CounterFlow ai

What is an “Analyzer”?

A Primer On ThreatEye’s* Approach To Machine Learning

At CounterFlow AI, the term “analyzer” describes the programs or scripts that fuel its intelligent sensor. Up until now, the vernacular of machine learning has centered around “models”. An analyzer, however, is a broader term that encomapsses the entire scriptable framework that provides the logic and organization for models and analyses.

Analyzers combine counts, rules, statistics, advanced analytics and machine learning models to solve specific network security and network performance use cases. Because each network is unique, an analyzer can be used either in standard form or customized for specific situations. ThreatEye® Sensor also provides data caching and a statistical infrastructure to support the development of new analyzers.

In practical terms, analyzers provide an end-user with a programmable interface for running multiple analytics and models in parallel, either on the full stream of network data or on a filtered stream of specific flow types. In addition to running *n* number of scripts, ThreatEye Sensor can ingest both Threat Intelligence data and policy rules as additional input sources for machine learning.

As network dynamics continue to evolve, the flexible nature of ThreatEye’s scripting architecture allows customers to implement new analyzers as new interests and use cases arise.

Crypto Analyzers

statistically investigate and classify encrypted network traffic.

EXAMPLES

- Web Application Identification
- TLS Encryption Protocol Tracking
- OS and Host Fingerprinting

Flow Analyzers

utilize flow information and computed fields to identify anomalous traffic.

EXAMPLES

- Novelty Detection
  - MAC address
  - Flow metrics: Protocol, IP
  - Domain (TLD, Second Level Domain)
- Layer 7 Mismatch
- PCR DNS

Graph Analytics Analyzers

apply graph theory methods to the incoming stream of connections.

EXAMPLE

- Graph Clustering
- Communities of Interest ID

Policy Analyzers

enforce specific rules and network policies on incoming network traffic.

EXAMPLES

- Allowed Servers (DNS, DHCP, NTP ,et al)
- Time Fence
- Geo Fence
- TLS Enforcement

Statistical Flow Analyzers

apply statistical models to available flow information and computed fields to identify anomalous traffic.

EXAMPLES

- Time Anomaly
- Country Anomaly
- DGA Detection

Advanced Anomaly Detection

- Streaming Point Process models
- Naive Bayesian Joint Modeling (Multi-variable histogram estimation)

Timeseries Analyzers

look at patterns of traffic over time to identify unexpected bursts and changes in mean and variance.

EXAMPLES

- Burst/Outlier Detection
  - Z-score
  - Median Absolute Deviation
- Changepoint detection
  - Mean
  - Variance

**ThreatEye® refers to CounterFlow AI’s AIops Network Forensics Platform*

Anatomy of an Analyzer

ThreatEye API provides models and utility functions for applying machine learning

Analyzers use a powerful, complete scripting language based on the Lua syntax

Each analyzer must define two required functions:

REQUIRED FUNCTION 1

Setup to initialize the model

REQUIRED FUNCTION 2

Loop to apply the model to each incoming message

```
-- ------------------------------------------------
-- Import necessary API functions
-- ------------------------------------------------
require 'analyzer/utils'
require('analyzer/feature') -- contains data transformation helper functions
require 'analyzer/logistic-regression'

local analyzer_name = 'Logistic Regression Analyzer'
-- ------------------------------------------------
-- Setup: Initialize Analyzer Settings
-- ------------------------------------------------
function setup()
    -- Logistic Regression model parameters
    weights = {-2.42064408839,1.99545844518,-1.70241301806,0.00306950873423,
    -0.15181071537,2.29227920798,-0.797475159835,-1.96870692592,-0.883096645108,-0.99507358149,
    -0.574770439156,0.279708188326,-0.0993450421238,3.18976170491,1.35485232316,0.151034941312,
    0.154731245945,0.618496661902,-0.17331944488,-0.251627327936,0.111298556062,-1.06460092327,
    -0.285532716969,-0.555498868092}
end

-- ------------------------------------------------
-- Loop: Apply analyzer to the message
-- ------------------------------------------------
function loop(msg)
    -- Confirm the message is complete
    if not check_fields(msg, {'domain', 'ips', 'seq', 'stime'}) then
        return
    end
    -- Apply model to score the domain
    pred = logreg_classify(transform(msg.domain), weights)

    -- Streaming path provides input for other analyzers
    msg['score'] = pred  -- Add score to original message
    dragonfly.analyze_event(default_analyzer, msg) -- pass msg to chained analyzers

    -- Archival path stores data for retrospective analysis
    label = { seq=msg.seq, label=string.format("dga_score=%0.3f", score) }
    dragonfly.output_event("labels", label)
end
```

Analyzers can integrate with other analyzers, or downstream processes such as packet capture

Each output is archived for future retrospective analysis

LEARN MORE AT: www.CounterFlowAI.com

CounterFlow ai

About CounterFlow AI, Inc.

CounterFlow AI is a cybersecurity software company offering an AIOps platform for network forensics. The flagship product, ThreatEye®, integrates advanced security technologies into a streaming machine learning pipeline to identify network faults, anomalies, and threats at wire speed. ThreatEye® is built for hybrid cloud deployments to easily extend customer network and security operations.

CounterFlow AI, Inc.

6181 Rockfish Gap Turnpike, Crozet, VA 22932
info@counterflowai.com