CERIAS
The Center for Education and Research in Information Assurance and Security

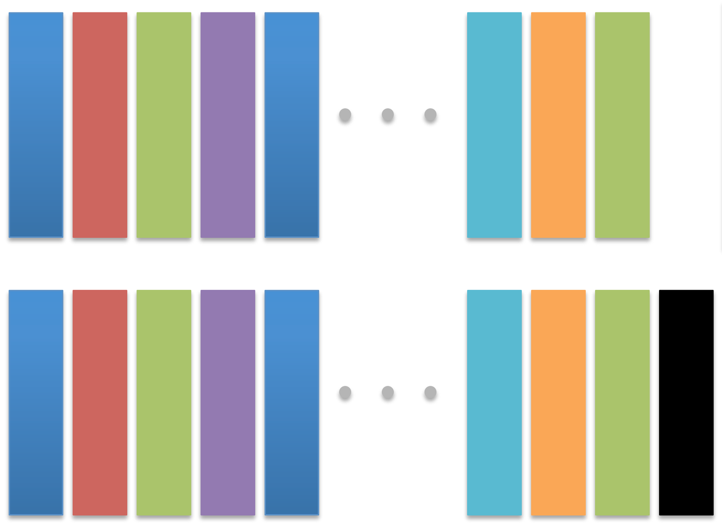
A Deep Learning Based Anomaly Detection Approach for Intelligent Autonomous Systems

Miguel Villarreal-Vasquez and Bharat Bhargava
Department of Computer Science, Purdue University, West Lafayette, IN, USA

MOTIVATION

Given this sequence

Should this sequence occur?



Legend
Each color bar represents one of the all possible events (system calls) in the system.

- ❑ Advanced modern exploits characterize by their sophistication in stealthy attacks.
- ❑ **Code-reuse attacks** such as return-oriented programming and memory disclosure attacks allow attackers executing malicious instruction sequences on victim systems without injecting external code.
- ❑ This research proposes a new Deep Learning based anomaly detection technique that probabilistically models program control flows for behavioral reasoning and live monitoring.
- ❑ We aim to answer **the binary classification problem** of given a sequence of events $e_1e_2e_3\ldots e_k$ **whether or not the sequence should occur**?
- ❑ An event e_i in the sequence is a function call in a a given trace.

ANOMALY DETECTION ALGORITHM

- ❑ We defined an event as a function call. Each possible function call must be identified as they will form the vocabulary of events.
- ❑ The dynamic code behaviors can be learned by training the model with non-malicious program traces.
- ❑ At any time *t* each possible event (system call or library call) in the system is assigned a probability estimated with respect to the sequence of events observed until time *t-1*.
- ❑ At classification, the decision is made with respect to a pre-defined threshold of the *K*-top most probably sequences.

Input: Sequence of events in the system
Output: normal or anomalous

- **Step 1:** Define a finite set E of events $e_1, e_2, \ldots, e_N$ in the system. Events occur in a time-series fashion.
- **Step 2:** At time $t-1$, given an observed series of events $\{e_i{}^1, e_i{}^2, \ldots, e_i{}^{t-1}\}$ (with $i = 1, 2, \ldots or N$) find the set K of the top k events to occur in time t.
- **Step 3:** At time t, the sequence $\{e_i{}^1, e_i{}^2, \ldots, e_i{}^{t-1}, e_i{}^t\}$ is non-anomalous if $e_i{}^t \in K$, otherwise anomalous.

Algorithm 1: Anomaly detection algorithm

THREAT MODEL

The Eternal War

Attack	• Code Injection (e.g., Buffer Overflow)
Defense	• W⊕X (e.g., Windows DEP)
Attack	• Code Reuse (e.g., ROP)
Defense	• ASLR and variants
Attack	• Memory Disclosure (JIT-ROP): Enables Code Reuse
Defense	• Code Re-Randomization / **Deep Learning based Anomaly Detection**

❑ **Invalid and abnormal control flow of a program:**

- ✓ Code Injection
- ✓ Code Reuse
- ✓ Memory Disclosure

❑ **Can be caused by:**

- ✓ Human error (e.g., unauthorized use or operation of the program
- ✓ Software flaws (e.g., buffer overflow vulnerabilities)
- ✓ Attacks by remote attackers
- ✓ Malicious insiders (e.g., through drive-by downloads)

IMPLEMENTATION

MODEL

```
# -*- coding: utf-8 -*-
"""
Deep Learning Based Anomaly Detection Model
"""
# Imports
from __future__ import print_function
import torch
from   torch.autograd import Variable
```

python

PYTORCH
theano
TensorFlow

CUDA

GPU

❑ **Model**

- ✓ Based on Recurrent Neural Networks (RNN): Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU)

❑ **Programming Language: Python**

- ✓ Compatible with several numerical computing libraries suitable for the use of GPUs
- ✓ Some examples: Pytorch, TensorFlow and Theano

❑ **Computing Library: PyTorch**

- ✓ Scientific computing package
- ✓ Replacement of Numpy to take advantage of the power of GPUs
- ✓ Python-friendly platform for neural networks (Deep Learning)

❑ **Parallel Computing Platform: CUDA**

- ✓ Programming model to use GPUs for general purpose computing

CONTRIBUTIONS

- ❑ Systems can be trained with data of benign flows only in isolation (the data is full available.)
- ❑ Flow sensitive anomaly detection. Given the execution paths P_1, P_2 and P_3 our technique captures their occurrence probabilities, vital for high-precision anomaly detection.

REFERENCE

- D. Ulybyshev, M. Villarreal-Vasquez, B. Bhargava, G. Mani, S. Seaberg, P. Conoval, D. Steiner, J. Kobes "Blockhub: Blockchain-based Software Development System for Untrusted Environments", Submitted to IEEE CLOUD 2018.
- M. Villarreal-Vasquez, B. Bhargava, P. Angin, N. Ahmed, D. Goodwin, K. Brin and J. Kobes. An MTD-based Self-Adaptive Resilience Approach for Cloud Systems. In IEEE CLOUD 2017.
- N. Ahmed and B. Bhargava. "Mayflies: A moving target defense framework for distributed systems," in Proceedings of the 2016 ACM Workshop on Moving Target Defense 2016.

Acknowledgement: This research is supported by NGC Research Consortium. We collaborated with Paul Conoval, Donald Steiner and Jason Kobes.

PURDUE UNIVERSITY Discovery Park

CERIAS