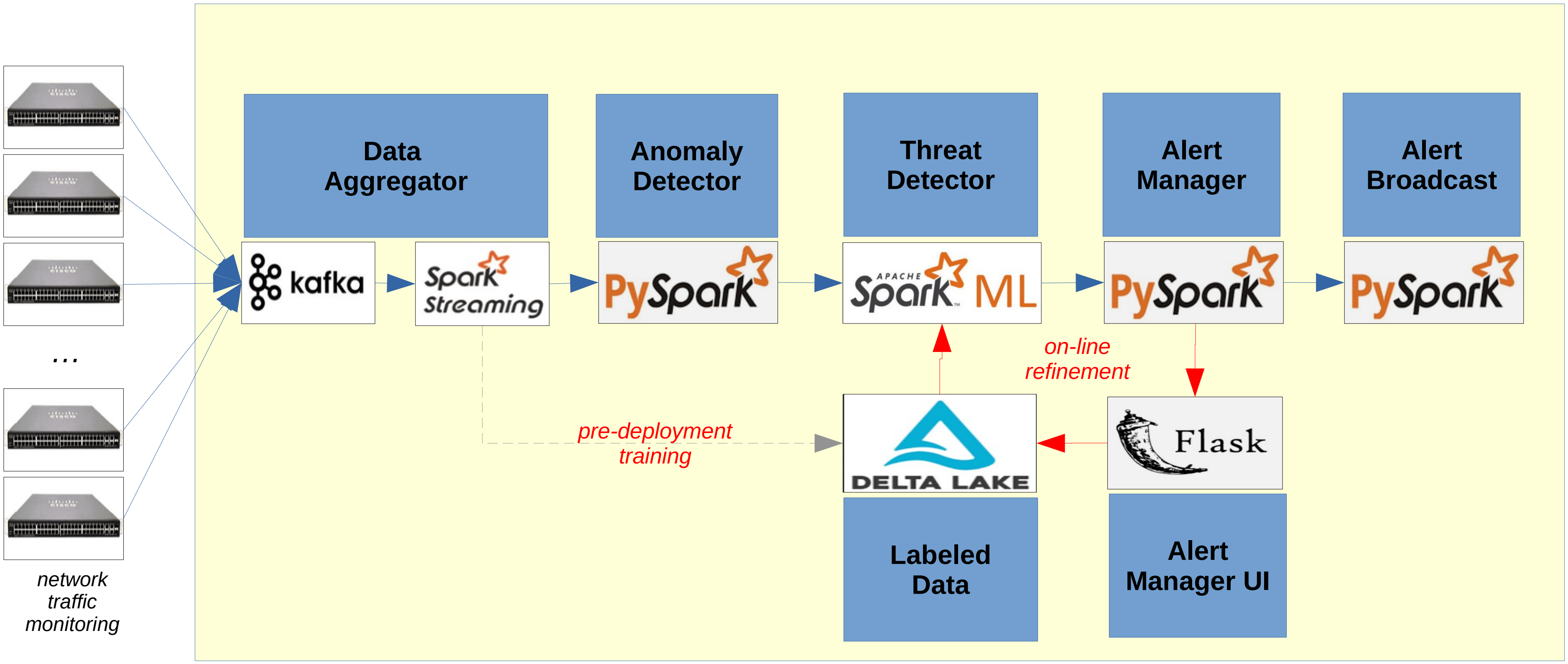
Typical approaches include:

- Pre-deployment training of ML classifiers for desirable alarms using existing threat definitions
- Post-deployment refinement by user contributed threat definitions from a user interface
- Some combination of the two approaches

A *Network* Threat Detection System

What might a Network Threat Detection System Look Like?

A threat-detection system must recognize anomalies in a data source and selectively report those anomalies that align with the end-user's notion of threat.

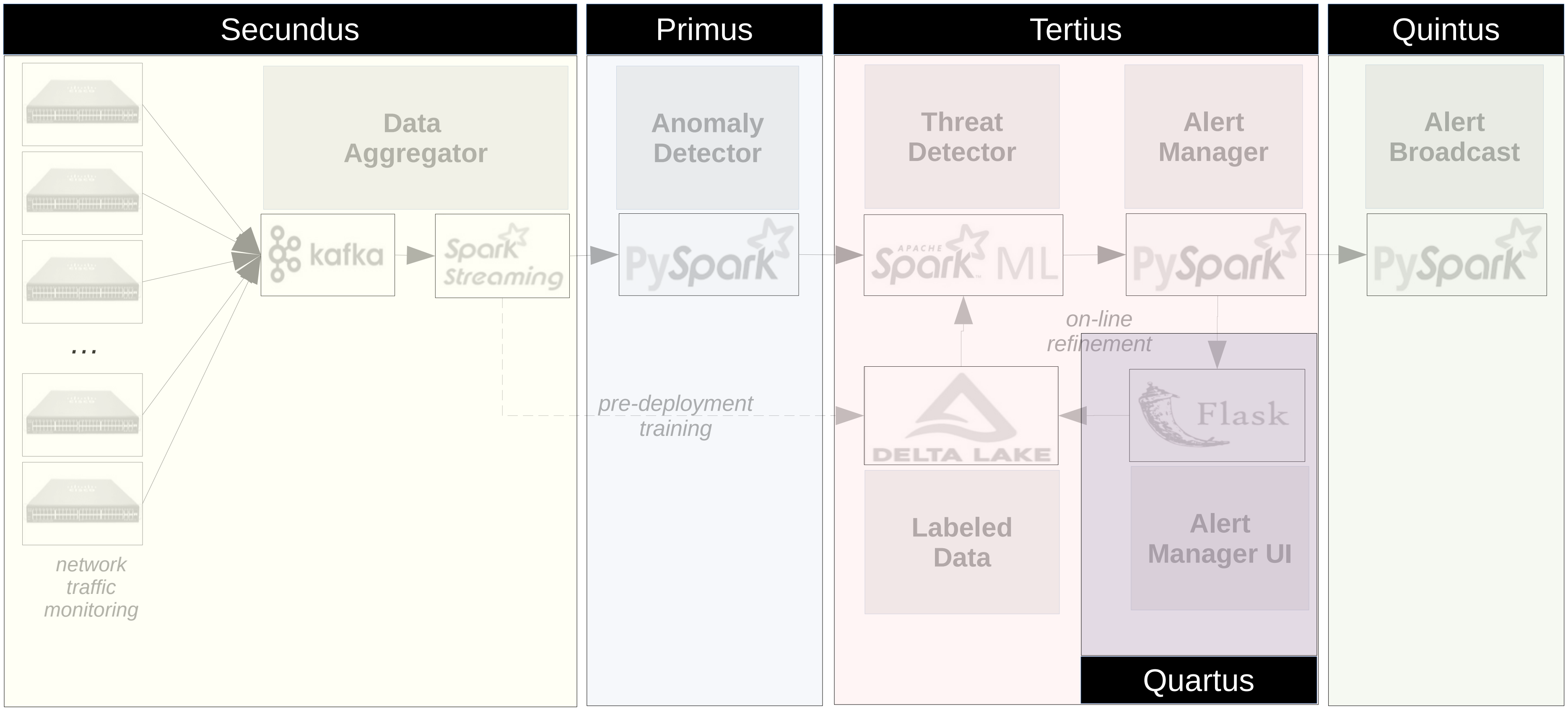


Typical approaches include:

- Pre-deployment training of ML classifiers for desirable alarms using existing threat definitions
- On-line refinement by user contributed threat definitions from a graphical interface
- Some combination of the two approaches

Implementing NTDS

Development Proceeds in Stages



- Primus – Explore and Implement Network Traffic Anomaly Detection
- Secundus – Implement A Streaming Solution for Collecting Data *at Scale*
- Tertius – Implement at Threat Identification / Threat-model / Model Refinement Loop
- Quartus – Develop and Refine the Alert Manager GUI
- Quintus – Develop and Refine the Alert Broadcast / Notification Module

Primus – Sample Data

Dataset-Unicauca-Version2-87Atts

- **From Kaggle (a familiar Data Science website) via the link:**
  - *https://www.kaggle.com/datasets/jsrojas/ip-network-traffic-flows-labeled-with-87-apps*
  - Captured April 26th, 27th, 28th, May 9th, 11th, 15th in 2017 at University of Cauca in Popayan, CO
  - 3.6 million rows of network traffic with 87 statistical features labeled with traffic protocols
  - Data was *labeled* using Rojas' "FlowLabeler" tool
    - *https://github.com/jsrojas/FlowLabeler*
    - Tool for processing either pcap files or live streaming data
    - Produces formatted data containing bidrectional statistics and application layer protocol

- **Distribution**
  - There are 1,501,758 million of the 3,577,296 events involve inbound or outbound traffic
  - There are 21,531 external sites that either sending or receiving traffic

- **Protocols**

  '99TAXI', 'AMAZON', 'APPLE', 'APPLE_ICLOUD', 'APPLE_ITUNES', 'BGP', 'BITTORRENT', 'CITRIX', 'CITRIX_ONLINE', 'CLOUDFLARE', 'CNN', 'CONTENT_FLASH', 'DEEZER', 'DNS', 'DROPBOX', 'EASYTAXI', 'EBAY', 'EDONKEY', 'FACEBOOK', 'FTP_CONTROL', 'FTP_DATA', 'GMAIL', 'GOOGLE', 'GOOGLE_MAPS', 'H323', 'HTTP', 'HTTP_CONNECT', 'HTTP_DOWNLOAD', 'HTTP_PROXY', 'INSTAGRAM', 'IP_ICMP', 'IP_OSPF', 'LASTFM', 'LOTUS_NOTES', 'MAIL_IMAPS', 'MICROSOFT', 'MQTT', 'MSN', 'MSSQL', 'MS_ONE_DRIVE', 'NETFLIX', 'NFS', 'NTP', 'OFFICE_365', 'OPENSIGNAL', 'OPENVPN', 'ORACLE', 'OSCAR', 'QQ', 'RADIUS', 'RTMP', 'SIMET', 'SKINNY', 'SKYPE', 'SNMP', 'SOCKS', 'SPOTIFY', 'SSH', 'SSL', 'SSL_NO_CERT', 'STARCRAFT', 'TEAMSPEAK', 'TEAMVIEWER', 'TELEGRAM', 'TIMMEU', 'TOR', 'TWITCH', 'TWITTER', 'UBUNTUONE', 'UNENCRYPED_JABBER', 'UPNP', 'WAZE', 'WHATSAPP', 'WHOIS_DAS', 'WIKIPEDIA', 'WINDOWS_UPDATE', 'YAHOO', 'YOUTUBE'

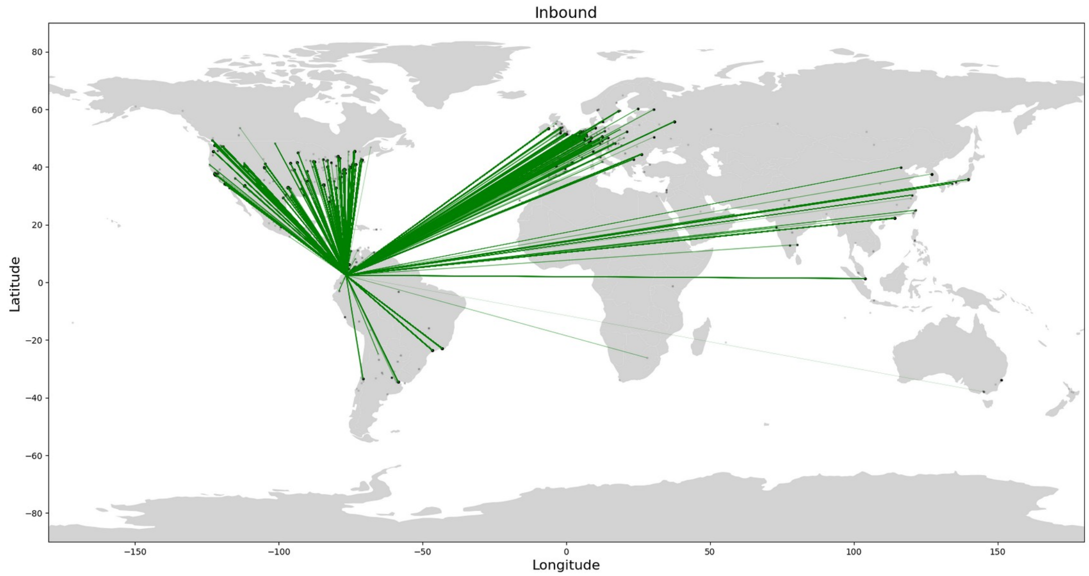
- **Statistical Features**

  'Flow.ID', 'Source.IP', 'Source.Port', 'Destination.IP', 'Destination.Port', 'Protocol', 'Timestamp', 'Flow.Duration', 'Total.Fwd.Packets', 'Total.Backward.Packets', 'Total.Length.of.Fwd.Packets', 'Total.Length.of.Bwd.Packets', 'Fwd.Packet.Length.Max', 'Fwd.Packet.Length.Min', 'Fwd.Packet.Length.Mean', 'Fwd.Packet.Length.Std', 'Bwd.Packet.Length.Max', 'Bwd.Packet.Length.Min', 'Bwd.Packet.Length.Mean', 'Bwd.Packet.Length.Std', 'Flow.Bytes.s', 'Flow.Packets.s', 'Flow.IAT.Mean', 'Flow.IAT.Std', 'Flow.IAT.Max', 'Flow.IAT.Min', 'Fwd.IAT.Total', 'Fwd.IAT.Mean', 'Fwd.IAT.Std', 'Fwd.IAT.Max', 'Fwd.IAT.Min', 'Bwd.IAT.Total', 'Bwd.IAT.Mean', 'Bwd.IAT.Std', 'Bwd.IAT.Max', 'Bwd.IAT.Min', 'Fwd.PSH.Flags', 'Bwd.PSH.Flags', 'Fwd.URG.Flags', 'Bwd.URG.Flags', 'Fwd.Header.Length', 'Bwd.Header.Length', 'Fwd.Packets.s', 'Bwd.Packets.s', 'Min.Packet.Length', 'Max.Packet.Length', 'Packet.Length.Mean', 'Packet.Length.Std', 'Packet.Length.Variance', 'FIN.Flag.Count', 'SYN.Flag.Count', 'RST.Flag.Count', 'PSH.Flag.Count', 'ACK.Flag.Count', 'URG.Flag.Count', 'CWE.Flag.Count', 'ECE.Flag.Count', 'Down.Up.Ratio', 'Average.Packet.Size', 'Avg.Fwd.Segment.Size', 'Avg.Bwd.Segment.Size', 'Fwd.Header.Length.1', 'Fwd.Avg.Bytes.Bulk', 'Fwd.Avg.Packets.Bulk', 'Fwd.Avg.Bulk.Rate', 'Bwd.Avg.Bytes.Bulk', 'Bwd.Avg.Packets.Bulk', 'Bwd.Avg.Bulk.Rate', 'Subflow.Fwd.Packets', 'Subflow.Fwd.Bytes', 'Subflow.Bwd.Packets', 'Subflow.Bwd.Bytes', 'Init_Win_bytes_forward', 'Init_Win_bytes_backward', 'act_data_pkt_fwd', 'min_seg_size_forward', 'Active.Mean', 'Active.Std', 'Active.Max', 'Active.Min', 'Idle.Mean', 'Idle.Std', 'Idle.Max', 'Idle.Min', 'Label', 'L7Protocol', 'ProtocolName'

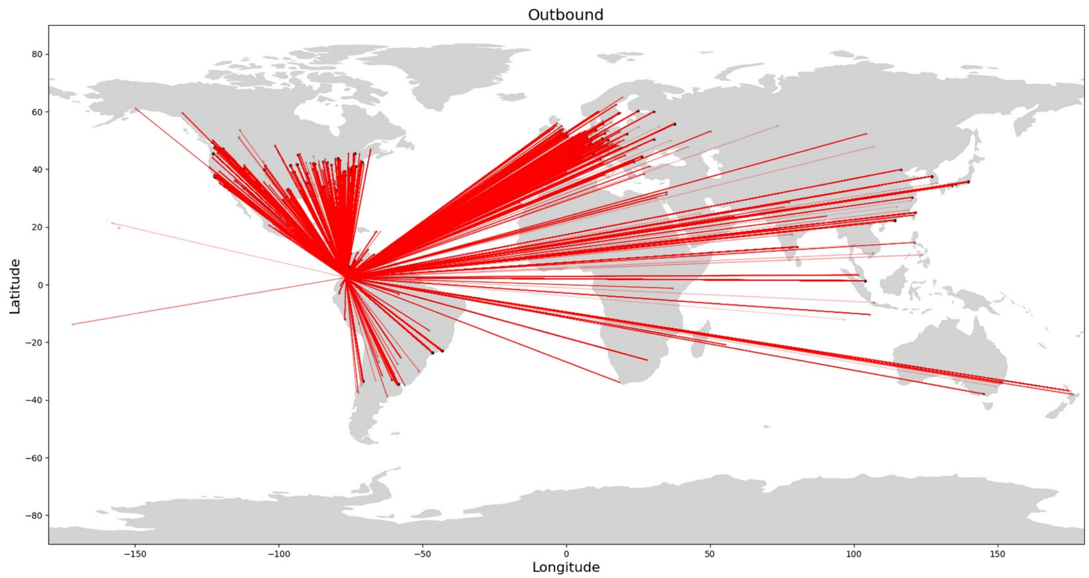
Data Source

Traffic Distribution

Traffic Entering The University's Network…



Traffic Leaving The University's Network...

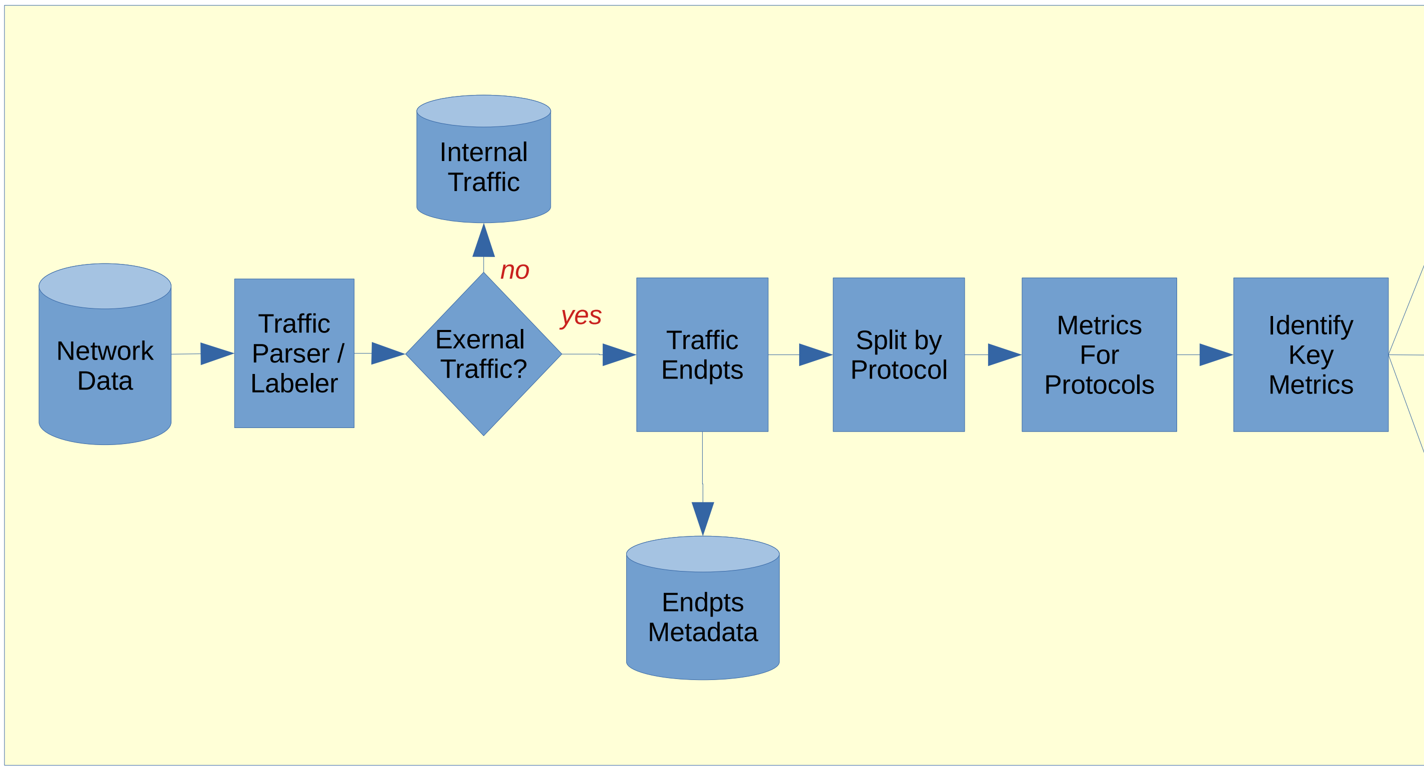


Primus – Anomaly Detection

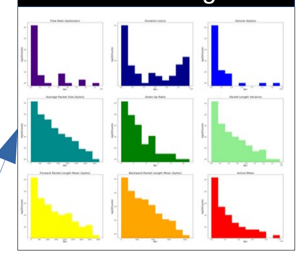
This X-Challenge Focuses on Detecting Anomalies

Notional Network Traffic Anomaly Detector

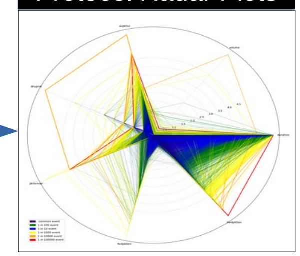
- *Focus on external traffic (entering or leaving local network)*



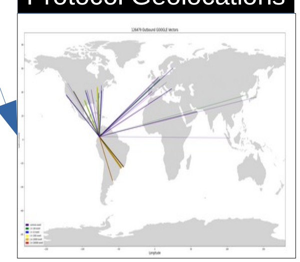
Protocol Histograms



Protocol Radar Plots

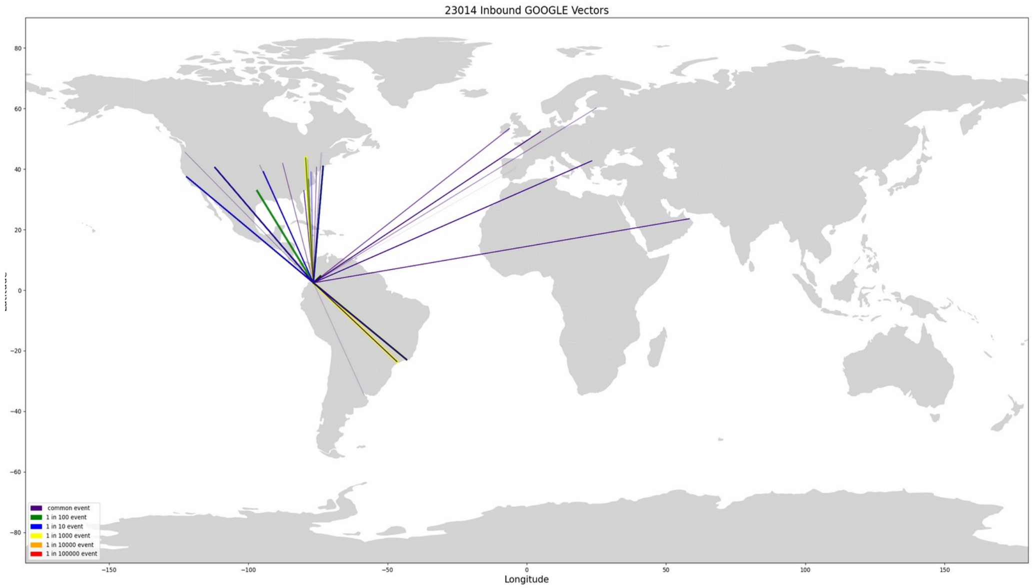


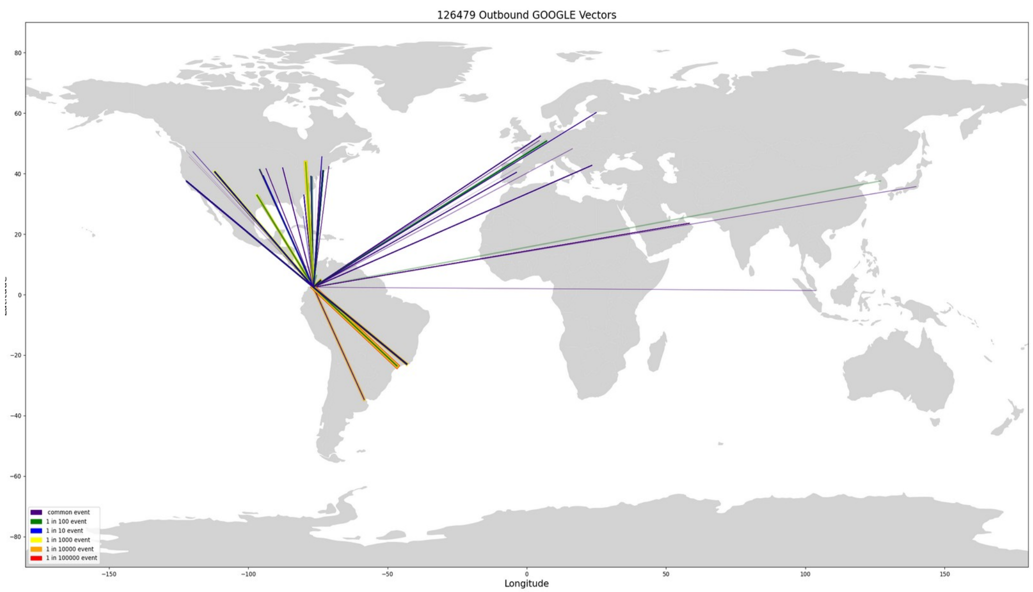
Protocol Geolocations

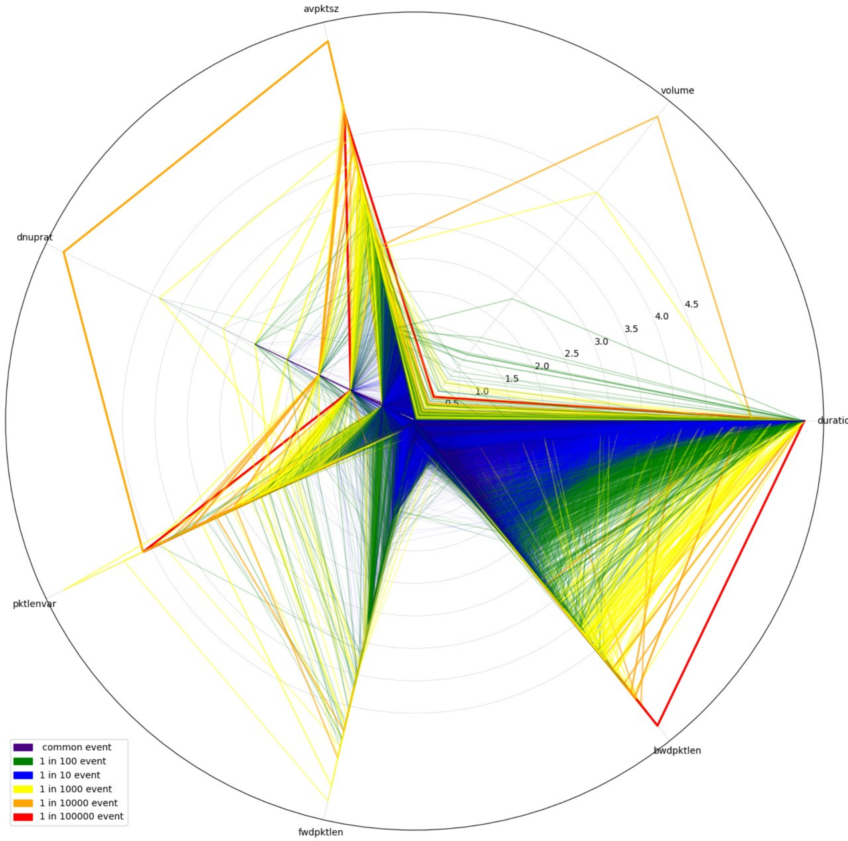


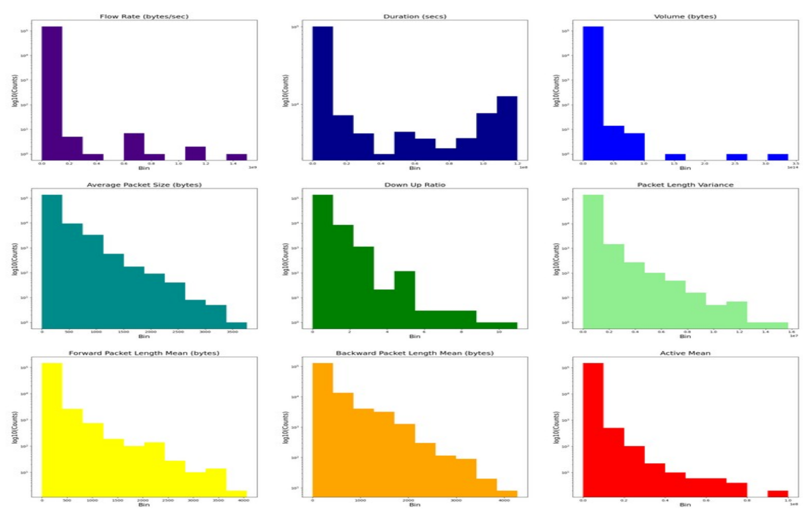
Primus – Sample Results.1

GOOGLE Traffic



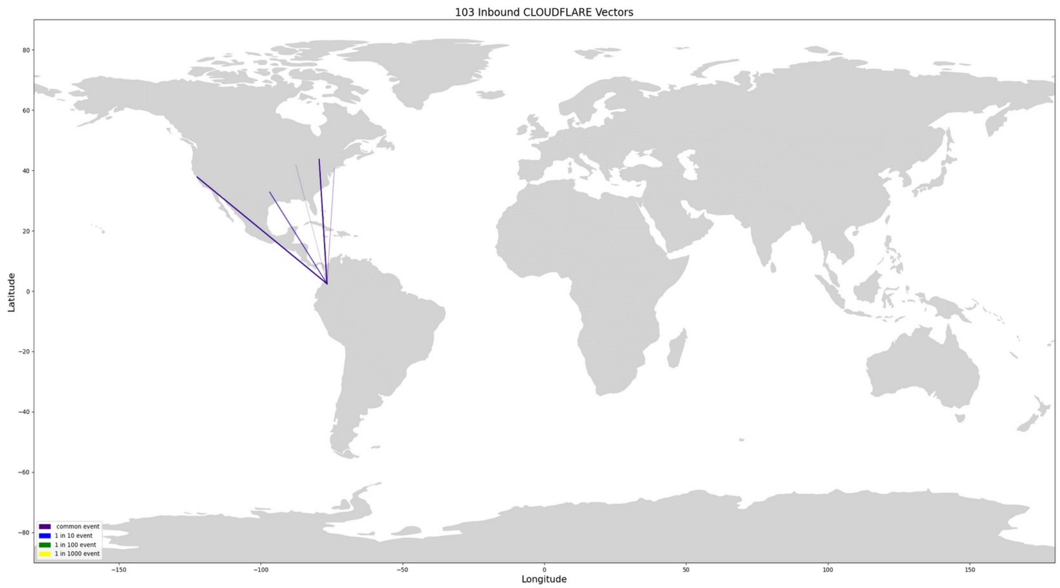


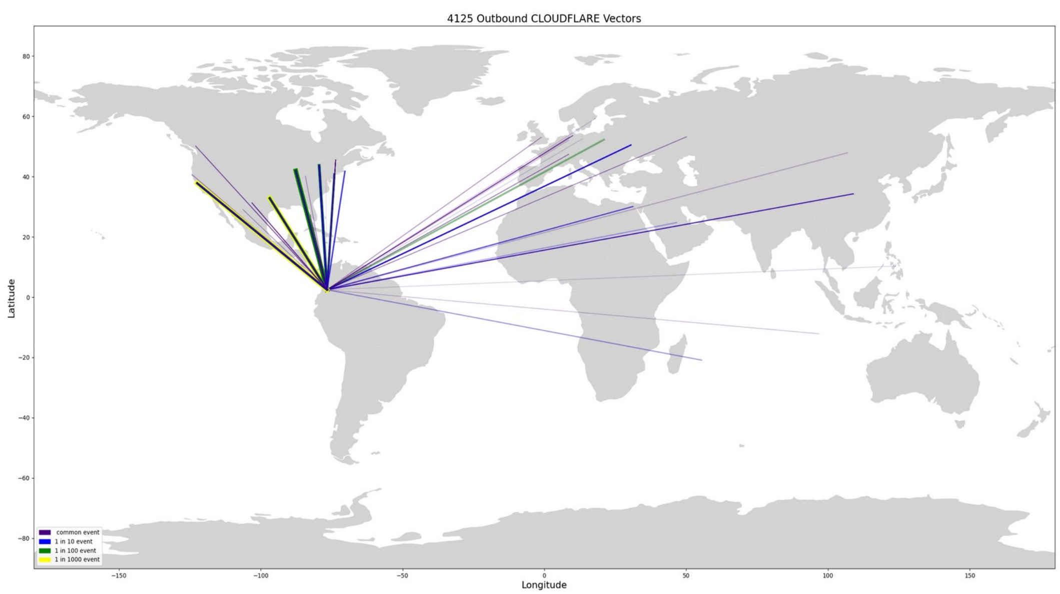


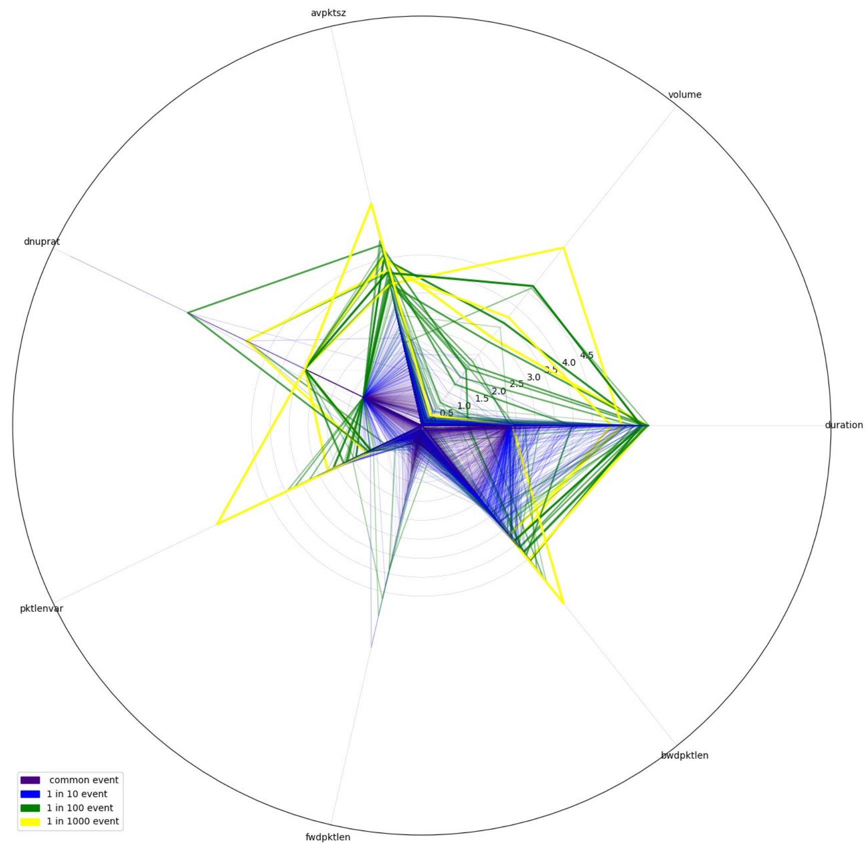


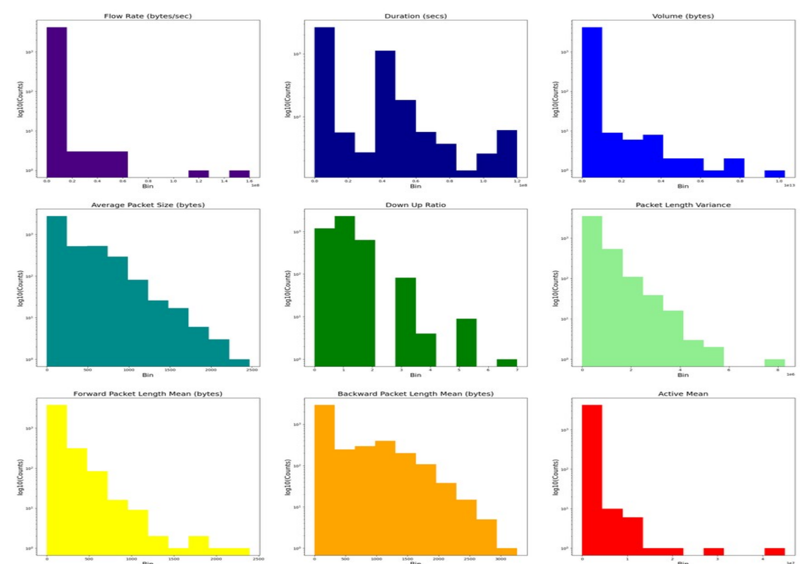
Primus – Sample Results.2

CLOUDFLARE Traffic









Discussion

Anomalies, Threats, False Alarms

- Identifying an Anomaly is based on *statistics...*

- Identifying a Threat is based on *relevance*

- Data Science techniques excel at recognizing Anomalies, *however Most Anomalies are **<u>not</u>** Threats*

- Successful Threat Detection Systems avoid identifying irrelevant anomalies as threats...

Novel Differentiation

How does this differ from the original accelerator?

Approach:

- Anomaly detection
- Initial Threat definition from historical data (optional)
- Active refinement of Threat Model (online training with user input)
- Alert Manager GUI

Techniques (not yet implemented)

- Kafka & SparkStreaming (to handle massive data volume)
- Pipelined decomposition of traffic into “FlowLabeler” format
- On-line Deep-Learning refinement of model
  - *Model adapts as users suppy inputs*
  - *Model adapts as nature of traffic evolves*
- Web based alerting and assessment UI

Market Alignment

How quickly could this be made into a product for our customers?

Market Alignment:

- Scalability Choices:
  - *Approach 1. Massive Scalability (use DataBricks ecosystem)*
    - Approach 1a. uses python and pyspark with DataBricks
    - Approach 1b. uses Scala and SparkML with DataBricks
  - *Approach 2. Modest Scalability (use standard Data Science ecosystem)*

- Elements Needed
  - *Need to develop the threat detector*
  - *Need to develop the alert manager UI*
  - *Need to fully flesh out the training loop*
  - *Need to implement the streaming pipeline*
  - *Need to port various bits and pieces*

- Development Timescales:
  - *Approach #2 – Standard Data Sci stack is mature*
    - Fast, perhaps 10-12 weeks to MVP
  - *Approach #1a – PySpark + Python Data Sci on DataBricks*
    - Medium, perhaps 15-18 weeks to MVP
  - *Approach #1b – Scala + SparkML + 1st Principles Data Sci on DataBricks*
    - Slow, perhaps 24-28 weeks to MVP

Partnership Alignment

How difficult would it be to adapt this for our partners to use?

Partner Uptake:

- Nearly all of the proposed technologies...
  - *Kafka, SparkStreaming, PySpark, SparkML, DeltaLake*

*...*are found in the DataBricks ecosystem!