Entropy approaches to detecting attacks

Michele Pagano

Department of Information Engineering
University of Pisa

XIII International Asian School-seminar
Problems of complex systems' optimization
September 20-23, 2017
Akademgorodok

Outline

Outline

General Background

Intrusion Detection System

An intrusion detection system or IDS is a software/hardware tool used to detect unauthorized accesses to a computer system or a network

Anomaly based IDS

- Identifies intrusions by classifying activity as either anomalous or normal
- Needs a training phase to recognize normal activity
- Able to detect "new" attacks
- Generates more false alarms than a misuse based IDS

Anomaly based IDS

Key elements of an Anomaly based IDS

- Statistical method
- Aggregation level of the input data
- Information used for detection

Information theoretic approaches

- Different definitions of entropy
- Different level of data aggregations and interpretation of the entropy
  - *Randomly* aggregated data: entropy of a distribution
  - Flow level analysis: entropy of a string
- Different traffic descriptor (for the aggregated data)

Shannon entropy

- Discrete distributions P, Q with a finite number L of elements
- Entropy of a RV X (or of its distribution P), often called Shannon entropy

$$H(X) = -\sum_{l=1}^{L} p_l \log_2 p_l = \mathbb{E}\left[-\log_2 P(X)\right]$$

where $P = \{p_1, p_2, \ldots, p_L\}$ is the probability distribution of X

$$0 \leq p_l \leq 1 \quad \text{and} \quad \sum_{l=1}^{L} p_l = 1$$

- $H(X)$ is a measure of the uncertainty (or variability) associated with the RV X

$$0 \leq H(X) \leq \log_2 L$$

  - The infimum corresponds to the degenerate distribution (i.e., $p_l = \delta_{k-l}$ for some $k \in [1, L]$)
  - The supremum is attained in case of uniform distribution (i.e., $p_l = 1/L \; \forall l$)

Beyond Shannon

- In anomaly detection, entropy can be used to "summarize" the distribution of specific traffic features $\Rightarrow$ variations between its values for P_{cur} and P_{ref}
- Shannon entropy weights each p_l according to its logarithm

$$H(X) = -\sum_{l=1}^{L} p_l \log_2 p_l = \mathbb{E}\left[-\log_2 P(X)\right]$$

  - If the natural logarithm is is used, well-known Boltzman–Gibbs entropy in statistical mechanics
- More general definitions of entropy that provide additional information about the importance of specific events
  - An additional parameter must be introduced
  - Generalised entropies: Tsallis and Rényi entropies
- *Measure* the difference between the two probability distribution
  - Relative entropy between two distributions
  - Kullback–Leibler divergence and its symmetrized version, Jensen–Shannon divergence

Tsallis entropy

$$S_q(X) = \frac{1 - \sum_{l=1}^{L} p_l^q}{q - 1}$$

- $q \in \mathbb{R}$ is the nonextensivity parameter or entropic index
- $q \to 1 \quad \Rightarrow$ usual Boltzmann–Gibbs entropy
- $S_q(X) = 0$ in case of degenerate distribution
- The maximum is attained in case of uniform distribution

$$S_q^{\max} = \frac{1 - L^{1-q}}{q - 1}$$

- When q assumes large positive values, $S_q(X)$ is more sensitive to events that occur often
- For large negative q rare events contribute more
- If two systems A and B are independent (i.e., $p_{lm}^{A+B} = p_l^A \cdot p_m^B$), then

$$S_q(A + B) = S_q(A) + S_q(B) + (1 - q) S_q(A) S_q(B)$$

Rényi entropy of order α

$$H_\alpha(X) = \frac{1}{1-\alpha}\log_2\left(\sum_{l=1}^{L} p_l^\alpha\right) \qquad \alpha \geq 0 \quad \alpha \neq 1$$

- In general, it is a non-increasing function in α, apart from the case of uniform distribution (when $H_\alpha(X) = \log_2 L \ \ \forall\alpha$)
  - H_0 is the Hartley entropy of X:

$$H_0(X) = \log_2 L$$

  - the limiting value of H_α as $\alpha \to 1$ is the standard Shannon entropy

$$H_1(X) = H(X) = -\sum_{l=1}^{L} p_l \log_2 p_l$$

  - as $\alpha \to \infty$, H_α converges to the min-entropy

$$H_\infty(X) = \min_l(-\log p_l) = -\log \max_l p_l$$

Relative entropies

- Kullback–Leibler divergence of Q from P

$$D_{\mathrm{KL}}(P\|Q) = \sum_{l=1}^{L} p_l \log \frac{p_l}{q_l}$$

under the assumption of absolute continuity: $q_l = 0$ implies $p_l = 0\ \forall l$

  - $D_{\mathrm{KL}}(P\|Q) \geq 0$ and equality holds iff $P = Q$ almost everywhere

- Rényi divergence of order α (or α-divergence)

$$D_{\alpha}(P\|Q) = \frac{1}{\alpha - 1} \log \left(\sum_{l=1}^{L} \frac{p_l^{\alpha}}{q_l^{\alpha-1}} \right)$$

- Jensen–Shannon divergence

$$D_{\mathrm{JS}} = \tfrac{1}{2} D_{\mathrm{KL}}\left(P\|M\right) + \tfrac{1}{2} D_{\mathrm{KL}}\left(Q\|M\right)$$

where M is the average of the two distributions, i.e.

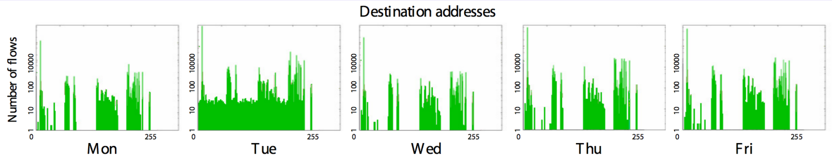
$$M = \frac{1}{2}(P + Q)$$

Chaitin-Kolmogorov entropy

- Symbols in a message are, in general, correlated and so the information carried by each symbol **is less** than the Shannon entropy of the corresponding alphabet
- The entropy of a string is the length (in bits) of the smallest program which produces as output the string
- The entropy represents a lower bound to the compression rate that we can obtain
- Different compression algorithms can be considered
  - Dictionary–based algorithms
  - Model–based algorithms (of different order)
  - Block–sorting algorithms
- The entropy can be used to detect the language of a given plain text
- The presence of anomalies should affect the entropy of the *related traffic sequence*

Outline

The main idea. . .

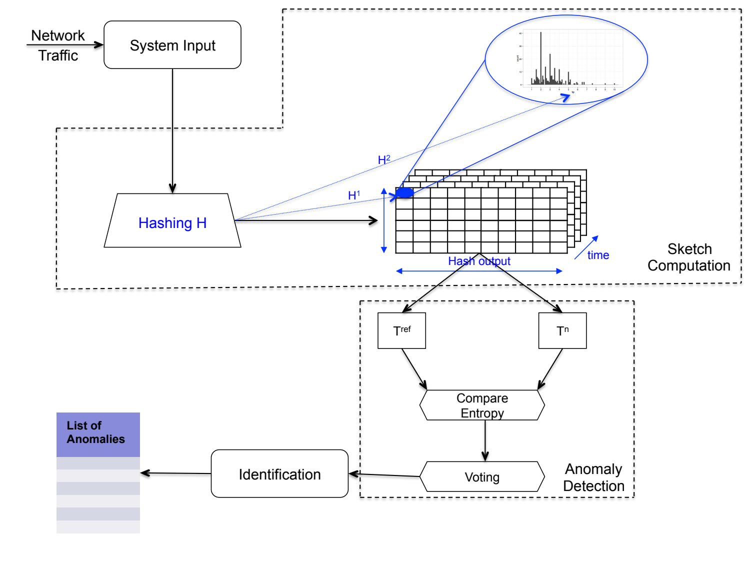


- Histograms of *significant* features should significantly change between normal and anomalous behavior
- The idea is simple, BUT:
  - which are the *significant* traffic features?
  - how to *measure* significant changes between histograms?

Histogram-Based Traffic Anomaly Detection

A. Kind, M. Ph. Stoecklin, and X. Dimitropoulos, IEEE Transactions on Network Service Management, Vol. 6, No. 2, June 2009

System Architecture



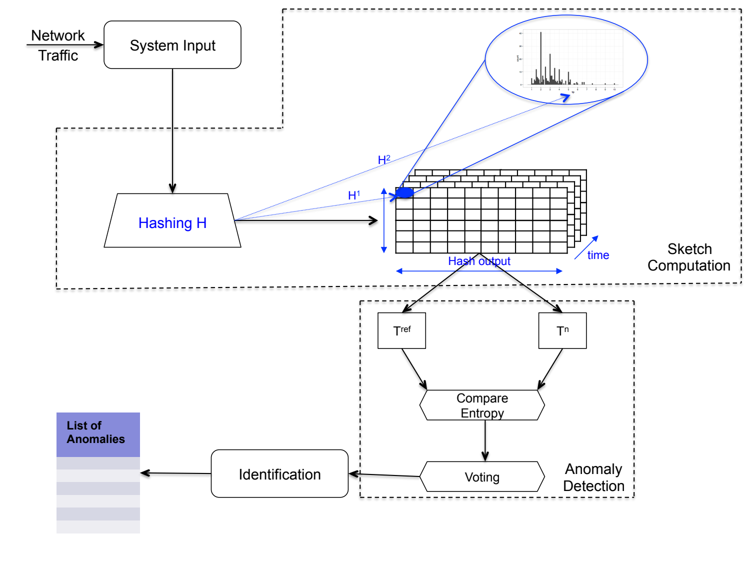
System Input

- The proposed system takes as input traffic data over a predefined time-bin
- For each time-bin the system extracts a list of keys i_t (e.g., the list of destination IP addresses) and the associated weights c_t (in our case, the number of bytes and flows for that IP address)

Turnstile Model

- Input data: $I = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$
- Item σ_k:
  - key i_k
  - weight c_k
- Underlying function: $U[i_k] += c_k$

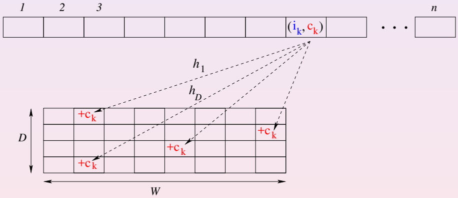
System Architecture



Sketch

Definition

Sketches are two-dimensional $D \times W$ arrays $T[l][j]$, where each row d ($d = 1, \cdots, D$) is associated to a given hash function h_d. These functions give outputs in the interval $(1, \cdots, W)$ that are associated to the columns of the array

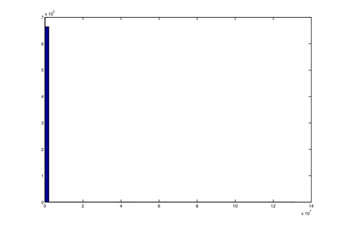


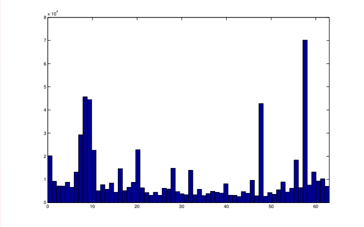
Update procedure

$$C[d][h_d(i_k)] += c_k$$

3-D Sketch

- Sketches are modified to store histograms $T[d][w][l]$
- A second hashing scheme is used to map input keys to histogram bins ⇒ *Random Histogram*
  - unknown range of relevant variables
  - possibility of highly-peaked distributions (e.g., number of bytes)
  - additional protection against “mimicry” attacks



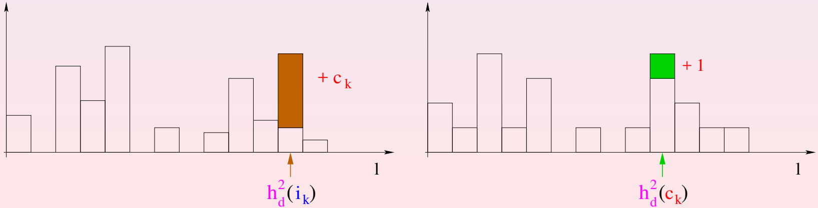


3-D Sketch

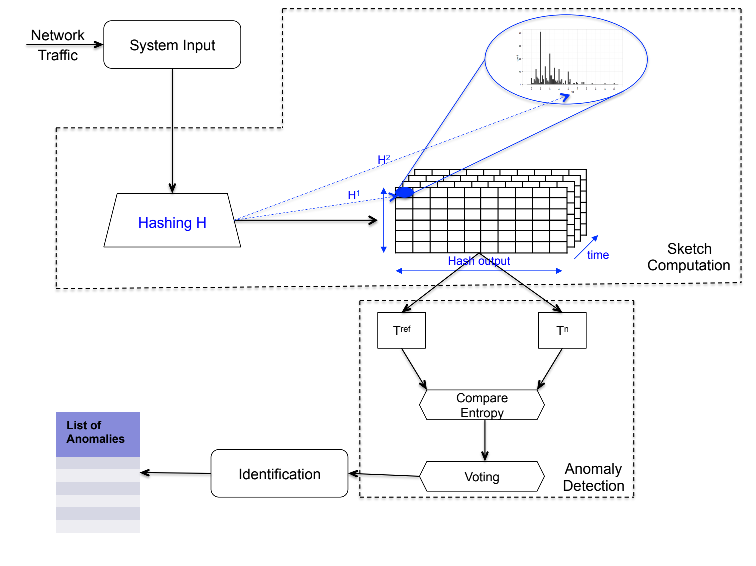
- Sketches $\in \mathbb{N}_{D\times W\times L}$, where D, W, and L can be varied (in the experimental tests $D = 16$, $W = 512$, and $L = 96$)
- Formally, for each new data, the update procedure of the sketch is described by

Method 1: $T[d][h_d^1(i_k)][h_d^2(i_k)] \leftarrow T[d][h_d^1(i_k)][h_d^2(i_k)] + c_k$

Method 2: $T[d][h_d^1(i_k)][h_d^2(c_k)] \leftarrow T[d][h_d^1(i_k)][h_d^2(c_k)] + 1$



System Architecture



Anomaly Detection

- N distinct sketches $T^n_{D\times W\times L}$, where $n \in [1, N]$ denotes the time-bin
- Inputs to the anomaly detection block
  - the current sketch T^n
  - the *reference sketch* T^{ref}, i.e. the last observed non anomalous sketch
- For each bucket $T^n[d][w][\cdot]$, the system compare the entropy of the stored histogram with the entropy of $T^{\mathrm{ref}}[d][w][\cdot]$ or calculate the divergence between the two histograms

Detection and Identification

- If entropy difference exceeds a given threshold the correspondent sketch bucket is labeled as anomalous
  - thresholds are set via Monte-Carlo simulation
  - binary matrix ($A \in \mathbb{N}_{D \times W}$) that contains a "1" if the bucket is considered anomalous
- Each traffic flow is part of D random aggregates, corresponding to the D different hash functions
- A voting algorithm is applied to the matrix A
  - if at least K rows of A contain at least one bucket set to "1", the time-bin is considered as anomalous and an alarm is generated
  - otherwise T^{ref} is updated
  - K is a tunable parameter and it has been set $K = D/2 + 1$
- Anomaly identification is performed by exploiting the reversible sketch functionalities

MAWI-Lab Traffic Traces

- MAWI (Measurement and Analysis on the WIDE Internet) archive (sample-points B and F)
- Anomaly labelling is obtained combining the output of four anomaly detectors
  - Hough transform
  - Gamma distribution modelling
  - Kullback-Leibler divergence
  - Principal Component Analysis

MAWILab : Combining Diverse Anomaly Detectors for Automated Anomaly Labeling and Performance Benchmarking

R. Fontugne, P. Borgnat, P. Abry, and K. Fukuda, ACM CoNEXT 2010

MAWI-Lab Traffic Traces

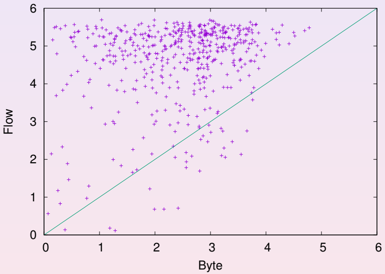
- Traffic taxonomy
  - *anomalous*: anomalous with high probability
  - *suspicious*: probably anomalous, but not clearly identified by the MAWI classification methods
  - *notice*: non anomalous, but reported by at least one anomaly detector
  - *benign*: normal
- Some information about the kind of anomaly
  - *attack*: anomalies representing a well known attack
  - *special*: anomalies involving well known ports
  - *unknown*: unknown kinds of anomalies
- Performance indicators
  - ROC (Receiver Operating Characteristics) curve: Detection Rate vs. False Alarm Rate
  - AuC (Area under the Curve)

Interpretation of the MAWILab labels

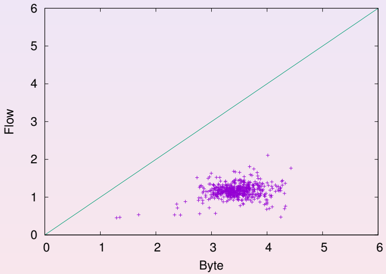
- False positives: flows that are not labeled as “anomalous” or “suspicious” in the MAWI archive
- False negatives
  - all: flows labeled as “anomalous”
  - fn 2/3/4 detector: flows labeled as “anomalous” and detected at least by two/three/four detectors
  - fn attack: flows labeled as “anomalous” belonging to the “attack” category (known attacks)
  - fn attack special: flows labeled as “anomalous” belonging to the “attack” category or the “special” category (attacks involving well-known ports)
  - fn unknown: flows labeled as “anomalous” belonging to the “unknown” category (unknown anomalous activities)
  - fn unknown 4 detector: flows labeled as “anomalous” belonging to the “unknown” category and detected by all the four detectors

Preliminary Analysis: different traffic features

Scatter Plot - H(Byte) vs. H(Flow)

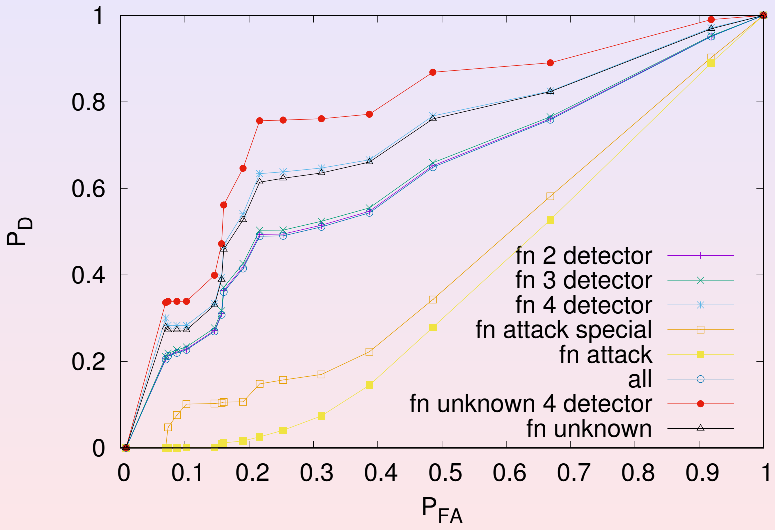


Method 1

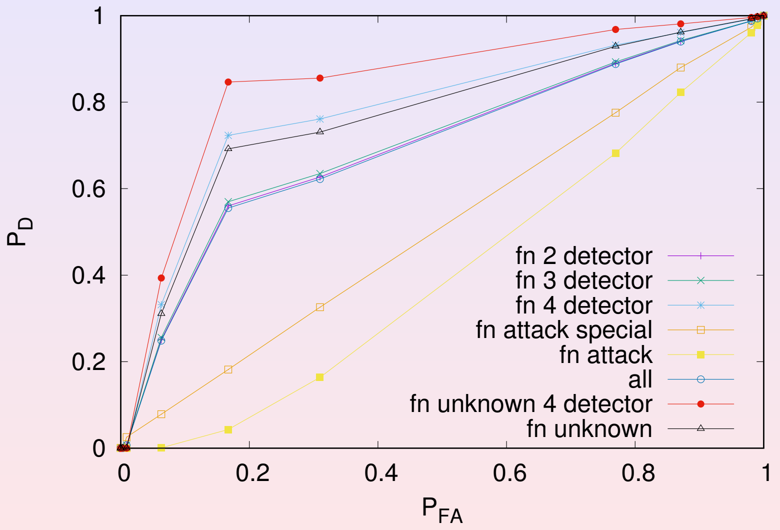


Method 2

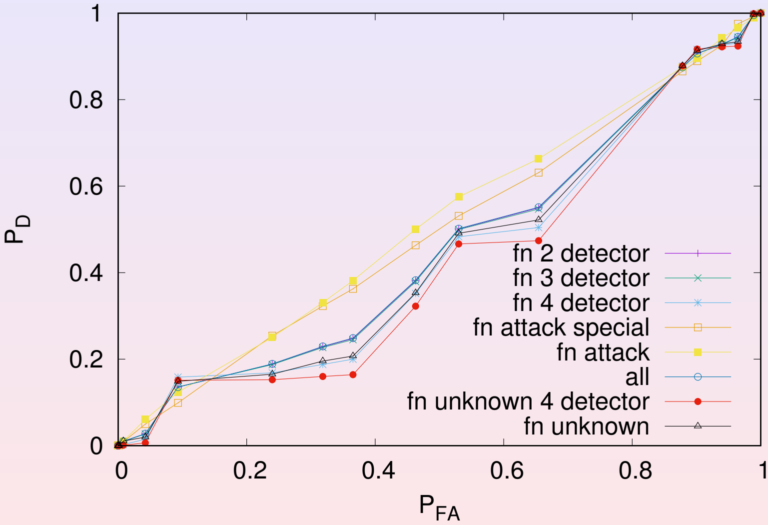
Shannon Entropy: Method 1 – Flow



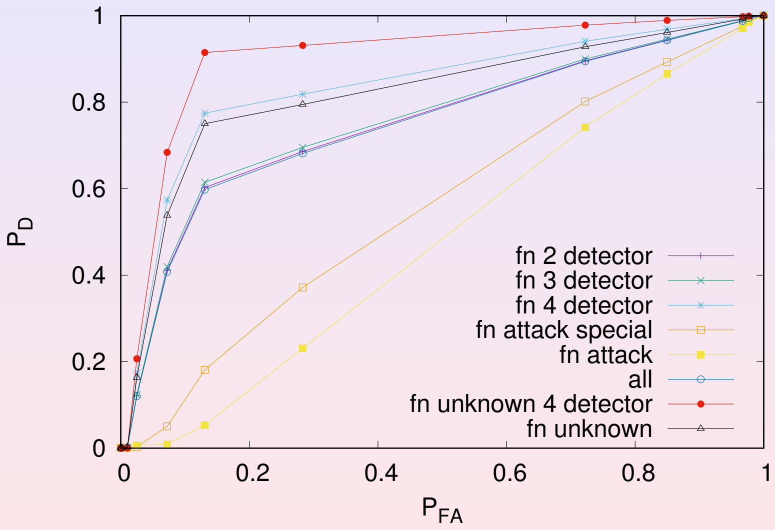
Shannon Entropy: Method 2 – Flow



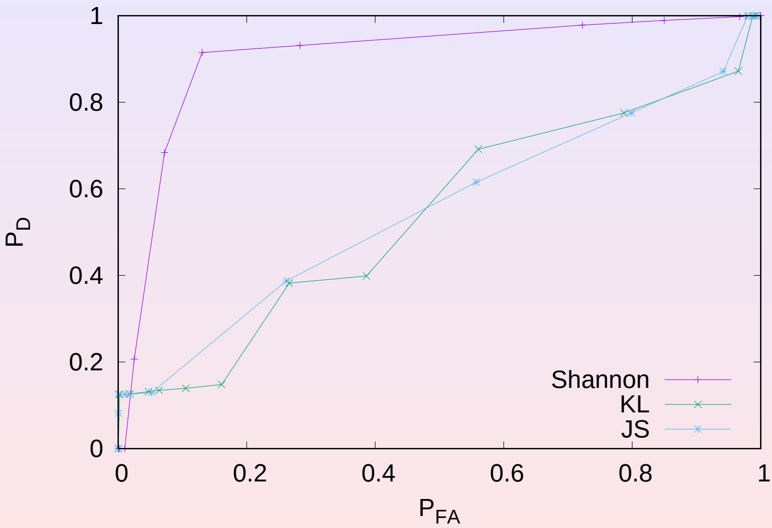
Shannon Entropy: Method 1 – Byte



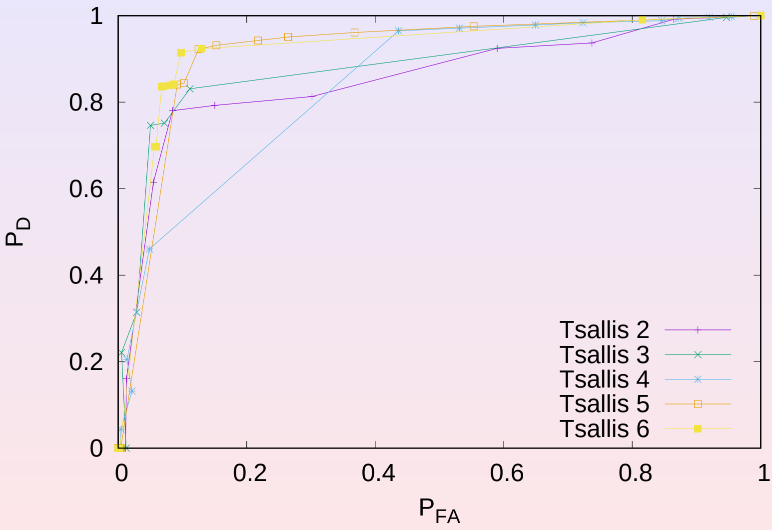
Shannon Entropy: Method 2 – Byte



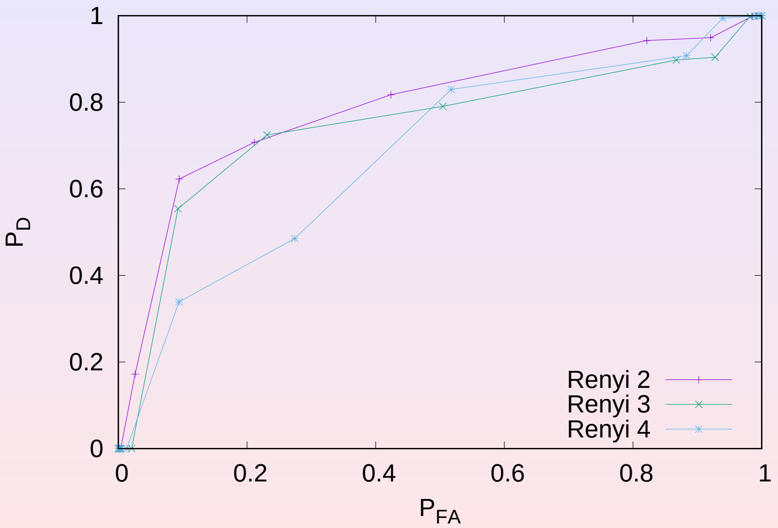
Shannon, KL, JS: Method 2 – Byte



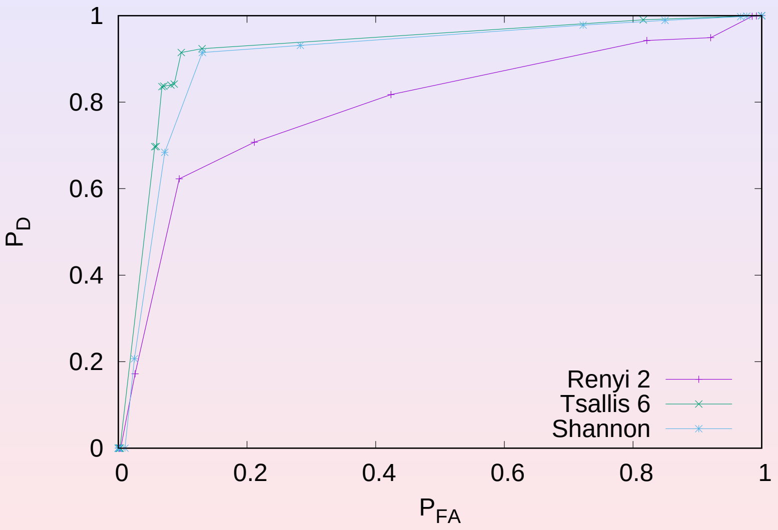
Tsallis Entropy: Method 2 – Byte



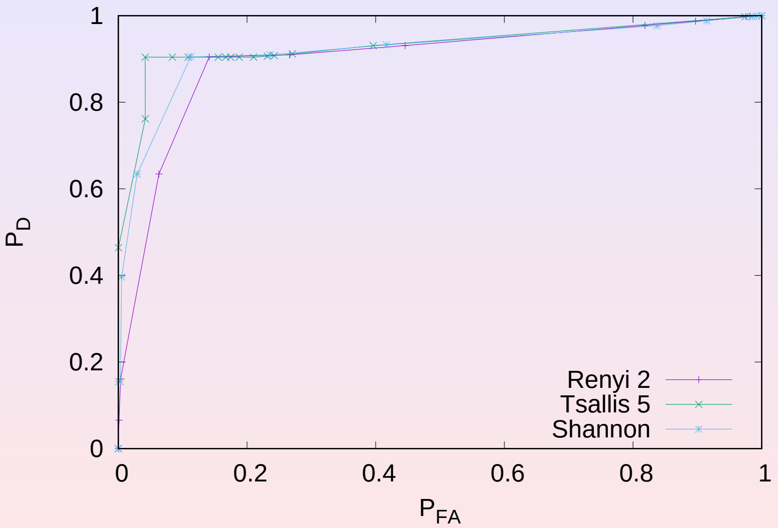
Rényi Entropy: Method 2 – Byte



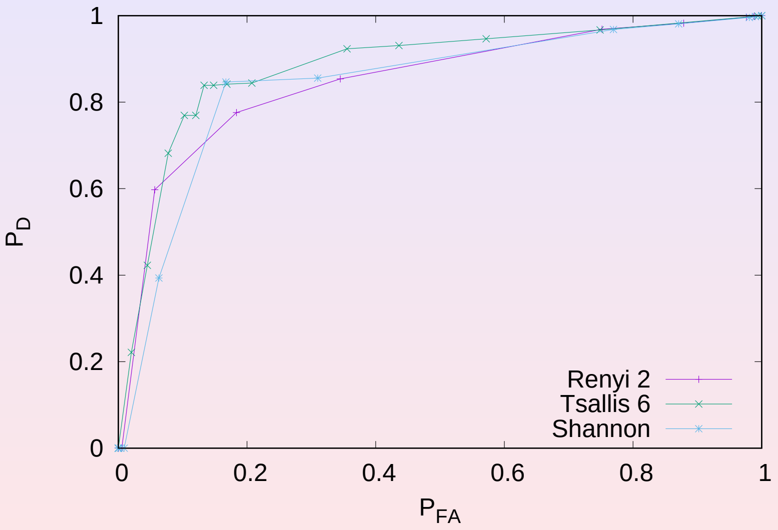
Method 2 – Byte



Method 2 – Packet



Method 2 – Flow



Outline

Related Works

Traffic Classification based on the TCP flags

A Markovian signature-based approach to IP traffic classification
H. Dahmouni, S. Vaton, D. Rossé
3rd annual ACM workshop on Mining network data, 2006

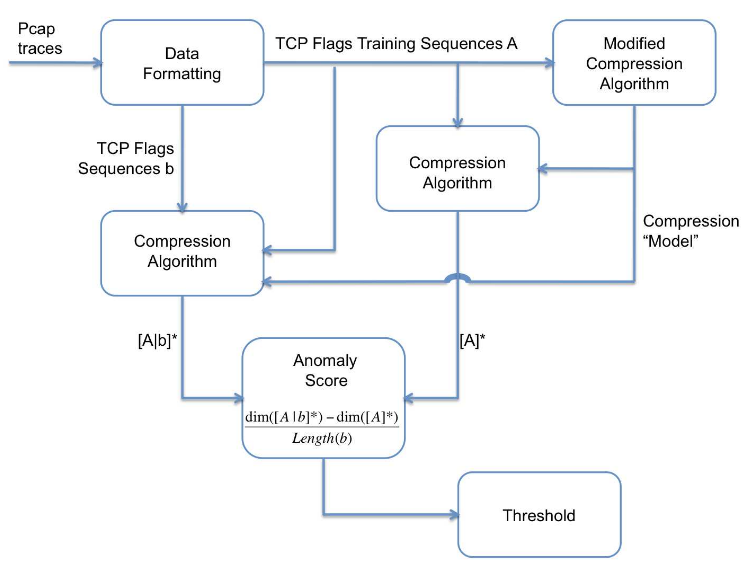
Anomaly Detection based on the TCP flags

A New Statistical Approach to Network Anomaly Detection
C. Callegari, S. Vaton, and M. Pagano
International Symposium on Performance Evaluation of Computer and Telecommunication Systems, 2008

Language Classification through entropy

Language trees and zipping
D. Benedetto, E. Caglioti, and V. Loreto
Physical Review Letters, January 2002

System Architecture



System Input

Input Data

- The system input is given by raw traffic traces in libpcap format
- Only TCP packets are passed as input to the classification block
- The 5-tuple

  (Src. address, Dest. address, Src. Port, Dest. Port, Protocol)

  is used to identify a connection, while the value of the TCP flags is used to build the *profile*
- A value s_i is associated to each packet:

  $$s_i = SYN + 2 \cdot ACK + 4 \cdot PSH + 8 \cdot RST + 16 \cdot URG + 32 \cdot FIN$$

- Each *mono-directional* connection is represented by a sequence of symbols s_i, which are integers in $\{0, 1, \cdots, 63\}$

Compression Algorithms

- Dictionary–based algorithms: based on the use of a dictionary, which can be static or dynamic, and they code each symbol or group of symbols with an element of the dictionary
  - LZ77
  - LZ78 (LZW)
- Model–based algorithms: each symbol or group of symbols is encoded with a variable length code, according to some probability distribution
  - static coders (Morse code)
  - semi–adaptive coders: the translation table has to be sent together with compressed data (static Huffman Coding)
  - dynamic coders: the translation table is directly built during encoding/decoding (Dynamic Markov compression)
- Block–sorting algorithms: a transformation of the data is performed, so as to obtain a format which can be easily compressed
  - Burrows-Wheeler transform (bzip2)

Lempel-Ziv-Welch

- Created by Abraham Lempel, Jacob Ziv, and Terry Welch. It was published by Welch in 1984 as an improved implementation of the LZ78 algorithm, published by Lempel and Ziv in 1978
- Universal adaptative (the coding scheme used for the k^{th} character of a message is based on the characteristics of the preceding $k - 1$ characters in the message) lossless data compression algorithm
- Builds a translation table (also called dictionary) from the text being compressed
- The string translation table maps the message strings to fixed-length codes

Huffman Coding

- Developed by Huffman (1952)
- Based on the use of a variable-length code table for encoding each source symbol
- The variable-length code table is derived from a binary tree built from the estimated probability of occurrence for each possible value of the source symbols
    - It expresses the most common characters using shorter strings of bits than are used for less common source symbols
    - Prefix-free code: the bit string representing some particular symbol is never a prefix of the bit string representing any other symbol
    - Although optimal among methods encoding symbols separately, Huffman coding is not always optimal among all compression methods
    - Other methods such as arithmetic coding often have better compression capability

Dynamic Markov Compression

- Developed by Gordon Cormack and Nigel Horspool (1987)
- Adaptative lossless data compression algorithm
- Based on the modelization of the binary source to be encoded by means of a Markov chain, which describes the transition probabilities between bits
- The built model is used to predict the future bit of a message
- The predicted bit is then coded using arithmetic coding
- Arithmetic coding encodes the entire message into a single number, an arbitrary-precision fraction q where $0.0 \leq q \leq 1.0$

bzip2

bzip2

- Open source, patent free, high-quality data compressor
- Higher compression rate wrt to more conventional LZ77/LZ78-based compressors
- Much more complex and considerably slower

Structure of the encoder

An input file is divided into fixed sized blocks that are compressed independently, by applying three transformations

- Burrows-Wheeler transform (block-sorting lossless transformation)
- Move-to-front transform (*continuously evolving* coding table, that is a list of all possible characters in a specific order)
- Lossless Compression (Huffman)

Training Phase

- Possibility to choose the compression algorithm
- The learning phase (i.e., building of the *normal data model*) is stopped as the training phase is over

Learning phase for the different compressors

- Huffman case: the occurrence frequency of each symbol
- DMC case: the estimation of the Markov chain used for the compression
- LZW case: the construction of the dictionary
- bzip2 case: the final Huffman phase is modified as above

- The detection phase is performed with a compression scheme that is optimal for the training data and suboptimal for the new data, especially in case of anomalous connections

Detection Phase

- Append each distinct "observed" connection b to the training dataset A
- Compute the *ratio*

$$X = \frac{\text{length}([A|b]^*) - \text{length}([A]^*)}{\text{length}(b)}$$

where $[S]^*$ represents the compressed version of S

- Choose between a single hypothesis H_0 (normal traffic) and the composite hypothesis H_1 (anomaly)

$$X \underset{H_1}{\overset{H_0}{\lesseqgtr}} \xi$$

Dataset and Performance Parameter

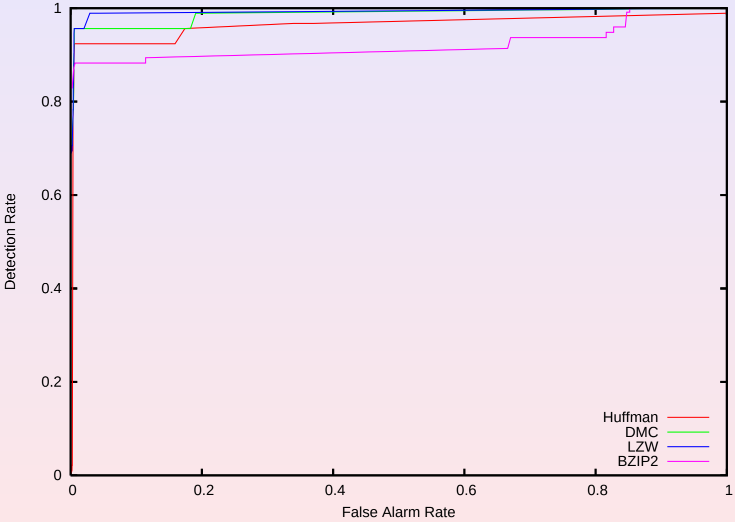
DARPA

- 1999 DARPA/MIT IDS evaluation program
- It provides a corpus of data, modelling the network traffic measured between a US Air Force base and the Internet
- 5 weeks data (several thousands connections per application)
    - week 1 and 3: used for training
    - week 4 and 5: used for detection

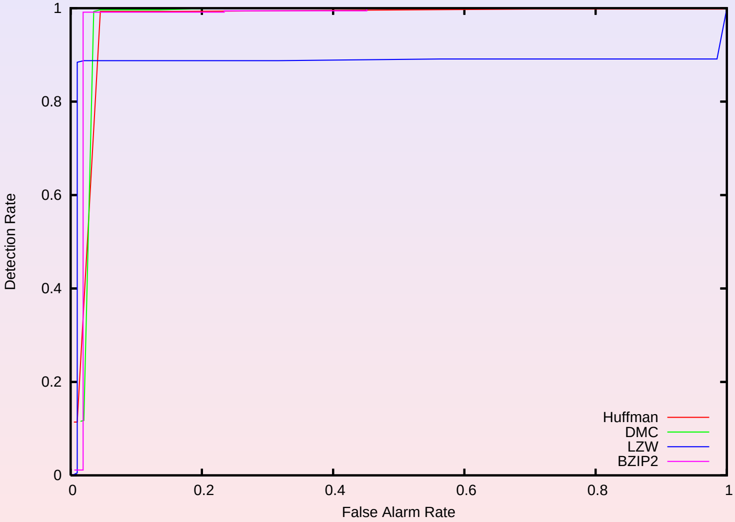
Receiver Operating Characteristic (ROC)

- 20/TCP (FTP data)
- 22/TCP (Secure Shell)
- In DARPA dataset we can trust the port number for classifying the traffic

ROC curve - 20/TCP



ROC curve - 22/TCP




СПАСИБО ЗА
ВНИМАНИЕ
И ПОПЫТКУ
ПОНИМАНИЯ