Online surveillance of critical computer systems through advanced host-based detection

Harmonized Anomaly Detection Techniques – Project Track 3

Wael Khreich and Wahab Hamou-Lhadj

Software Behaviour Analysis (SBA) Research Lab
Concordia University
Montreal, QC, Canada

May 2, 2013

New Research Directions

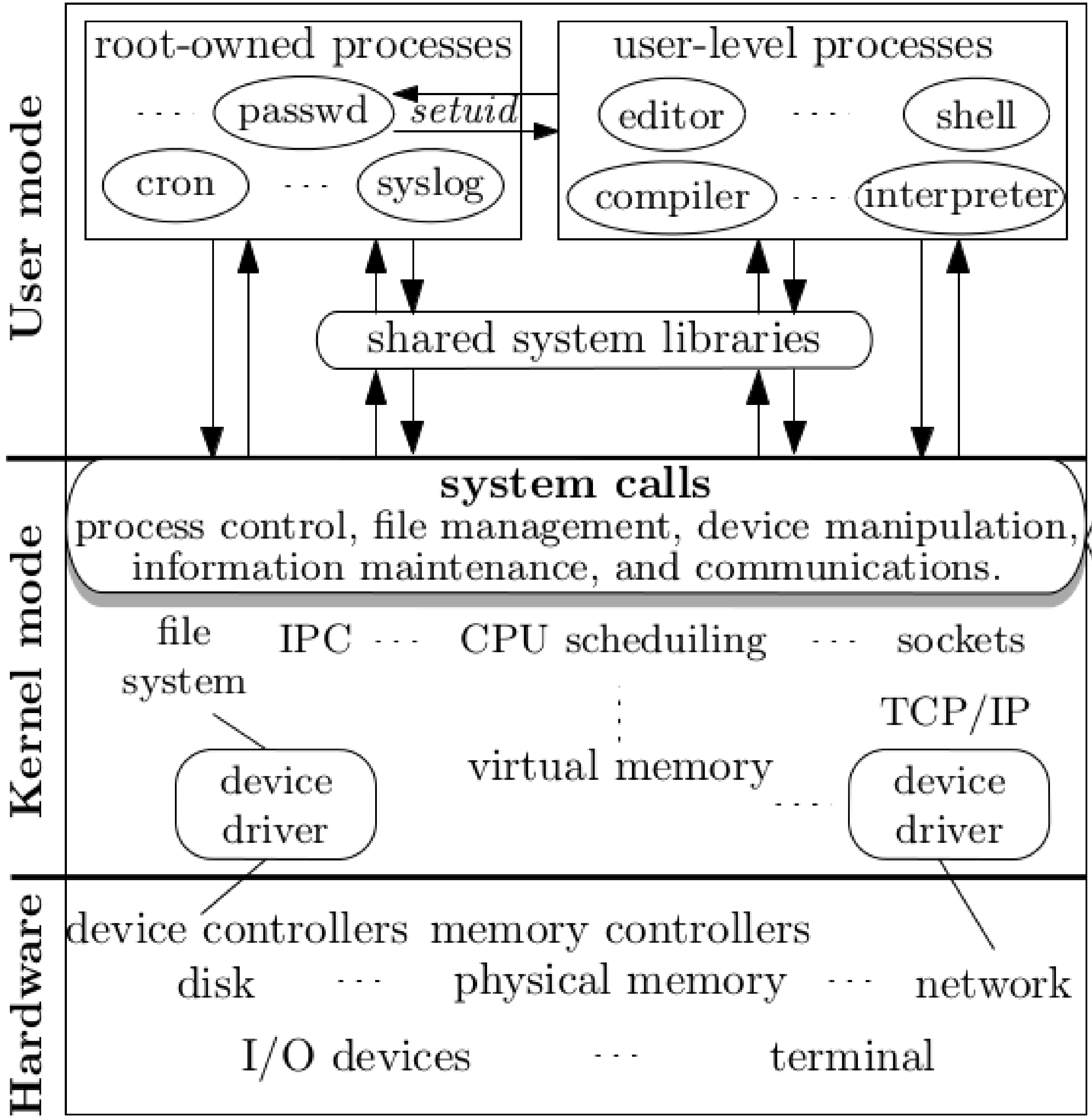
- Focus on anomaly detection at the **system call** level
  - According to feedback received from DRDC during a workshop held in Feb 2013 at Valcartier
  - Based on the exploratory study conducted last year and the available resources

Objectives

- Develop **modular, adaptive,** and **scalable** Anomaly Detection Systems (ADS) based on system calls
- Reduce **false positives (alarms)** and improve the **true positives**
- Develop comprehensive **test beds** and **evaluation protocols**
- Provide preliminary analysis/recommendations for future research on feedback integration and collaborative ADS

System Calls

- Gateway between User/Kernel
- Apps/Libs invoke system calls to request kernel services
- Shown effective in describing normal process behavior



Anomaly Detection based on System Call Sequences

- Constructs profiles of expected normal behavior
  - Using system call traces collected over a period of normal "attack-free" process activities
- Attempts to detect events that deviate significantly from the normal profile
- These deviations are considered as anomalous activities
  - However they are not necessarily malicious
  - Coding or configuration errors

Challenges – False alarms

- ADS is capable of detecting **novel** attacks
  - Unlike signature-based detection techniques, which look for patterns of **known** attacks
- ADSs generate large numbers of false alarms
  - Misclassify normal events as anomalous
- Extensive investigation required to ascertain if an alarm was produced by an attack
- Frequent false alarms reduce the confidence and could lead to deactivation of the ADS

Challenges – False alarms

- False alarms are caused by several reasons including:
  - Unrepresentative normal data for training and attack data for validation and testing
  - Inappropriate model or feature selection
  - Poor optimization of models parameters
  - Overfitting (leads to poor generalization)
  - Inadequate assumptions such as static environments

Assumptions

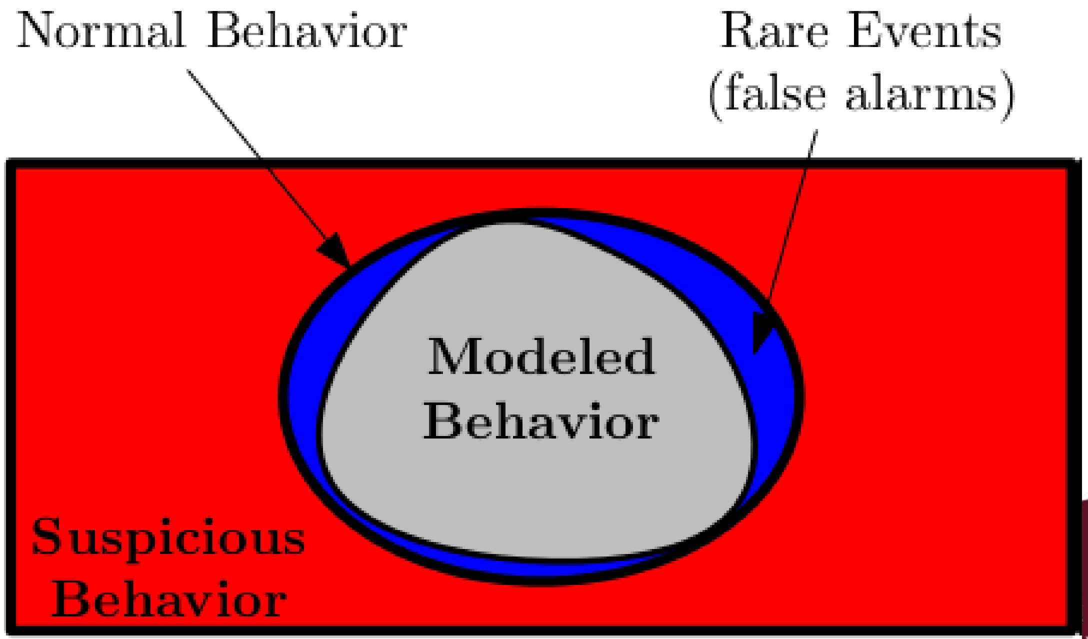
- Most of the work found in related literature assumes:

1. Representative amount of normal data provided for training
2. Static environments: normal behavior will not change over time

In Practice

- ADSs are often designed using limited data
  - collection and analysis of representative data from each process (different version, OS, etc.) is costly

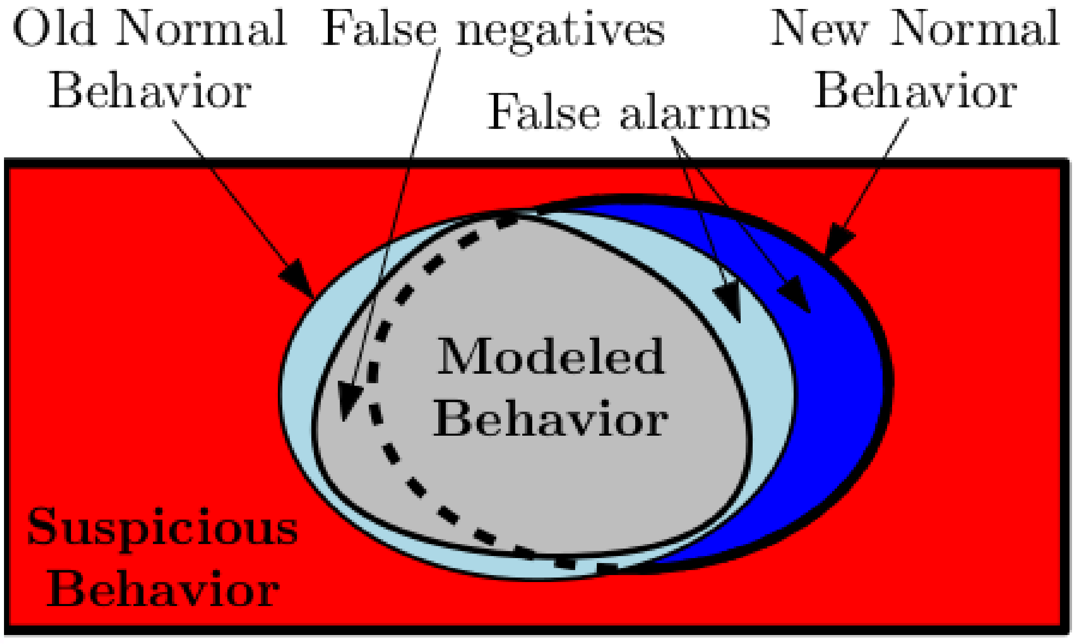
Anomaly detector will have **incomplete** view of normal system behavior



In Practice

- Dynamic environment
  - Changes in normal process behavior due, for instance, to application update

Internal model of normal behavior **diverges** with respect to the underlying data



ADS Requirements

- ADSs should be able to efficiently accommodate new data to:

1. Account for rare normal events (false alarms)
2. Adapt to differences among hosts
   - (e.g., different configurations or OS versions)
3. Adapt to changes in the application
   - (e.g., application update or patches)

- Scalable and modular: can add, replace or remove models or features over time

Proposed Research Solutions

AHLS – Track 3

Performance	Adaptability	Scalability
Attack Taxonomy & Manifestation	Incremental and Online Learning	Data Space Reduction
Dataset Generation	Multiple Classifier Systems	Model Selection
Evaluation Protocol	Integration of Human Feedback	Feature Selection

Performance – Attack Taxonomy and Manifestation

- Today, we still lack clear understanding about how attacks manifest at the system call level
  – Which attack or family of attacks can be detected by using system call sequence?
- Limited information about the level/degree of vulnerability of each system call or sequence of system calls

Performance – Attack Taxonomy and Manifestation

- We have previously analyzed most research papers using system call **arguments,** mainly to detect “mimicry attacks”
  – Mimic normal behavior of system call sequences
- Still unclear which attacks can be detected using system call arguments or return values
- We **started** to create our own taxonomy and analyse attack manifestation
  – at system call sequence and argument levels

Performance – Dataset Generation

- UNM datasets (1998) for benchmarking ADSs based on system calls **sequences**
- DARPA datasets (1999) include system calls and their **arguments**
- Both are not representative for current attacks
- We will **create** comprehensive system call datasets for training, validation, evaluation
  - Improve anomaly detection techniques
- Based on insights from our taxonomy and analysis of attacks and their manifestation

Performance – Evaluation Protocol

- Another issue is lack of unified methodologies and performance metrics for benchmarking
- We will evaluate the proposed solutions under different conditions based on:
  - Accuracy: ROC, PR, Cost curves and other derived measures (e.g., area under these curves)
  - Adaptability: Time required to adapt (models, thresholds, etc.) to changes
  - Efficiency: time and memory complexity during design and operation

Adaptability – Incremental and Online Learning

- Incremental learning techniques try to update model parameters based on new data only
  - Assumes data become available after a model has already been trained and deployed for operations
- Other advantages (besides adaptability) :
  - Reduce data storage (old data could discarded)
  - Reduce time and memory complexity required to update the model parameters.

Adaptability – Incremental and Online Learning

- Investigate various machine learning technique that are suitable for incremental learning
- Possibility of using online learning, data stream mining, and digital signal processing techniques to map and visualize the system call stream over time
- Develop improve incremental/online techniques

Adaptability – Multiple Classifier Systems

- Adaptive systems based on a single detector (one-class classifier) may not be accurate
  - May approximate the underlying data structure or distribution inadequately
- Ensemble methods and multiple classifier systems try overcome this issue by combining the decision from different classifiers
  - Different classifiers may provide different expertise and solutions, commit different errors, etc.
  - Increase in system accuracy

Adaptability – Multiple Classifier Systems

- Most techniques based on ensemble learning or multiple classifiers can be extended to allow the system to adapt to new data
- For instance, by using a learn-and-combine approach
  - train new detectors on the newly-acquired data, and combine their outputs with previously-generated detectors
- In addition to improved accuracy, ensure **modularity** and scalability

Adaptability – Integration of Human Feedback

- Rapid integration of human (or other kind of) feedback in the ADS will help reduce the false alarms over time
- For instance, there should be a mechanism to update the internal models not to generate the same false alarms
- Semi-supervised learning techniques could be suitable for such interactive integration
- We will conduct preliminary analysis

Scalability – Data Space Reduction

- Reduce trace size:
- Extend our previous work on "frequent common pattern"
- Investigate clustering and other data stream mining techniques
- Explore techniques based on information theory
  - suitable metrics to measure the information loss

Scalability – Model Selection

- Model size could vary largely depending on underlying algorithm, e.g.,
  - STIDE sequence matching stores sequences
  - Hidden Markov model stores compact models
- For ADSs based on multiple detectors
- We will find criteria and measures to select most compact and diverse set of models
- Develop mechanisms to manage models (remove, replace, add or update) over time

Scalability – Feature Selection

- Feature selection techniques help reducing both the storage space and the model size
- Could also provide a complimentary view to the anomaly detection problem
- Analysis of attacks and their manifestation will provide invaluable insights on feature selections:
  - Most dangerous system calls
  - Most vulnerable system call sequences
  - Suspicious arguments
  - Etc.

Current and near-future activities

AHLS – Track 3

Performance
- Attack Taxonomy & Manifestation
- Dataset Generation
- Evaluation Protocol

Adaptability
- Incremental and Online Learning
- Multiple Classifier Systems
- Integration of Human Feedback

Scalability
- Data Space Reduction
- Model Selection
- Feature Selection

Thank You

Wael Khreich

Postdoctoral Fellow

wkhreich@encs.concordia.ca

Software Behaviour Analysis (SBA) Research Lab
Concordia University
Montreal, QC, Canada