MAXS: Scaling Malware Execution with Sequential Multi-Hypothesis Testing

Authors: Phani Vadrevu and Roberto Perdisci

Presented by : Ashwag Altayyar

CISC850
Cyber Analytics

Bare-metal Analysis Environments

- Forcing the malware sample to run on a native system.
- Incurring a high hardware costs.
- Therefore, limiting the number of malware samples.

Problem statement

- Malware analysis environments execute each sample blindly
- Most new malware is repackaged previously analyzed malware.

Resource savings vs Information loss

- Increasing the number of malware samples
- Increasing execution time
- Saving information

- Reducing the number of malware samples
- Reducing execution time
- Losing information

- Increasing the number of malware samples.
- Reducing the amount of execution time.
- Minimizing the risk of information loss.

UNIVERSITY OF DELAWARE

MAXS(Malware Analysis eXecution Scaler)

A novel probabilistic multi-hypothesis testing framework for scaling execution in malware analysis environments, including bare-metal execution environments.

Goals and Benefits:

- Increasing the capacity of malware analysis environments by reducing the execution time for each sample.

- Minimizing the information loss.

- MAXS provides a new probabilistic decision framework .

- Every time a new event is observed :

1- The probability that the sample belongs to a previously learned malware family.

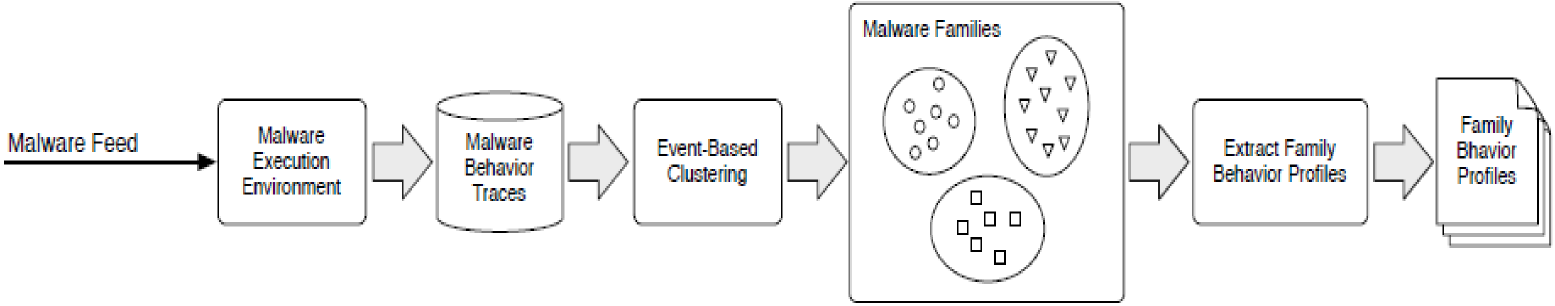
2- The probability that the sample will generate previously unseen malware behaviors.

MAXS FRAMEWORK

1- A learning phase

2- An operational phase

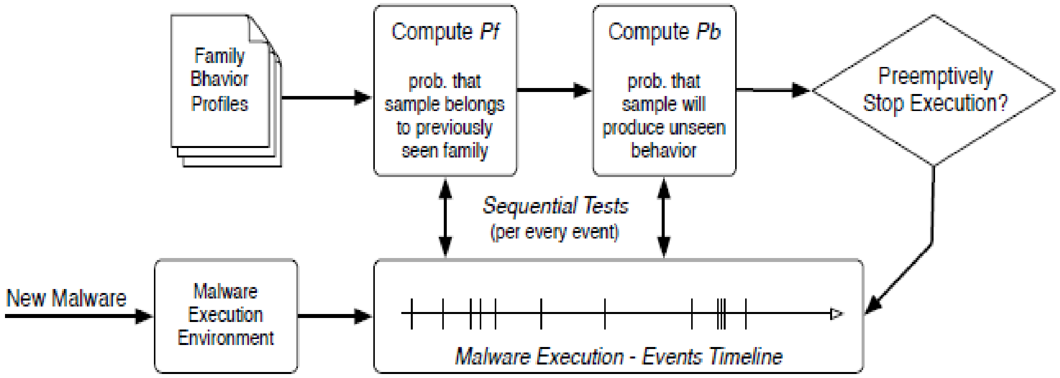
Learning Phase



(a) Learning phase: malware family discovery and family behavior profile extraction.

- Measuring the similarity by computing the Jaccard index.
- Using DBSCAN clustering algorithm (Density-based spatial clustering of applications with noise) .

Operational Phase



(b) Operational phase: overview of sequential tests applied to new malware samples.

Figure 1: MAXS framework.

main parameters to examine the Probabilities

β	Threshold to examine the probability (Pf)
γ	Threshold to examine the probability (Pb)

EVALUATION

Goal :

- Decreasing the execution time while minimizing the information loss
- Dataset:
- Two large collections of malware execution traces obtained from two different production-level analysis environments (SA , SB)
- 1,251,865 malware samples from SA, and 400,041 from SB

dataset	prefixed run time	collection days	avg. samples / day	avg. samples with DNS queries / day
M_A	240s	77	16,258	15,431
M_B	360s	6	66,674	62,063

Table 1: Summary of malware dataset properties.

Experiments Setup

- Appling to different types of events:
  - Domain name queries extracted via dynamic analysis
  - Malware information extracted via static analysis
- Measuring time savings and information loss

Experiment 1: Malware Domain Intelligence

- MAXS monitors the sequence of domain name queries
- performed on both datasets MA and MB.

Parameter Selection

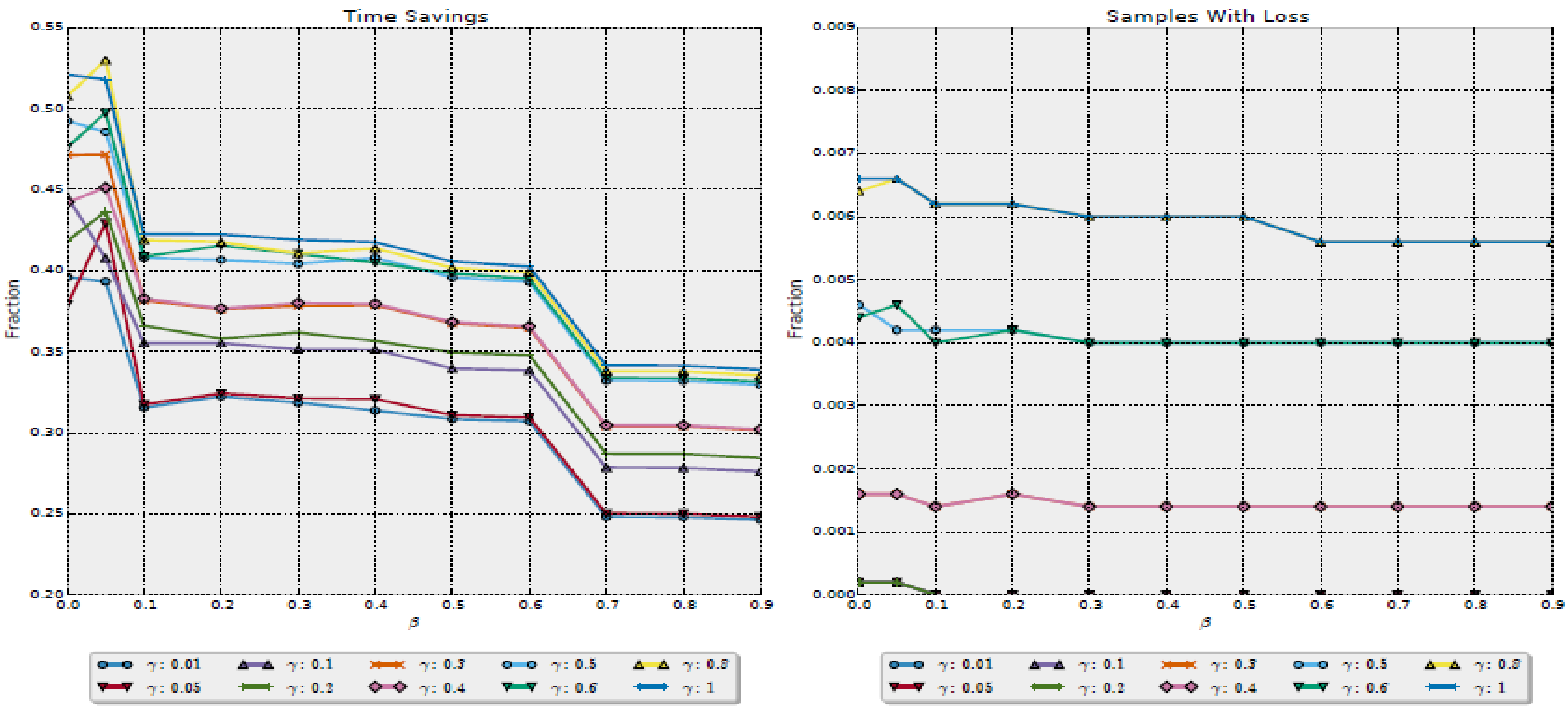


Figure 2: Parameter Selection Experiment

β = 0.05 and γ = 0.1, time savings above 40% with less than 0.1% of sample with information loss

Longitudinal Train-Test Experiments

Dataset MA:

- Over three months (July, August, and December 2013)
- Three contiguous days for training and building the family behavior profiles.
- The next day for testing and measuring the time savings and information loss .

Dataset MB:

- Over six days (November 2014)
- One day of malware samples for training and one day for testing.

Longitudinal Train-Test Experiments

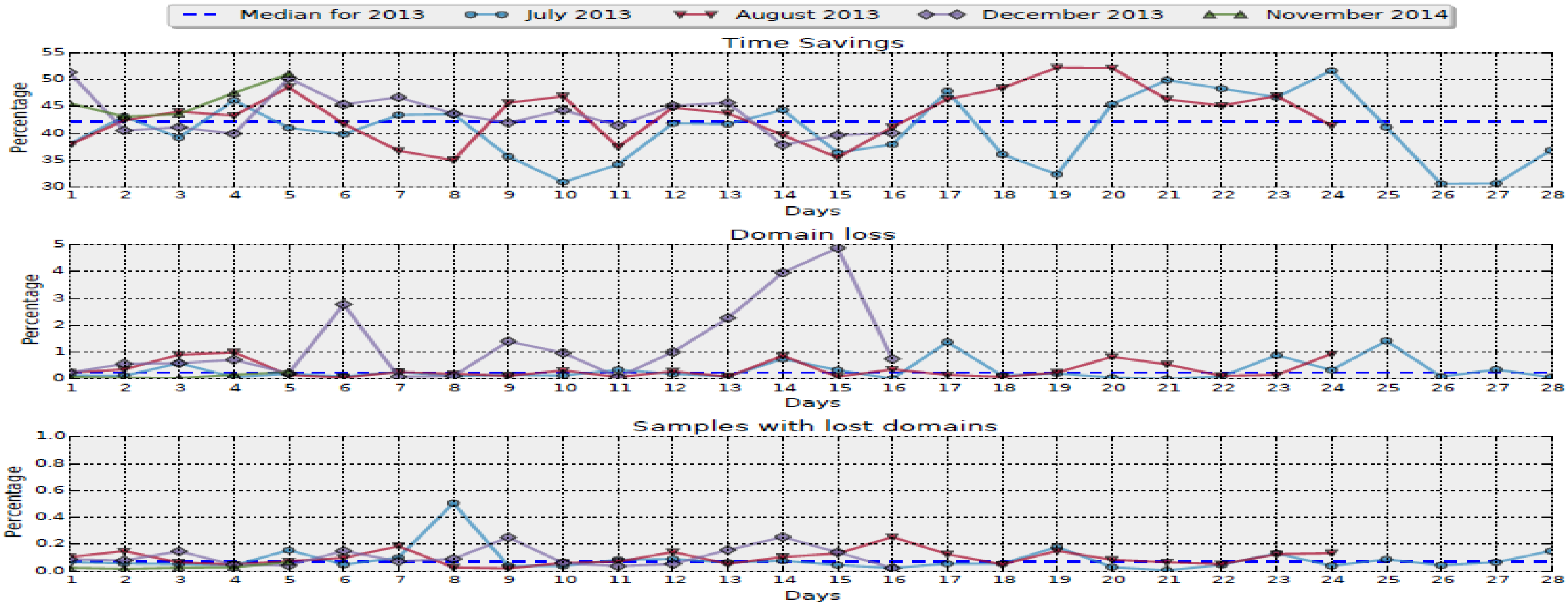


Figure 3: Longitudinal Study Experiment

dataset	median time savings	median domain-based information loss	median samples responsible for loss
MA	42.2%	0.25%	0.07%
MB	45.5%	0.08%	0.03%

Summary of Result for Longitudinal Experiments

Dataset	samples	samples assigned to a family	avg. family assignment time	avg. stop time	median time savings
M_A	15,431	9,201	24.4s	69.6s	42.2%
M_B	62,063	34,305	28.3s	50.4s	45.5%

Table 2: Summary of results for longitudinal study experiments.

dataset	prefixed run time	collection days	avg. samples / day	avg. samples with DNS queries / day
M_A	240s	77	16,258	15,431
M_B	360s	6	66,674	62,063

Table 1: Summary of malware dataset properties.

Experiment 2: Leveraging Static Analysis Information

- Clustering the malware samples based on static analysis features and building family behavior profiles.

- Testing a new sample to decide whether it should be executed or not

The Result of Applying MAXS on Static Analysis Information

Information	Time Saving %	Domain Loss %	# Domains Lost
Static+Network	50.93 %	0.3 %	114
Static	37.16 %	0.22 %	82
+Network	22.01 %	0.08 %	32
Network	45.5 %	0.08 %	35

Table 3: Average results for a "cascade" decision process using both static analysis information and network events.

Combining Static and Dynamic Analysis

- Appling MAXS on static analysis information

- For every malware sample executed in the first step, apply MAXS over the network events

Information	Time Saving %	Domain Loss %	# Domains Lost
Static+Network	50.93 %	0.3 %	114
Static	37.16 %	0.22 %	82
+Network	22.01 %	0.08 %	32
Network	45.5 %	0.08 %	35

Table 3: Average results for a "cascade" decision process using both static analysis information and network events.

Conclusion

The experimental results show that:

- Reduce malware execution time in average by up to 50%, with less than 0.3% information loss.
- Lower the cost of bare-metal analysis environments.