Host based Intrusion Detection

Shubham Tripathi
201407646

Introduction

- An **intrusion detection** system (IDS) is a device or software application that monitors network(Network IDS) or system activities (Host IDS) for malicious activities or policy violations and produces reports to a management station.
- Host based intrusion detection system monitors and analyzes internals of computing system.
- Current trend in HIDS is to detect intrusion based on sequences of system calls.
- Host-based anomaly intrusion detection system design is very challenging due to the high false alarm rate.

- Anomaly Intrusion Detection: Build up a profile of normal behavior for a program of interest, treating deviations from this profile as anomalies. inspired by AIS / Sense of Self.
  - Zero-Day Attacks - We do not want system to learn the signatures of attacks as there can be totally new sequence of attack sequence possible
  - False Alarm Rate - High due to the difficulty of creating a robust baseline.
- Misuse Intrusion Detection: Learns signatures of attacks (eg. AVS)
  - Zero-Day Attacks - Incapable of detecting a totally new attack sequences.
  - False Alarm Rate - Low as robust baseline can be built.

Dataset:

- ADFA-LD: This dataset (2012) uses Ubuntu-12 operating system and the most recent publicly available exploits and methods .
- Suited for Anomaly IDS.

THE COMPOSITION OF ADFA-LD

Normal	
# of Training traces	833
# of testing traces	4373
Total attacks	
# of attacks	60
# of attacks traces	686

Experiment 1: Sliding Window Comparison

- Scan traces of normal system calls and build up database of all unique sequences of length k.

  open, read, mmap, mmap, open, read, mmap

  For k=3

  open, read, mmap

  read, mmap, mmap

  mmap, mmap, open

  mmap, open, read

- Space Complexity: O(N*M*k) , {N seq of avg length M}
- Training sequences: 2600, Database Unique Traces = 87829 (k=5)

- Normal Trace: `open, read, mmap, mmap, open, getrlimit, mmap, close`

call	position 1	position 2	position 3
open	read, getrlimit	mmap	mmap, close
read	mmap	mmap	open
mmap	mmap, open, close	open, getrlimit	getrlimit, mmap
getrlimit	mmap	close	
close			

Testing:

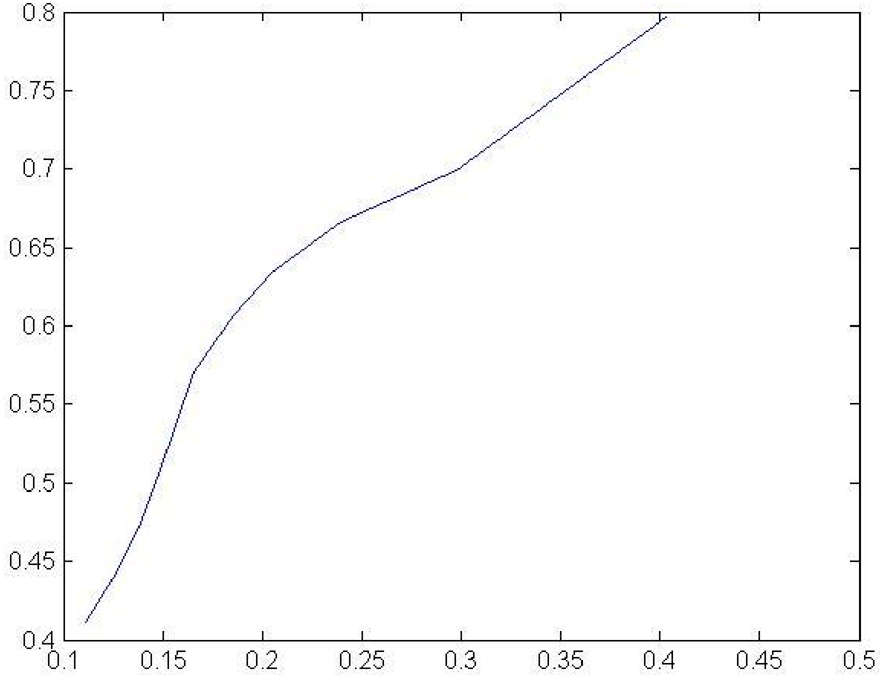
New trace: open, read, mmap, open, open, getrlimit, mmap, close

- Measuring Anomalous Behaviour:
    - Generate unique sequences of length k from the test sequence.
    - Compare against database of normal profile and Compute number of Mismatches as follows:
        - For each seq. i, Mismatches += 1 if no seq in DB starting with same system call(s) matches with i.
        - Total mismatches = Sum(Mismatches for all i)
    - Classify as Anomalous if total mismatches exceeds a threshold.
- open, read, mmap, mmap, open, *mmap*, mmap -> 2 mismatches

    mmap, open, *mmap*

    open, *mmap*, mmap

- Time Complexity: O(N*M*k), N=size of test seq, M=#sys calls starting with s (maximum=N), k=size of window



Detection: FPR = .79: .39
best for K = 5

Experiment 2: Bag of Words Approach

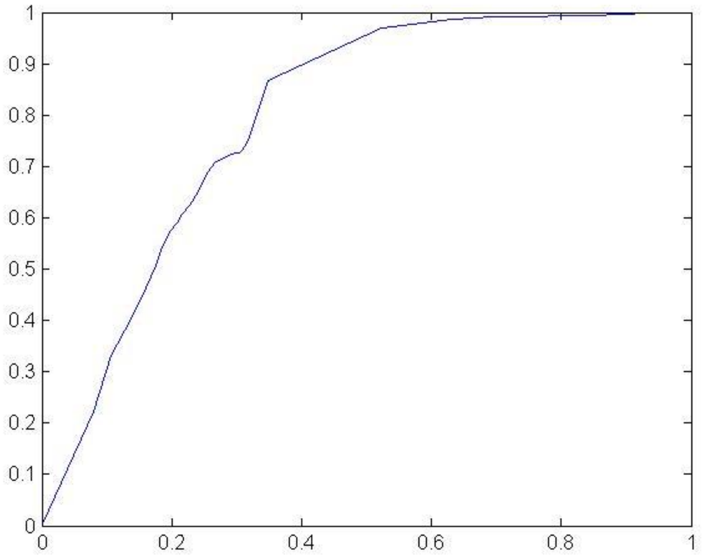
- Bag of System Calls:
    - Bag consists of unigrams and bigrams of system calls. Bigrams are required to capture the *contiguity* of system calls in normal training data.
    - Feature Vector: Each sequence is converted to vector of term frequency of unigrams and bigrams.
        - Size(M) = N + N*N , N = #unique sys calls(340)
    - #Parameters = M (Using Naive Bayes Assumption)
    - P(Sk | Normal) = (1+N(Sk, Normal)) / (M + Sum(N(Si, Normal)))

- Association Rules (Capturing Discontiguity)
    - S = set of system calls, D = set of normal sequences
    - A rule is defined as implication: X -> Y where X,Y belongsTo S, such that X AND Y = NULL
    - Support = P(X, Y) = N(X,Y) / N(D)
    - Confidence = P(Y | X) = P(X,Y) / P(X) = N(X,Y) / N(X)
- Apriori Algorithm is used to capture rules corresponding to high support and confidence.
- Additional Features for BoW: Unlike bigrams which are contiguous, this approach will capture discontiguous system calls that are most likely to occur in Normal Data.

- Feature Vector(F) => [Unigrams (U), Bigrams (B), Rules(R)]

  [P(Si) , P(Si,S(i+1)), P(X,Y)]
- Maximum Likelihood Estimate:
  - Let T be a test sequence,
  - create feature vector F from T consisting of term frequencies of U and B but fixed value for each R (Association Weight set as 100 (denial of service consists of large number of same calls))
  - Likelihood(params) = Sum (TF (Fi) * log(P(Fi)))
    - if Likelihood > threshold => Normal
    - else Attack
  - #Parameters = |U| + |B| + 2

Result



Detection: FP Rate = .85 : .33
Training : Test = 7:3
Min Support Value = 70%
# of Rules learnt = 107

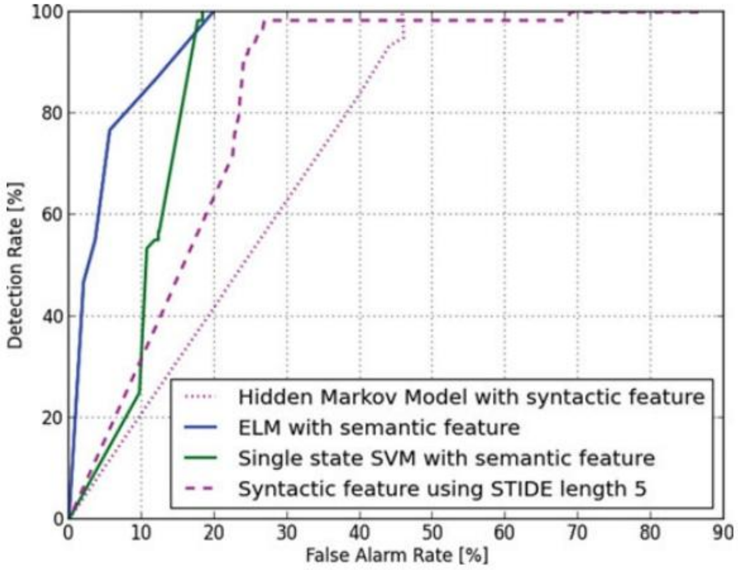
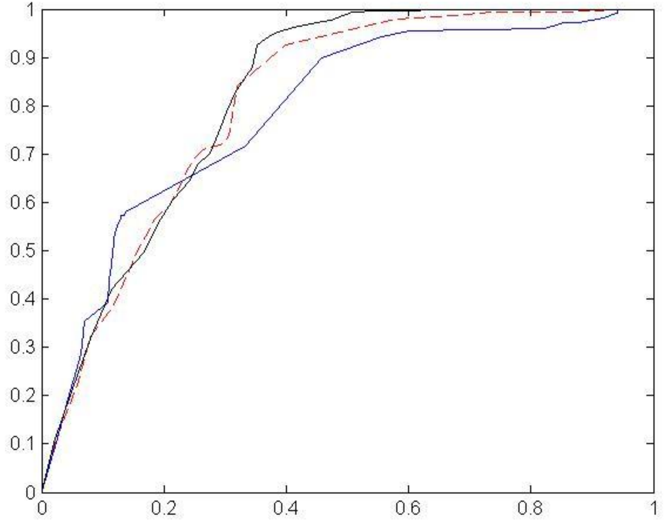
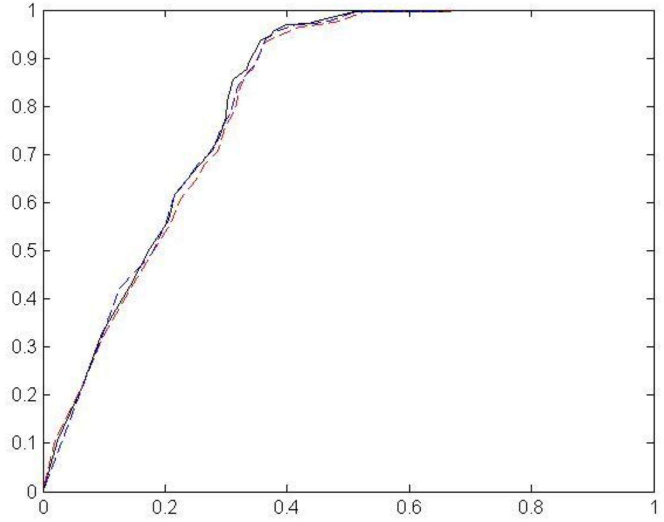


Fig. 2. ROC curves for assorted methodologies when assessing the ADFA-LD.

Varying Parameters



Train = 0.7, Support (Blue) = 0.85
(Red) = 0.7
(Black)= 0.5 (max Area)
Confidence = 0.7



Support = 0.5, Train= 0.5, 0.7, 0.9
Confidence = 0.7

Baseline Comparison

TABLE 3
Comparison between Contemporary IDS Algorithms

Algorithm	Detection Rate [%]	False Alarm Rate [%]
Data mining of audit files [60]	80.2	Not cited
Multivariate statistical analysis of audit data [33]	90	40
HMM and entropy analysis of system calls [61]	91.7	10.0
System call n-gram sliding window (assorted decision engines) [46]	$95.3 < DR < 96.9$	~ 6.0
RBF ANN analysing system calls [31]	96 mean	5.4 mean
MLP ANN on subset of KDD98 [62]	99.2	4.94
SVM on subset of KDD98 [62]	99.6	4.17
kNN with Smooth Binary Weighted RBF [63]	96.3	6.2
Rough Set Clustering [64]	95.9	7.2
ELM using original semantic feature proposed in this paper	**100.0**	**0.6**

Adapted from "CREECH AND HU: A SEMANTIC APPROACH TO HOST-BASED INTRUSION DETECTION SYSTEMS" 2014

END