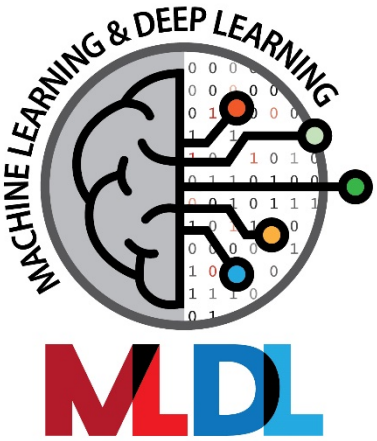
SAND2017-8429C



MLDL

Machine Learning and Deep Learning Conference 2017

An Evaluation of Sequence Learning Methods for Detecting Malware Using System Call Traces

Joe Ingram (9365)

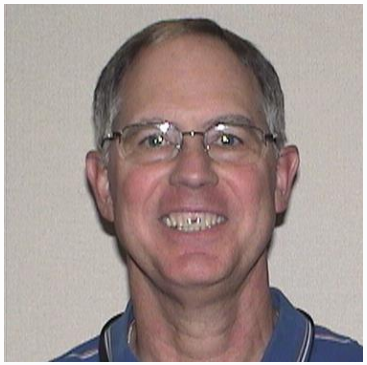
Tim Draelos (6362) , Chris Lamb (8851), Mike R. Smith (5851)

U.S. DEPARTMENT OF ENERGY

National Nuclear Security Administration

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.SAND NO. 2015-XXXXP

Acknowledgement



Tim Draelos

Chris Lamb

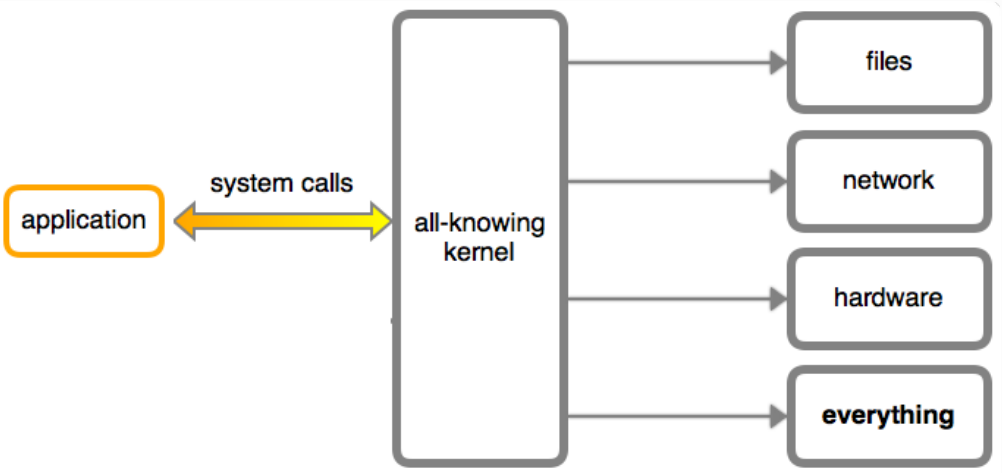
Mike R. Smith

Detecting Malware through Behavior

- **Problem:** most malware detection uses static analysis (e.g., byte signatures) that are brittle and easy for the adversary to subvert.
- **Solution:** behavior (i.e., intent) is harder to mask. Look at what the program is *doing* to infer intent.
- **Hypothesis:** infer program behavior by modeling system calls using sequential analysis techniques.

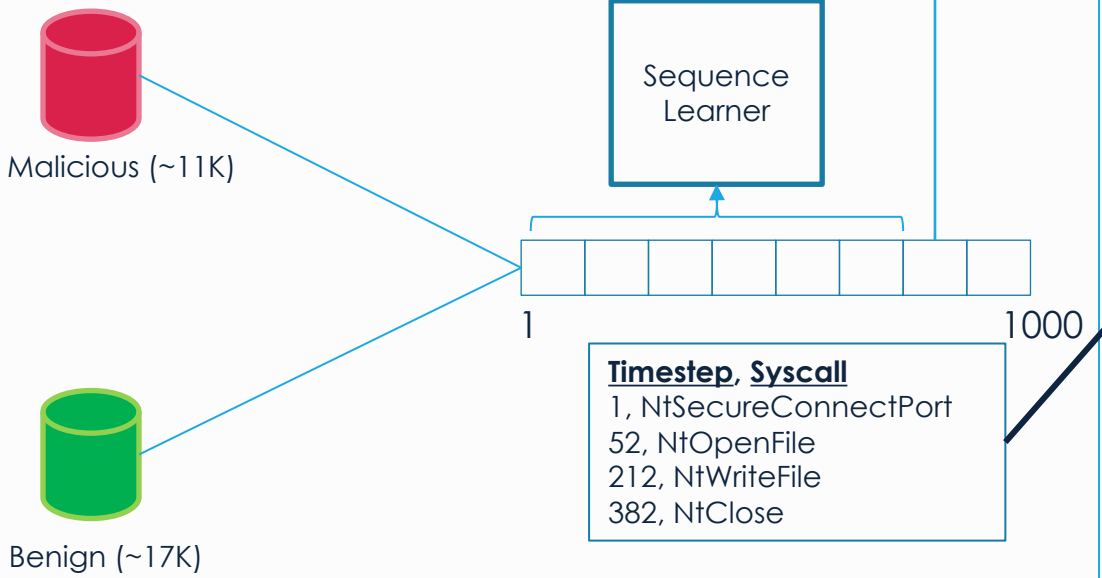
What are system calls?

- the standardized, well-defined pathways for programs to interact with the operating system.
- Programs use system calls to request and manipulate computer resources controlled by the operating system, including files and connections.

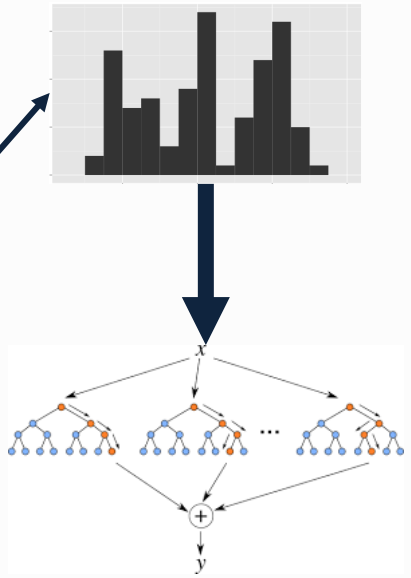


System Call Dataset

- The data was collected for several months in 2012. Malware samples from Arbor Networks.
- All samples are Windows 32-bit executables.
- Each sample was executed in a hypervisor environment for a specified period of time and the system calls made by the spawned process were collected.



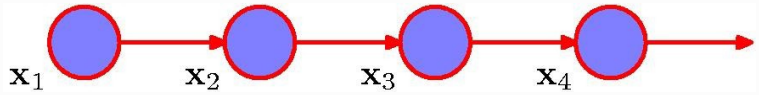
Baseline: Histograms + ML



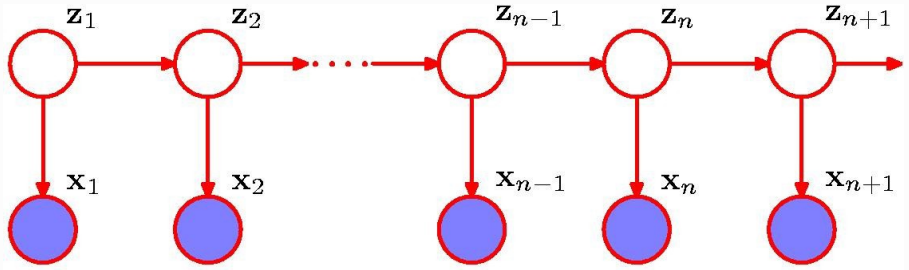
Hidden Markov Models (HMMs)

- Motivation: Observations are NOT independently and identically distributed but one state affects the next
- Markov Chain:
  - Markovian assumption: the n^{th} observation is influence only by the $n-1^{th}$ observation

$$p(x_n|x_1, x_2, \dots, x_{n-1}) = p(x_n|x_{n-1})$$

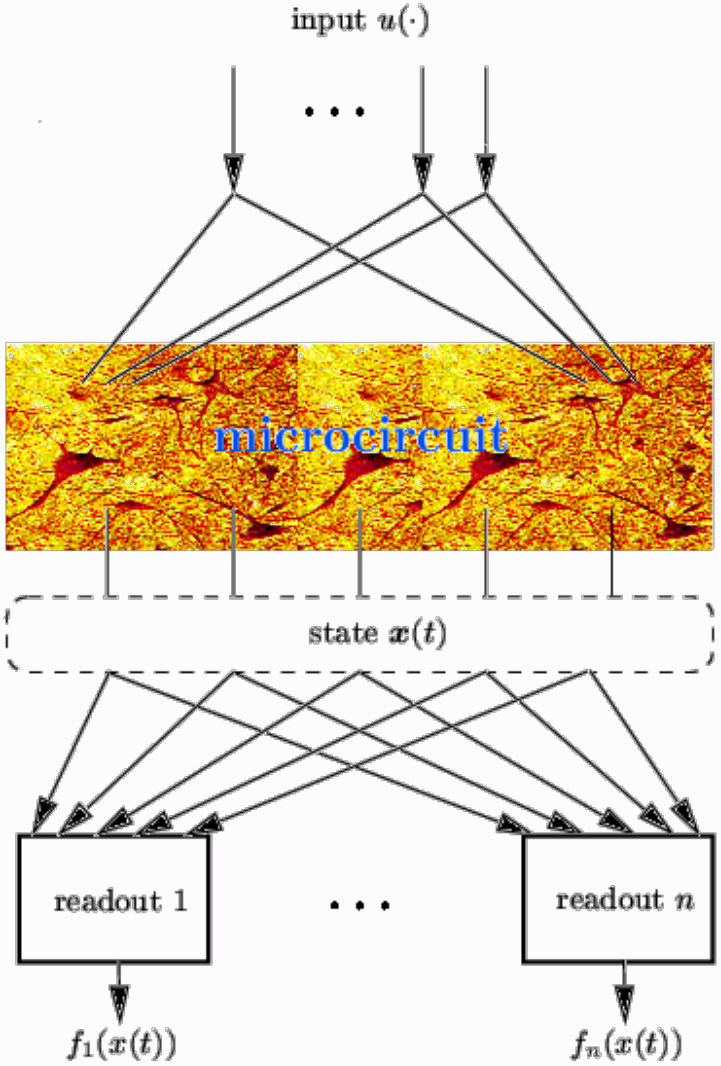


- Hidden Markov Chain:
  - The observations are influenced by a latent variable and the latent variables form a Markov chain



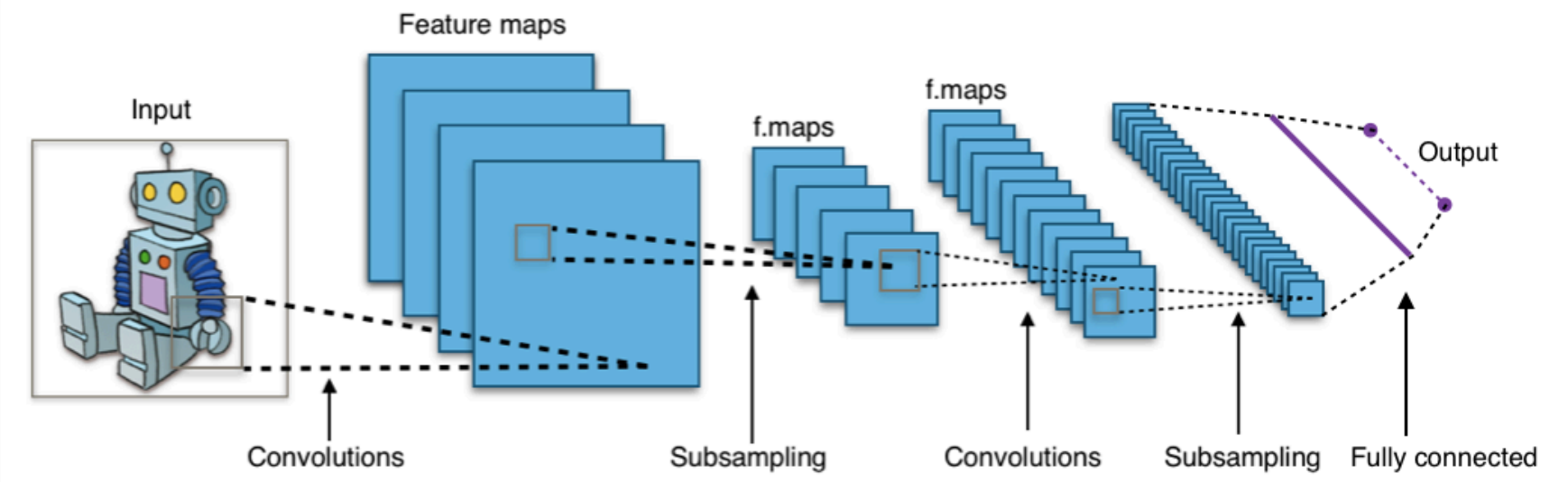
Liquid State Machines (LSMs)

- Input (spike trains)
  - Maps input streams to output streams
- Liquid (or microcircuit)
  - A recurrent neural network of spiking neurons (leaky integrate and fire)
  - Acts a preprocessor (temporal)
- State
  - Measure the state of the liquid at any given time t
- Readout neurons
  - Plastic synapses
  - By assumption, has no temporal integration capability of its own



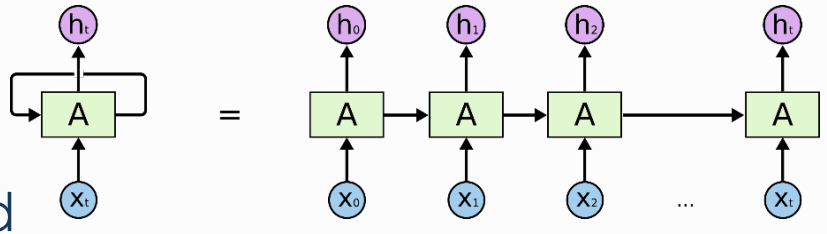
Convolutional Neural Networks (CNNs)

- Deep feed-forward neural network that learns patterns of local structure in the input signal.
- Hidden layers are either convolutional, pooling or fully connected.
- Subsequent convolutional layers learn combinations of features detected in previous layers.

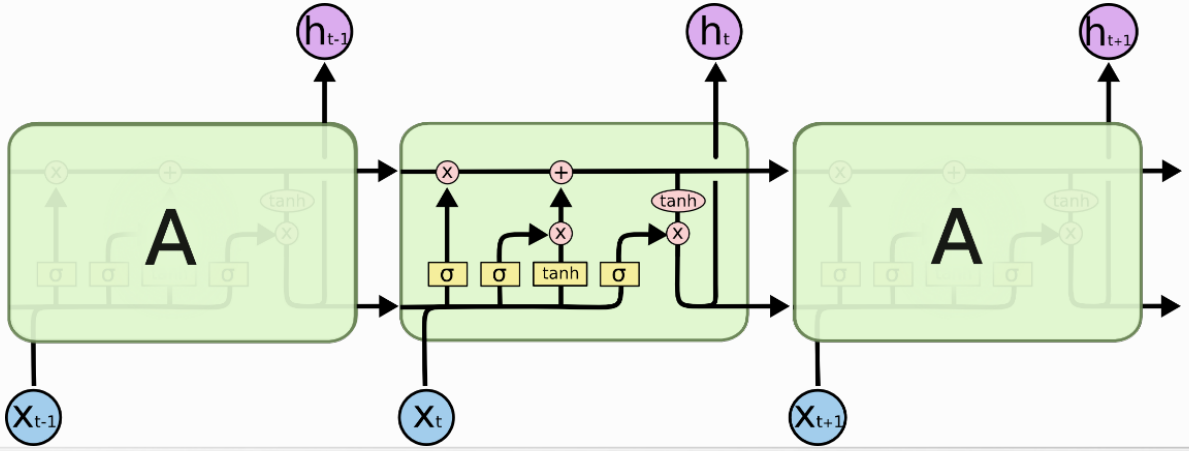


Long Short Term Memory (LSTMs)

- Recurrent Neural Network
  - Long term dependencies
- Traditional RNN uses tanh or sigmoid



- LSTMs have various different gates to maintain cell state
  - Forget gate: looks at h_{t-1} and x_t to decide what to throw away
  - What new information to keep
  - Combine this to update new cell state
  - Output is a combination of cell state and input



Results

Method	Goodware Accuracy	Malware Accuracy	CAA
Histograms + Random Forest	97.14%	92.47%	94.8%
Liquid State Machine (LSM)	91.24%	100%	95.6%
Convolutional NN (CNN)	95.26%	97.78%	96.52%
Long Short-Term Memory (LSTM)	97.4%	90.1%	93.75%
CNN + LSTM	96.32%	100%	98.16%

- Class-averaged accuracy (CAA) – average of per-class accuracy (more useful in the presence of skew).

- Example:
  - 99% of samples are goodware (1% malware).
  - Majority-class classifier ("everything's good")
    - Accuracy = 99%
    - CAA = 50%

Conclusions

- Static analysis for malware detection is easy to subvert. Behavior is harder to mask.
- Advanced sequence analysis techniques (e.g., deep learning) show promise in detecting malware behavior from system call traces.
- Better malware detection which is (hopefully) more robust to subversion.