Securing Your Network with Anomaly Detection using Distributed Learning Architecture (Learning Networks)

JP Vasseur, PhD, Cisco Fellow – jpv@cisco.com
Sukrit Dasgupta, PhD, Tech Lead – sukdasgu@cisco.com
What Self Learning Networks is About ...

• SLN is fundamentally a hyper-distributed analytics platform ...

• Putting together analytics and networking ...
  • Goldmine of untouched data on networking gear (sensing)
  • Network learns and computes models on premise (analytics)
  • The Network adapts, modifies its behavior (control)

• SLN for Security: attacks are incredibly sophisticated and targeted, exfiltration of data being a major concern, requiring a next-generation approach => Stealthwatch Learning Networks

• True Technology disruption ...
Botnets and Data Ex-Filtration Techniques

• Size can range from thousands to millions of compromised hosts

• Botnet can cause DDoS & other malicious traffic (spam, ...) to originate from the inside of the corporate network

• C&C (C2) servers become increasingly evasive
  • Fast Flux Service Networks (FFSN), single or double Flux
  • DGA-based malware (Domain Generation Algorithms)
  • DNS/NTP Tunneling
  • Peer-to-Peer (P2P) protocols
  • Anonymized services (Tor)
  • Steganography, potentially combined with Cryptography
  • Social media updates or email messages
  • Mixed protocols ....
  • Timing Channels
A true paradigm shift

(Current) Generation of Security Architectures and Product

- Specialized Security gear connected to the network (FW, IPS, ...)
- Heavily signature-based ... to detect *known* Malwares
- Dynamic update of signatures

SLN is Machine Learning based and pervasive

- Use of adaptive Machine Learning (AI) technology to detect advanced, evasive Malware: build a model of normal pattern and detect outlier (deviations)
- High focus on 0-day attacks
- Use every node in the network as a security engine to detect attacks
- Complementary to all other technologies (FW, IPS, ...)
SLN Architecture

- Orchestration of SLN Agent
- Advanced Visualization of anomalies
- Centralized policy for mitigation
- Interaction with other security components such as ISE and Threat Intelligence Feeds
- North bound API to SIEM/Database (e.g. Splunk) using CEF format
- Evaluation of anomaly relevancy

**SLN Controller**

**SLN Agent**

- **Sensing (knowledge):** granular data collection with knowledge extraction from NetFlow but also Deep Packet Inspection on control and data plane & local states
- **Machine Learning:** real-time embedded behavioral modeling and anomaly detection
- **Control:** autonomous embedded control, advanced networking control (police, shaper, recoloring, redirect, ...)

**ISE**

**Threat Intel**

**Internet**

**WAN**

- Threat Grid, OpenDNS, WBRS, ... Other TI feeds
An Open Architecture (Controller)

Identity Services Engine Context Enrichment: IP Address (key), Audit session ID, User AD Domain, MAC address, ESP Status, NAS IP & port (important !!), Posture, TrustSec information including SGT, Endpoint Profile name,

Syslog messages using CEF format pushing anomalies events into DB and SIEM

Various Source of Threat Intelligence: Talos (blacklists), Threat Grid (sandboxing), OpenDNS (AS, URL, historical association to domains, ...)

API triggering Mitigation form external Sources such as Firewall, IPS/IDS, ... Abstracting networking complexity

© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential
Before we start ... Few (random) facts:

• Two camps 😊 ... Super Pro ML and Anti-ML, both have good arguments
• Extremely wide range of ML algorithms with no one-size-fits-all
• Machine Learning/AI incredibly powerful if applied to solve the right problems
• Hard to tune ? Yes if naively applied ...
• Capable of solving all issues ? Not quite but still ...
• Other aspects that do matter: Interpretability, scalability and UX is essential
• Is is that disruptive ? Hard to do (constrained environments, high scale, ... )
Discussing Recall, Precision, FP, ...

• Few simple notions required when discussing Machine Learning: False Positive (FP), True Positive (TP), False Negative (FN), True Negative (TN), Recall and Precision.
• Take a Classifier C trained to detect if an event E is relevant (Like) or not (irrelevant).
  • TP: E is classified as relevant and is indeed an relevant
  • FP: E is classified as relevant and is in fact irrelevant (noise)
  • TN: E is classified as irrelevant and is indeed irrelevant
  • FN: E is classified as irrelevant and is in fact an relevant
• Recall = TP / (TP + FN) (notion of sensitivity)
• Precision = TP / (TP + FP) (positive predictive value)
• Accuracy ACC = (TP + TN) / (TP + TN + FP + FN),
  • Example: if a classifier that is trained to detect dogs in a picture detects 15 dogs, only 10 of them are dogs, and there are 20 dogs in the picture then the Precision = 10/15 = 0.66 and Recall = 10/20 = 0.50
Clusters, Self Organizing Learning Topology and Anomalies

Key question: how can we model host behaviors?

- Modeling mixed-behaviors unavoidably leads to hiding anomalies ...
- The fundamental idea of dynamics clustering is to “group” devices according to behavioral similarity
- Self Organizing Learning Topologies (SOLT): ability to build Virtual topologies used to learn models between dynamic clusters
  - For example, find a model for the traffic from cluster A to cluster B, for HTTP traffic,
Dynamic Clustering

Cluster: known/internal/network

Cluster: known/internal/collab

Cluster: known/internal/inet::windows
Life of an Anomaly

SCA

DLA

Anomaly

Clustering: dynamic clustering according to behavioral degree of similarity

SOLT

NSC: Traffic analysis from multiple data feeds
Hierarchical ML Models

Model

Collab models
C1-D1, C1-D2, C1-D3, C2-D1, ...

File Transfer models
C1-D1, C1-D2, C1-D3, C2-D1, ...

Collab models
from C1, from C2, from C3

File Transfer models
from C1, from C2, from C3

Voice

Printing

Collab

File Transfer

Cluster Layer

Scr/Dest Cluster Layer

NYC

Germany

Boston
Multi-dimensional and Hierarchical models using stateless/statefull features

Inside a Model ...

(hundreds of dimensions) ...

High number of dimensions extracted from multi feeds (Netflow, DPI)

Rich DNS features: avr names length, # of consecutives vowels, average entropy of characters, ...

Multi-layer: cluster-cluster-app, cluster-app, app
Computing “SOLT” Scores

• Each scored flow update is evaluated against prior observations, computing the rank of the score over a sliding time window.

• Flow updates are then marked as anomalous or not based a set of criterion to be met (Maximum rank to be considered as anomalous, Score value, # of samples contributing to model, Maturity of the model (# of samples, time, ...).

• Boosting based on Expert knowledge (application sensitivity, # of features, ...)

• Computes an anomaly score and select TOP anomalies
Anomaly
Life of an Anomaly

SCA

DLA

NSC

Clustering: dynamic clustering according to behavioral degree of similarity

Modeling: dynamically learned baseline with multiple layers, high dimensions space, anomaly detection

NSC: Traffic analysis from multiple data feeds
Selective Anomaly Forwarder (SAF) and Selective Anomaly Puller (SAP)

1. When an anomaly is detected by a DLA, its Selective Anomaly Forwarder decides whether this anomaly is worth being sent to the SCA
2. If the SAF decides to forward the anomaly, a digest of the anomaly is sent to the SCA
3. When a digest of an anomaly is received by the SCA, its Selective Anomaly Puller decides whether this anomaly is worth being completely pulled
4. If the SAP decides to pull the anomaly, all the information about this anomaly is requested to the DLA
Selective Anomaly Forwarder (on the DLA)

- SAF role is to select the most interesting anomalies to be forwarded to the SCA according to Score of the anomaly, According to a forwarding Budget, *with exploration*

![Diagram showing Forwarded Anomalies and available budget](image-url)
Selective Anomaly Pullers (on SCA)

- SAP role is to select the most interesting anomalies from all DLAs to be shown to the user, according to Score of the anomaly for a given DLA and across all DLAs (ensuring good diversity of anomalies), local Budget with exploration.

Distributed Relevance Learning explained later in great details.
Life of an Anomaly

Anomaly Detection System (ADS)

1. **NSC**: Traffic analysis from multiple data feeds
2. **SOLT**: Modeling: dynamically learned baseline with multiple layers, high dimensions space, anomaly detection
3. **Modeling**
4. **Scoring & Ranking**
5. **Clustering**: dynamic clustering according to behavioral degree of similarity
6. **Selective Anomaly Forwarder**: select the most interesting anomalies according to their score, with exploration

Diagram: SCA → DLA (Life of an Anomaly) → Anomaly Detection System (ADS)
Traditional Anomaly Detection Systems

- Focus on Detection (wrong)
- Core challenge is not Detection itself but Precision (avoid False Positive / Irrelevant alarms)

SLN Approach

- Efficient detection and Precision
- Make the Network learn form its own mistakes and eliminate False Positive!
- There is a notion of subjectivity too
- Not a feature but an Architecture
Relevance can be subjective too!
Reinforcement Learning: Actor

Public/Private Network

SCA

DLA

Distributed Relevance Learning

Statistical Classifier

\[ P(y = +1|x; \theta) = \Phi \left( \frac{x^T \mu}{\sqrt{x^T \Sigma x + 1}} \right) \]

Optimal Forwarder

\[ g(x) = \sum_y -\theta(y|x; \theta) \log_2(\theta(y|x; \theta)) \]
Challenges ...

• Design an algorithm with the following properties:
  1) Remove False Positive (FP) (anomalies that are not of interest)
  2) Do not remove true positive (anomalies that are relevant)
  3) Learn quickly (do not require too many feedback from the user)
  4) Be consistent across data set (robustness)
  5) Handle inconsistency between users, changing decisions (unlearn)
Evaluating DRL Performance

• Dozens of metrics that can be used to measure DRL efficacy

• The performance $P_{DRL}$ of our algorithm is measured in terms of the proportion of “Dislike” (noise, irrelevant) that have been suppressed by the system.

$$P_{DRL} = \frac{\alpha_N - \alpha_D}{\alpha_N}$$

where $\alpha_D$ and $\alpha_N$ are the proportion of dislikes shown to the user with and without DRL

• A "random" classifier achieves a DRL performance of zero (likes and dislikes are suppressed at the same rate).

• A performance of 0.5 can be interpreted as a reduction of 50% of "false positives"
Life of an Anomaly

- **NSC**: Traffic analysis from multiple data feeds
- **SCA**: Clustering: dynamic clustering according to behavioral degree of similarity
- **DLA**: Distributed Relevancy Learning: Likelihood of relevancy (False Positive reduction)
- **SOLT**: Selective Anomaly Puller: select the most interesting anomalies according to their score per DLA and across all DLAs, with exploration
- **DRL**: Selective Anomaly Forwarder: select the most interesting anomalies according to their score, with exploration
- **Modeling**: dynamically learned baseline with multiple layers, high dimensions space, anomaly detection
- **Relevancy Learning**: Likelihood of relevancy (False Positive reduction)
- **Anomaly Selection**: Anomaly 1, Anomaly 2, Anomaly 3, Anomaly 4, Anomaly 5, Anomaly 6, Anomaly 7, Selective Anomaly Forwarder: select the most interesting anomalies according to their score per DLA and across all DLAs, with exploration

---

**Selective Anomaly Puller**: select the most interesting anomalies according to their score per DLA and across all DLAs, with exploration

**Selective Anomaly Forwarder**: select the most interesting anomalies according to their score, with exploration

**Modeling**: dynamically learned baseline with multiple layers, high dimensions space, anomaly detection

**Relevancy Learning**: Likelihood of relevancy (False Positive reduction)
On-Premise Edge Control

- Honeypot (Forensic Analysis)
- Controller infrastructure
- SCA
- Public Internet
- Control Policy
  - Smart Traffic flagging
  - Traffic segregation & selection
  - Network-centric control (shaping, policing, divert/redirect)

- DSCP ReWrite CBWFQ
- DLA
- Shaping
Packet Capture of Anomaly

- **Anomaly Detected**: The DLC detects an anomaly in the traffic and gathers all the details to characterize it: time, IP etc.

- **PBC Message**: Sends a message to the PBC with the characteristics of the anomaly

- **Anomaly Message**: Receives the anomaly details from the DLC

- **PBC Search and Extract**: Searches for all the packets that match the anomaly characteristics and extracts them to a compressed PCAP file

- **PCAP storage**: Maintains list of files per anomaly and purges unused files periodically

- **Push files**: Pushes all PCAP files for an anomaly from the DLA when a user requests it

- **Packet Details**: File contains packets that have either source or destination IP of the anomaly. Allows to see all activity around the anomaly

- **PCAP Size**: Typically ~ 10KB-100KB, 10K-500K packets
Quick Status on SLN ...

Findings ?

• The system does learn, *as expected*
• Relevant detected anomalies (time of day, volume, unexpected flows, long live flows, ....)
• SLN detected anomalies it had not explicitly programmed for (Cognitive Computing)
• *Does it detect everything without False Positive ?* No, such system simply do not exist but SLN learns and *quickly* adjusts to customer relevancy learning
• The Place In the Network (PIN) is fundamental => dramatically extending the protection surface and visibility
Anomaly: Tor client on corporate network

- Tor = anonymous/tunneled browsing system based on encryption and multiple hops
- Host on Beta customer network opened SSL connection to 3 Tor nodes
  - 2 are located in Europe, a 3rd one has a Japanese hostname but is geolocated in the US
Anomaly: retail branch subnet scanned for Telnet & SSH

- Host external to the branch performing a scan of ports TCP/22 & TCP/23
- Very subtle scan on a narrow scope and probing only two ports per host
• Abnormally high number of DNS requests for a printer
• Mix of UDP and TCP for DNS is also unusual
Anomaly: branch device scanning across the WAN

- Branch host is scanning addresses located elsewhere on the corporate network
- Wide port scan, NMAP-style
Anomaly: new branch host detected at night

- New host appears on branch network and starts Windows logon sequence
- Behavior is unusual at this time of day (after 6pm local time)
Anomaly: SSH session causing a large number of TACACS+ requests

- Branch network device performs 280 TACACS+ requests in a few seconds
- Occurs while an SSH session to the device was active
- Most likely command authorization and/or accounting requests
Anomaly: branch host transfers 2GB from SSH server running on HTTPS port

- Branch host downloads 2GB of data from an SSH server on the internet
- SSH connection terminates on port 443 which is assigned to HTTPS
- Manual check confirms port misuse, most likely to evade simple L4 firewalls
Anomaly: branch host performs miniature SYN Flood on server

- Nearly a thousand incomplete TCP handshakes to a CIFS server within <1 minute; almost like a miniature SYN Flood attempt
Anomaly: malware Command & Control using DNS as covert channel

- Active malware Command & Control (C2) channel going to another country
- Using DNS as covert channel (not fully RFC compliant, but enough to be classified as DNS)
- Only detected by SLN, although FW and IPS/IDS were active on the network
(Hyper) Distributed Architecture ... Scale

This *is* the challenge

Learning ... Adaptive, Ease of Use

With dynamic False Positive Reduction

Lightweight ... Pervasive

Conclusion
Demo time !