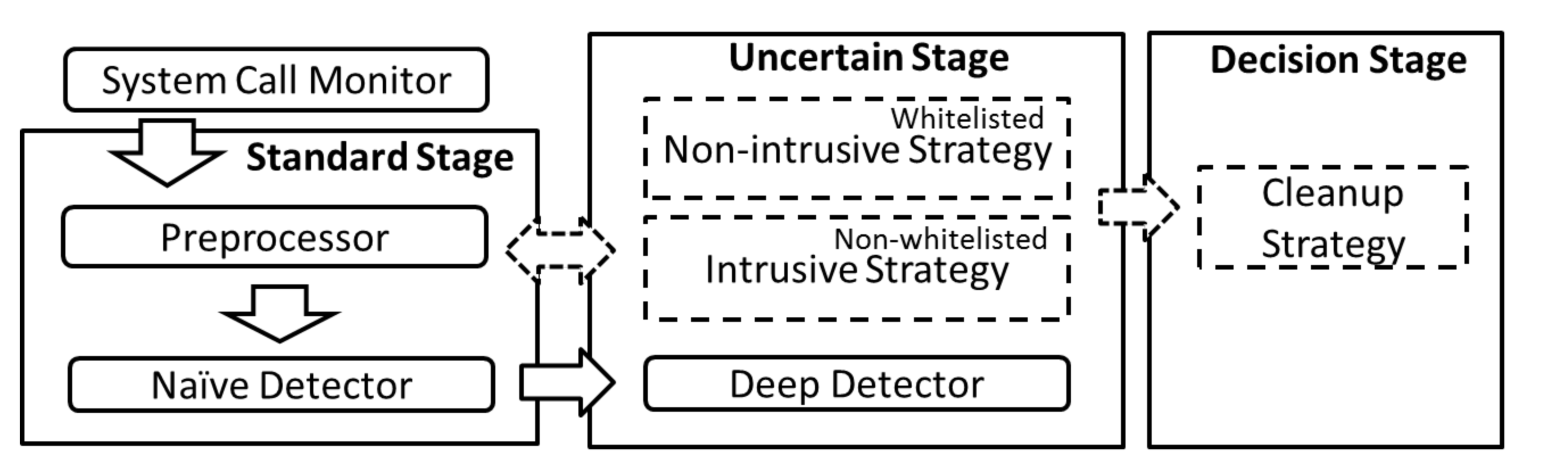
DeepMalware: Deep Models and Mechanisms for Malware Detection and Defense

PIs: Xiaolin Li, Daniela Olivera, Andre Gregio
Students: Xiaoyong Yuan*, Ruimin Sun*, Qile Zhu, Aokun Chen
University of Florida, Federal University of Parana
* equal contribution

Background

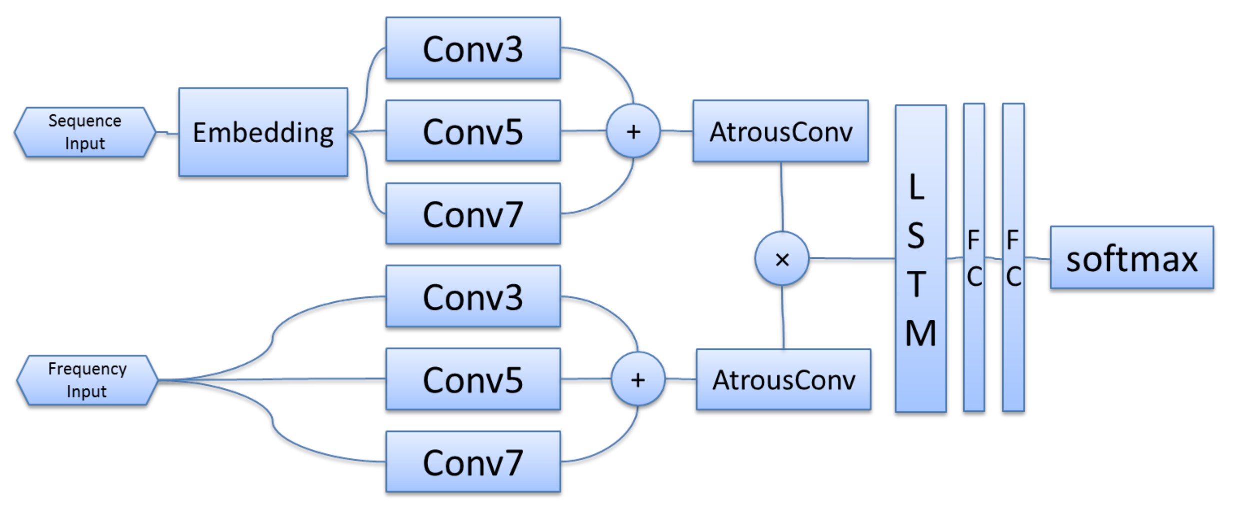
- **Conventional Malware Detection**
  - **Signature-based**
    - efficient/accurate for known attacks
    - evaded by poly/metamorphism
  - **Behavior-based**
    - address zero-day attacks
    - based on syscall sequences
- **Drawbacks** in conventional behavior detection:
  - false positives
  - cannot handle APTs
  - limited datasets for training

DeepMalware Approach

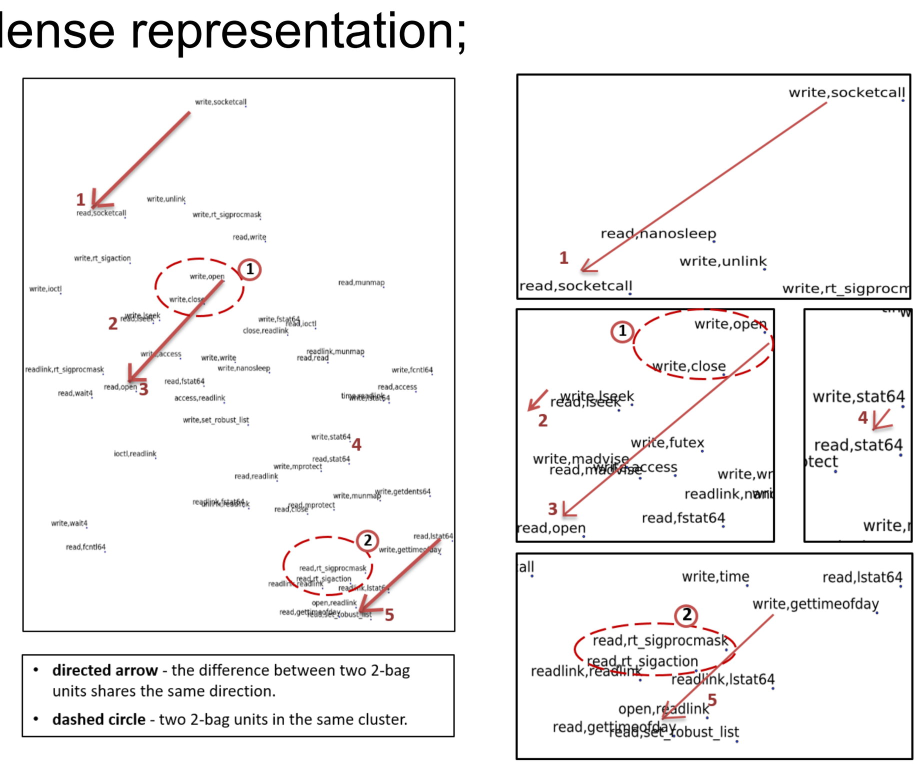


Two-stream Deep Detection Models

- ***Two*** key modules: ***filter-reconstruction module*** and ***deep learning module***.



- ***Filter-reconstruction module:***
  1) system-call filter
  2) n-gram model reconstruction
     - n-gram, n-bag, n-tuple
- ***Deep learning module***:
  1) n-gram word embedding to convert n-gram indices into dense representation;



The distribution of "read" and "write" related 2-tuple of system calls in the Embedding layer. Embedding layer extracts the high-level features in the deep learning model. Difference between "read" and "write" related bags shares the similar direction.

  2) multi-scale spatial models with inception multi-scale CNN to extract local and global information
     - Atrous-convolution layers to broaden the receptive span;
     - two streams of sequence inputs and frequency inputs;
  3) temporal models with LSTM layers to extract temporal features between system calls.

Motivation

- **DeepMalware**
  - **two-stream deep detection** models and multi-stage on-the-fly reaction
  - **Naïve Detector**
    - fast but inaccurate
  - **Deep Detector**
    - accurate but needs to observe long syscall sequences and takes large computation time
- **Preliminary Results for Linux malware**

	Accuracy	Time
Naïve Detector	84.48%	0.010s
Deep Detector	94.36%	0.292s

Multi-stage On-the-fly Reaction

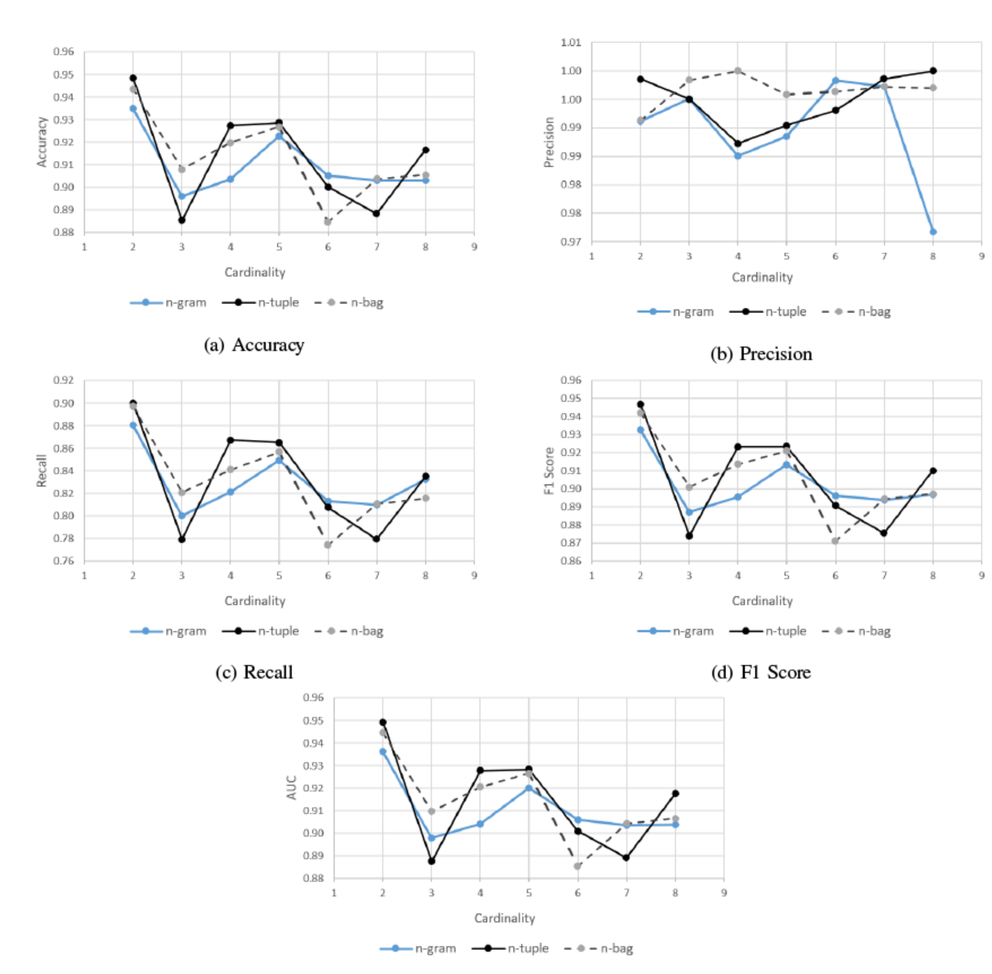
- ***Standard stage*** *(Naïve Detector)*
  - If borderline classification, software will be transferred to uncertain stage.
- ***Uncertain stage*** *(Deep Detector)*
  - Adds obstacles to process execution
  - Whitelisted software: non-intrusive strategies
  - Non-Whitelisted software: intrusive and non-intrusive strategies
- ***Uncertain stage***
  - Buys time for deep learning detection while containing actions of stealthy malware.
  - If software is found benign, transferred to *Standard Stage*.
  - If malicious, transferred to *Decision stage*.
- ***Decision stage***
  - Kill malware, clean-file system

Experiments and Datasets

- **Linux dataset** Ubuntu 14.04
  - 100 malwares from Virustotal
  - 400 benign applications
  - 120,000 samples
- **Windows dataset** Windows 7
  - 30,000 malwares collected from 2013 to 2015
  - 30,000 benign applications
  - 100,000 samples

Preliminary Results

- Deep learning model with the 2-tuple input performed best and achieved a 94.83% accuracy and a 94.66% F1 score in Linux dataset.



Deliverables and Milestones

- Q1: Linux malware detection (Done)
- Q2: Cross-platform (Linux, Windows, Android) malware detection
- Q3: Process based malware detection

References

1. C. Kolbitsch, P. M. Comparetti, C. Kruegel, E. Kirda, X. Zhou, and X. Wang, "Effective and efficient malware detection at the end host," in Proceedings of the 18th Conference on USENIX Security Symposium, ser. SSYM'09, 2009, pp. 351–366.
2. S. A. Hofmeyr, S. Forrest, and A. Somayaji, "Intrusion detection using sequences of system calls," Journal of computer security, vol. 6, no. 3, pp. 151–180, 1998
3. S. Revathi and A. Malathi, "A detailed analysis on nsl-kdd dataset using various machine learning techniques for intrusion detection," International Journal of Engineering Research and Technology. ESRSA Publications, 2013.
4. R. Sun, D. E. Porter, D. Oliveira, and M. Bishop, "The case for less preditable operating system behavior," in Proceedings of the USENIX Workshop on Hot Topics in Operating Systems (HotOS), 2015.
5. M. Xie, J. Hu, and J. Slay, "Evaluating host-based anomaly detection systems: Application of the one-class svm algorithm to adfa-ld," in Fuzzy Systems and Knowledge Discovery (FSKD), 2014 11th International Conference on. IEEE, 2014, pp. 978–982.
6. M. Egele, T. Scholte, E. Kirda, and C. Kruegel, "A survey on automated dynamic malware-analysis techniques and tools," ACM Computing Surveys (CSUR), vol. 44, no. 2, p. 6, 2012.