Detecting Intrusions Using System Calls: Alternative Data Models

Enrico Galli

Outline

- Purpose
- Choosing Applicable Methods
- Data Sets
- Experimental Design
- Building models of normal behavior
- Results
- Conclusions

Purpose

- System-call sequences for each program.
- Compare the ability of different statistically-based learning techniques to recognize intrusion.

Choosing Applicable Methods

- Enumerating Sequences
- Frequency-based methods
- Data mining approaches
- Finite State Machines

Enumerating Sequences

- Enumerating sequences that occur empirically in traces of normal behavior.
- Monitor for unknown patterns.
- lookahead pairs: A list for each system call of the system calls that follow it at a separation of 0, 1, 2, up to *k* system calls.
- sequence time-delay embedding(stide): contiguous sequences of fixed length

Frequency-based methods

- Frequency distributions of system calls in a sequence
- n-gram vector
  - Not suitable for on-line testing(program needs to terminate to calculate trace vectors)
  - Difficult to determine the vector size
  - Course clustering is not precise enough

Frequency-based methods(cont.)

- Helman and Bhangoo method
  - Expected frequency of sequence in normal vs intrusion
  - Frequency of sequence on all intrusions is unknown.(we must guess)
  - Invalid assumptions: data are independent and stationary
    - Sequences of system-calls are not stationary within traces
    - Sequences of system-calls are not independent

Frequency-based methods(cont.)

- SRI(Emerald system)
  - Compares short-term frequency distributions from new, unknown traces with the longer-term historical distribution.
  - Prior knowledge of abnormal frequencies not required.
  - The long-term distribution can be continually updated.
  - Intruder could shift the definition of normal towards intrusive behavior.

Frequency-based methods(cont.)

- Which one did they implement/test?
  - None of the above
- “Central to both methods is the idea that rare sequences are suspicious.”
- “We chose to implement a minimal version of a frequency-based method that would allow us to evaluate this central idea.”

Data mining approaches

- From a large data set, determine what features are most important.
- Compact definition of normal
- RIPPER
  - Small set of rules that captures common elements
  - Anything that violates the rules is anomalous.

Finite State Machines

- FSM to recognize the “language” of trace
- Determine the frequencies with which system calls occur conditioned on some number of previous system calls.
- Hidden Markov model
  - Computationally expensive
  - Very powerful

Data Sets

- Traces were collected from programs running on live production environments
- Different very different programs selected
  - lpr (data in MIT and UNM), named, xlock, ps, inetd, stide, and sendmail

Data Sets

Program	Intrusions	Normal data available		Normal data used for training		Normal data used for testing	
	Number of traces	Number of traces	Number of system calls	Number of traces	Number of system calls	Number of traces	Number of system calls
MIT `lpr`	1001	2,703	2,926,304	415	568,733	1,645	1,553,768
UNM `lpr`	1001	4,298	2,027,468	390	329,154	2,823	1,325,670
`named`	2	27	9,230,572	8	677,340	12	7,690,572
`xlock`	2	72	16,937,816	72	778,661	1	16,000,000
`login`	9	12	8,894	12	8,894	–	–
`ps`	26	24	6,144	24	6,144	–	–
`inetd`	31	3	541	3	541	–	–
`stide`	105	13,726	15,618,237	150	246,750	13,526	15,185,927
`sendmail`	–	71,760	44,500,219	4,190	2,309,419	57,775	35,578,249

Table 1. Amount of data available for each program. “Normal data used for training” refers to models built with sequence length six; sequence length ten models used more training data. The same test data were used for both sequence lengths; this includes all normal data not used for training either set of models.

Experimental Design

- Accurately detect normal and intrusion data.
- Combine results across all available programs to get a better picture of tradeoff between false positives and negatives on multiple intrusions.
- Threshold for how much training data to use.
- Intrusions are detected when the anomaly signal exceeds a certain threshold.
- The false positive rate is the percentage of decisions in which normal data was flags as anomalous.

Experimental Design

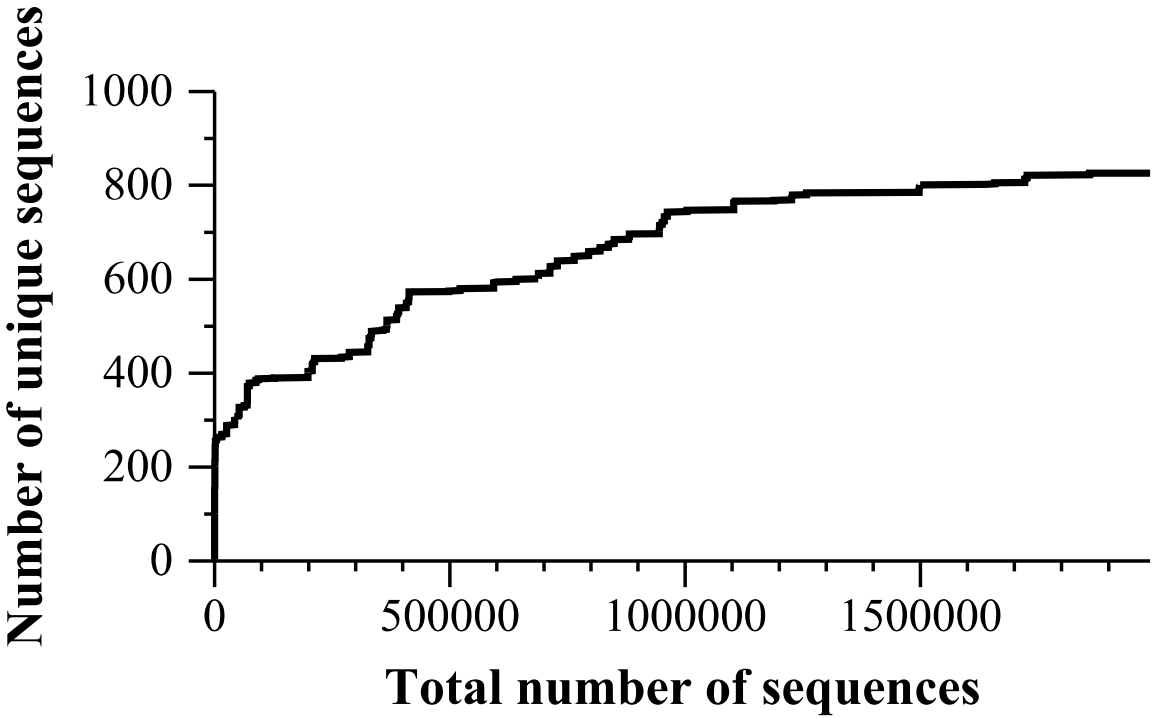

Figure 1. Typical database growth curve. The graph shows how the size of the normal database grows as traces are added chronologically.

Experimental Design

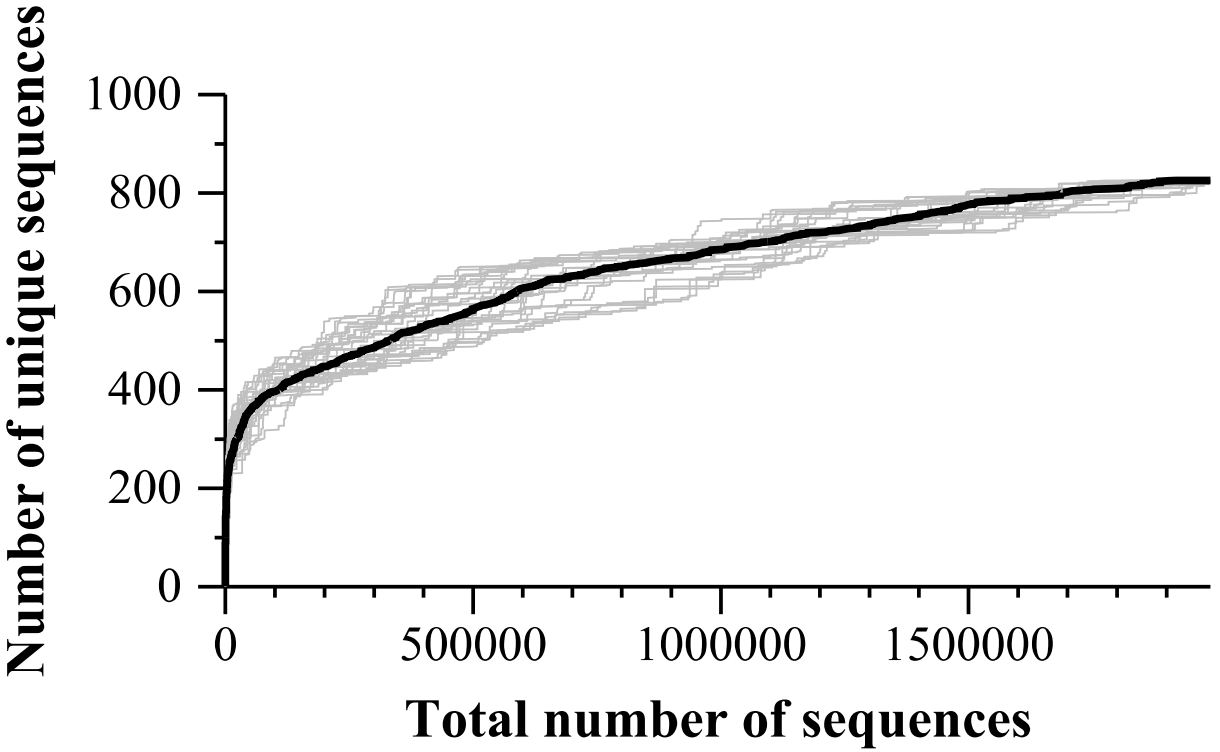

Figure 2. Alternate database growth curves for the same data used in Figure 1. Light lines show standard growth curves for different starting points in the training data; the dark line shows the mean.

Building models of normal behavior

- sequence time-delay embedding (stide)
- stide with frequency threshold (t-stide)
- RIPPER
- Hidden Markov Model

sequence time-delay embedding (stide)

- Sequence lengths of six and ten were used
- Sequence length six used a sliding window across each trace
- During testing, sequences in the trace are compared with the sequences on the database
- Mismatch sequences are considered anomalous
- Checked how many mismatches occurred within a locality frame(20 system calls).

stide with frequency threshold (t-stide)

- Rare sequences are suspicions
- In addition to mismatches, “rare” sequences(occurring less that 0.001% of the normal training data) are also counted as anomalous
- Uses Locality Frame Count as regular stile

RIPPER

- Training data: a list of all unique sequences
- RIPPER generates a list of rules that describe normal sequences
- Violation scores represents how often the rule was correctly applied on the training data.
- *High-confidence* rules are those that have a violation score greater than 80
- Violations of *high-confidence* rules count as mismatches
- Uses a Locality Frame Count as stide

Hidden Markov Model

- Numbers of states: number of unique system calls
- Training was expensive: multiple passes over training data, took approximately 2 months
- Read one system call, record transitions and outputs
- Normal traces should only require likely transitions and outputs.
- LFC was not used. Individual mismatches were used.

Results

- False-positive rates should be well below 0.001
- Results across modeling methods on a particular data set are closer than results for the same method across different data sets.
- t-stide consistently performs worse than the other methods
- No best choice for all data sets.
- Results dependent on training set

Results

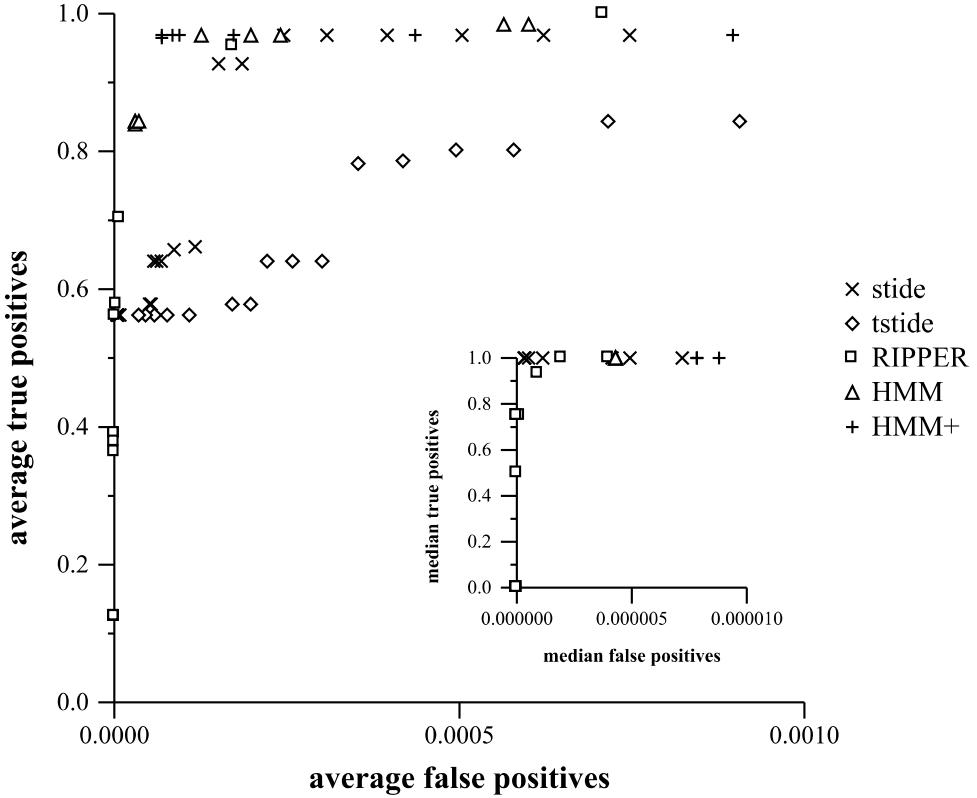

Figure 3. Composite results for each method on all data sets, sequence length 6. Each point represents performance at a particular threshold. True-positive values are the fraction of intrusions identified. For the sequence-based methods, false positives are the fraction of sequences giving mismatches at or above the specified locality frame count threshold. For HMMs, false positives are the fraction of system calls corresponding to state transitions or outputs below the specified probability threshold. Points labeled "HMM" are for only randomly-initialized HMMs, while those for "HMM+" use the specially-initialized HMMs designed to handle lpr data. No t-stide points appear in the median plot because the false positives are off the scale. Results for four HMM thresholds all map to the single median point shown.

Results

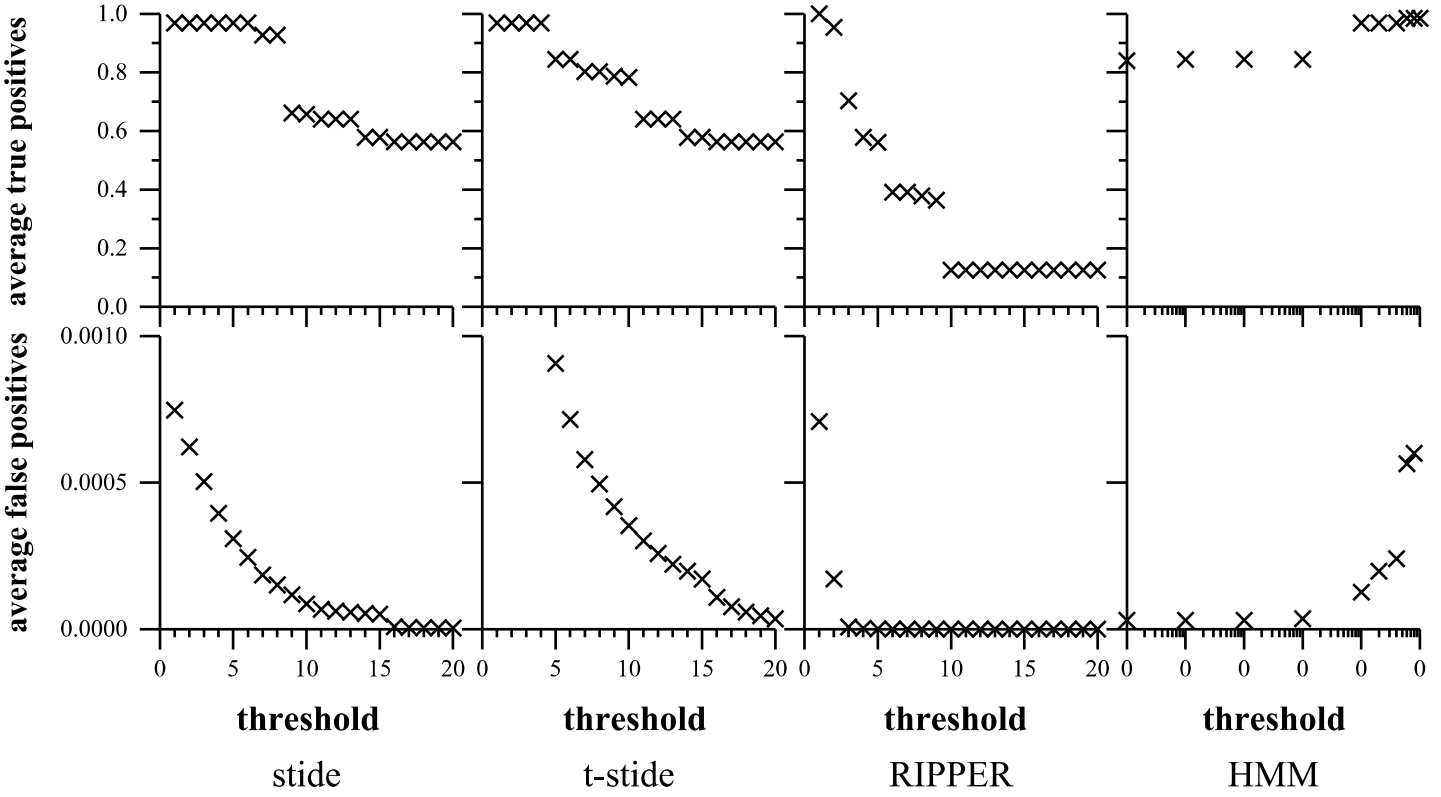

Figure 4. Average true and false positives versus threshold for each method, sequence length 6. HMM results are for randomly-initialized HMMs only.

Results

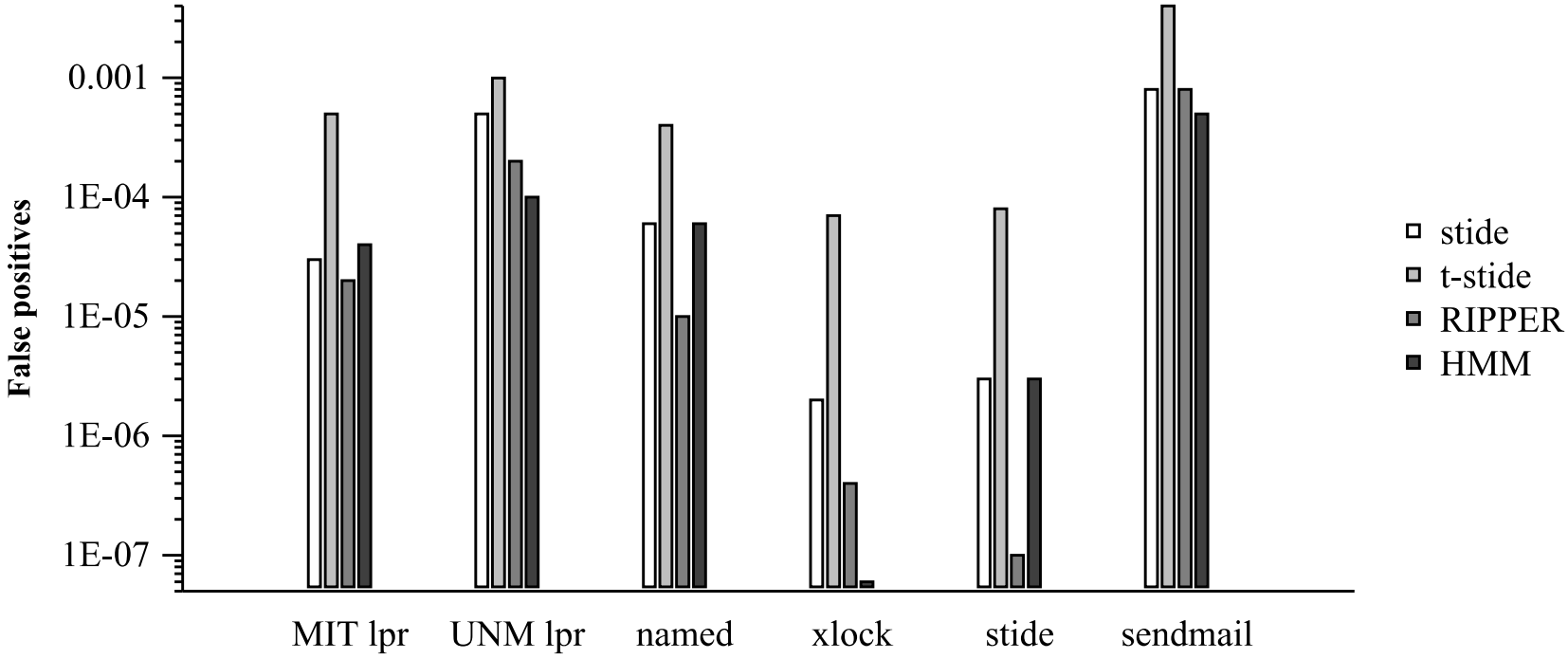

Figure 5. False-positive rates for each of six data sets, sequence length six. stide threshold: 6, t-stide threshold: 4, RIPPER threshold 2, HMM threshold = 0.001. Note that the RIPPER true-positive rate at this threshold is is slightly lower than those of the other methods. False-positive rates are shown on a logarithmic scale.

Conclusions

- Three of the four methods performed adequately.
- HMM gave the best accuracy on average, but at a high computational cost.
- Results between programs varied more than results between methods.
- More time should have be spent deciding which is the most effective stream to monitor.

Question?