הנדסת מערכות תוכנה ומידע

Software and Information Systems Engineering

Query-Efficient Black-Box Attack Against Sequence-Based Malware Classifiers

Authors:
Ishai Rosenberg
Asaf Shabtai
Yuval Elovici
Lior Rokach

Annual Computer Security Applications Conference (ACSAC) 2020

CBG / CSRC
Cyber@Ben-Gurion University of the Negev
Cyber Security Research Center
Israel National Cyber Bureau

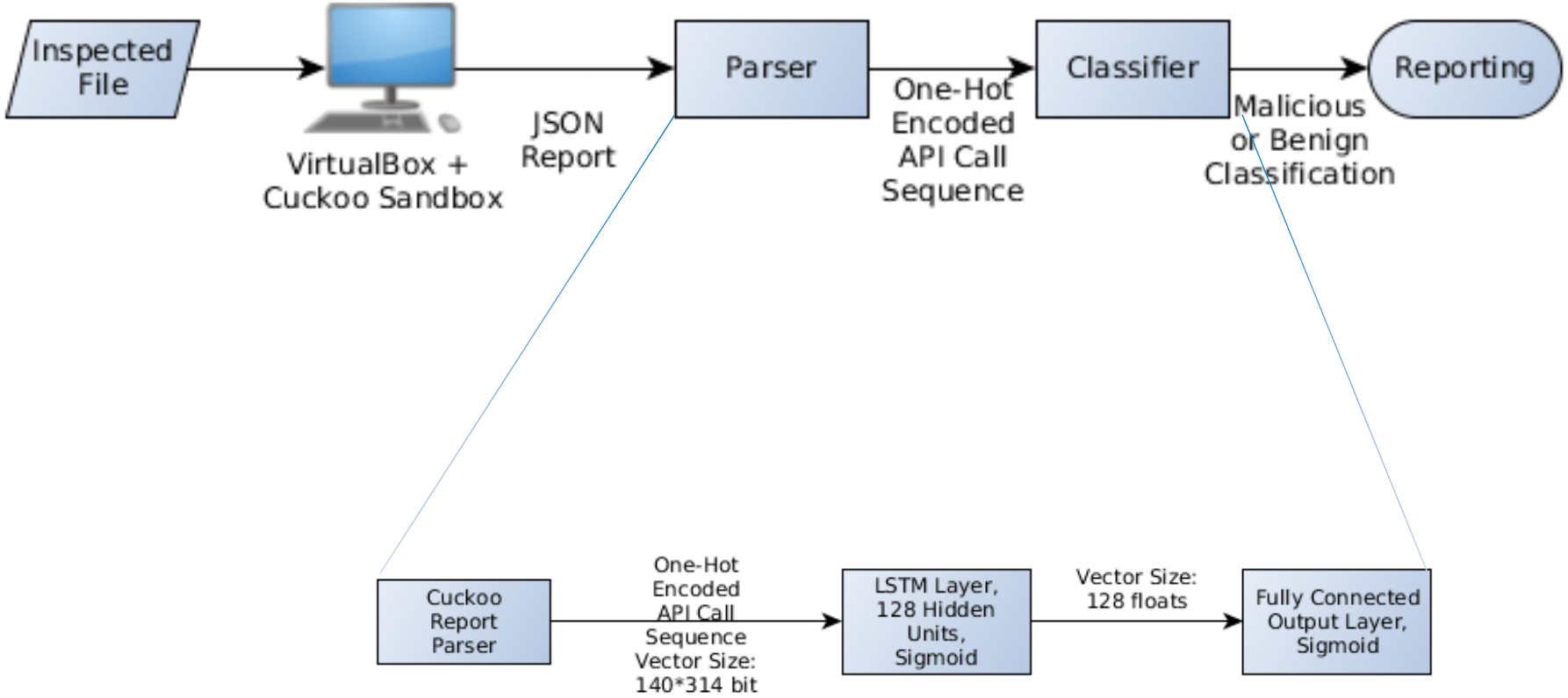
Outline

- Dynamic analysis classifier, using API call sequences as features
- The attack modifies malicious code to be classified as benign
  - Without modifying its malicious functionality.
- Two variants
  - Decision-based: Adding random perturbations or benign perturbations to the API call sequence, using logarithmic backtracking
  - Score-based: Using adaptive evolutionary algorithm as random search
- Generic attack - works against many classifiers' types
- Minimizes the number of target classifier queries (attacking cloud model scenario)

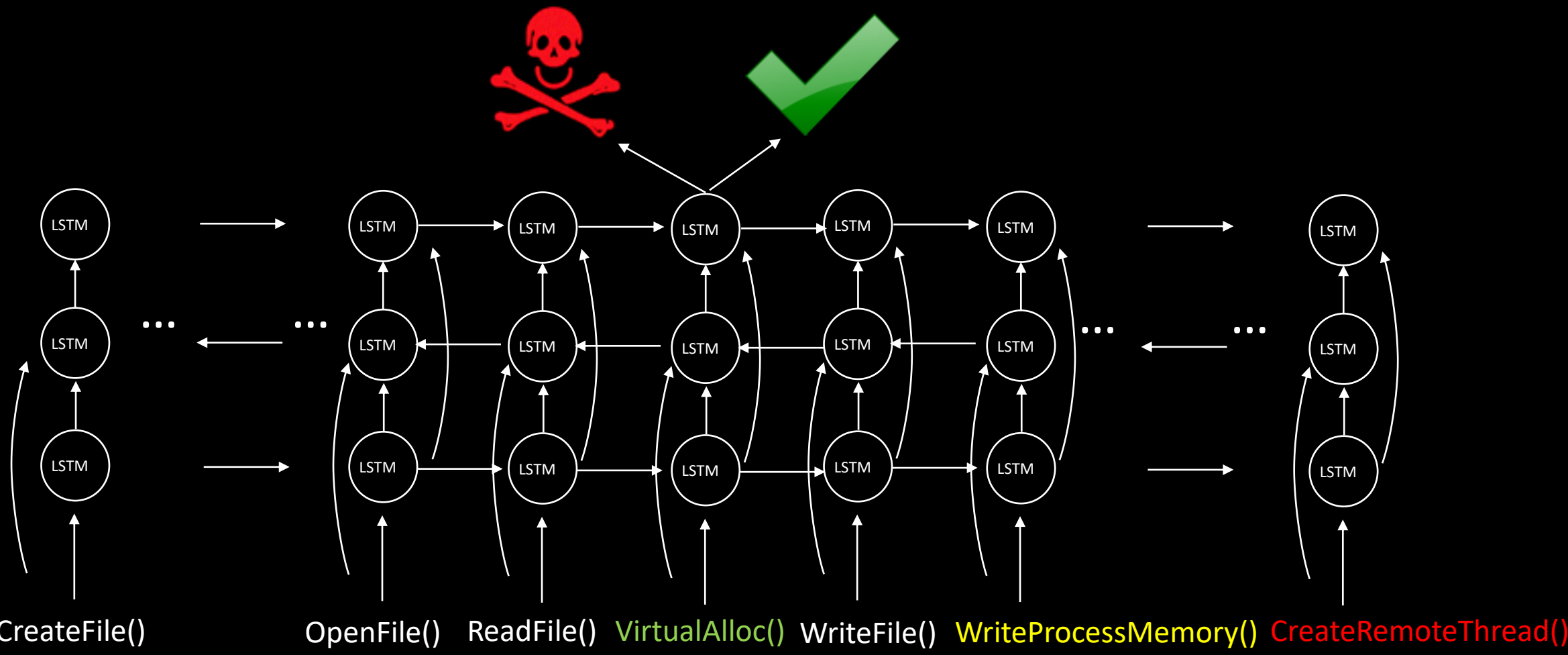
Proposed Attack Challenges

1. (Our goal:) Minimize the number of black box classifier queries
   - In a cloud service (MLAAS), every query costs money
   - More queries increase the chance of attack detection using stateful defenses that keep track of queries to the system (e.g., Chen et al. 2019)
2. Unique challenges in the cyber security domain:
   - The Original (Malicious) Functionality Must Remain Intact
     - Changing a pixel's color doesn't "break" the image
   - What would be considered a 'small perturbation' of WriteFile()?
     - Modifying it to ReadFile() (a different operation for the same media)?
     - Modifying it to RegSetValueEx() (a similar operation for a different media)?

Overview of the Malware Classification Process



(LSTM) Dynamic Classier Architecture



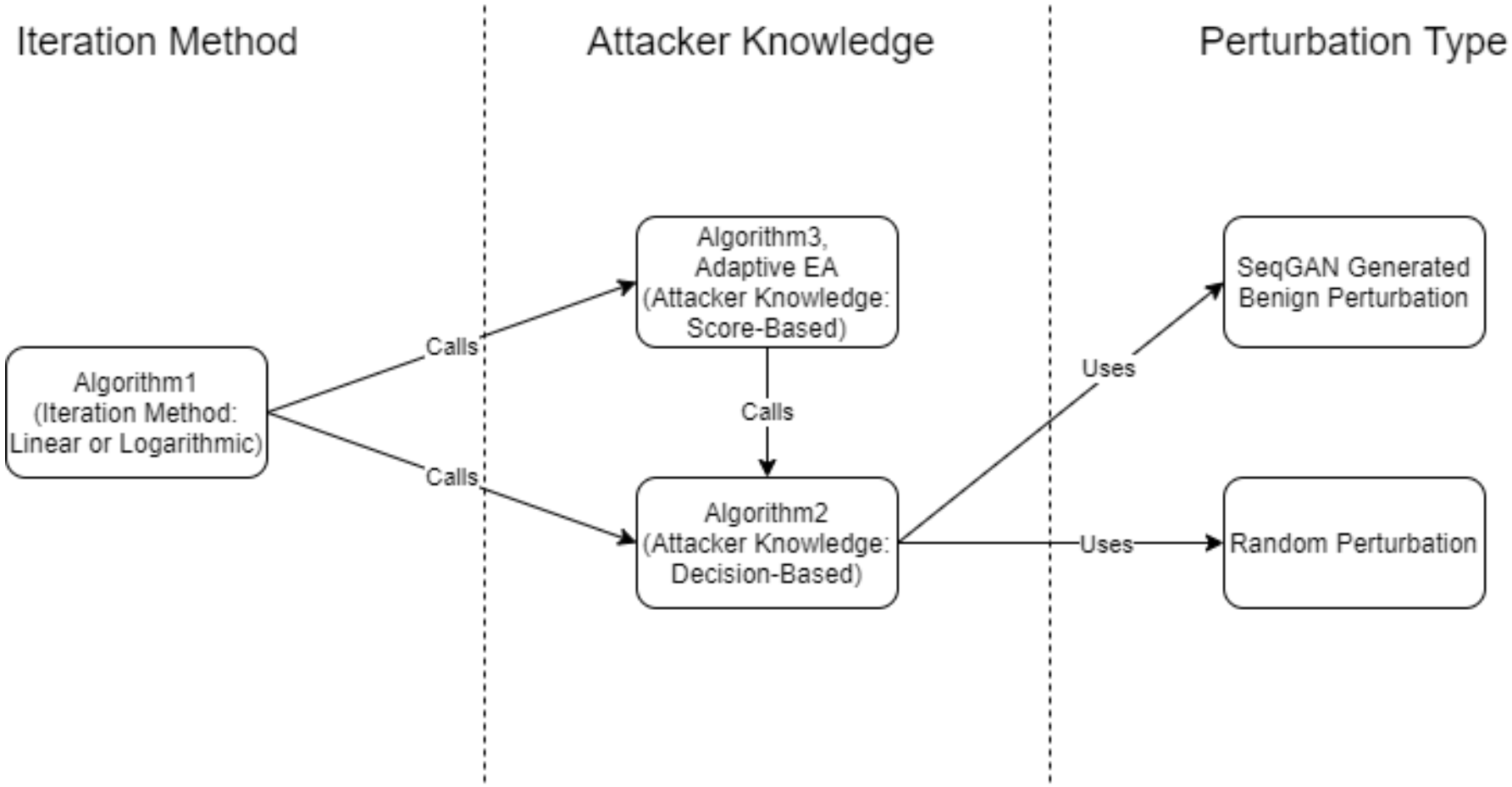

```
s.send("GET /" + sys.argv[2] + " HTTP/1.1\r\n
s.send("Host: " + sys.argv[1]  + "\r\n\r\n");
s.close()
for i in range(1, 1000):
attack()

import socket, sys, os
print "][Remote DDOS Address" + sys.argv[1]
print "injecting " + sys.argv[2];
def attack():
$pid = os.fork()
s = socket.socket(socket.AF_INET, socket.SOCK
s.connect((sys.argv[1], 80))
print ">> GET /" + sys.argv[2] + " HTTP/1.1"
s.send("GET /" + sys.argv[2] + " HTTP/1.1"
s.send("Host: " + sys.argv[2] + " HTTP/1.1\r\n
s.close()
```

Credit: daliu/Fotolia, https://www.sciencedaily.com/releases/2017/11/171101130507

The Threat Model

- The malware classifier is based only on API call sequences
- The adversary has access to:
  - The malware classifier as a black-box. Two variants:
    - Decision-based: Input -> Classification
    - Score-based: Input -> Confidence Score
  - The malware classifier's features (the monitored API calls)
    - Or at least a subset of them
  - No knowledge about the classifier's type, architecture or any other hyper-parameters

The Proposed Attack's Flow



Techniques used in The Proposed Attack

- Logarithmic Back-Tracking – Starting with a large ratio of perturbation and rapidly decrease it
- Benign Perturbation – SeqGAN (Yu et al., 2017) generated samples from benign class, in contrast to random perturbation (more effective)
  - Using hard-coded benign API subsequences can be easily black-listed
- Self-Adaptive Evolutionary Algorithm – Score-based optimization technique

Self-Adaptive Uniform Mixing Evolutionary Algorithm

- Similar to Dang and Lehre, 2016.
- In every generation, a new population of candidates is produced
  - The old generation dies.
- Besides the adversarial sequence, each candidate carries its mutation rate.
  - Mutation is adding an API call, not removing or modifying one.
    - With invalid arguments (e.g., writing a non existing file, etc.).
    - To prevent harming the malicious functionality of the API call sequence
- Each new candidate is produced in the same way:
  1. The best of two uniformly selected individuals is selected (**no crossover**).
  2. The selected parent individual changes its mutation rate between two mutation rates: low and high, with a probability p
  3. The parent replicates, with mutations occurring at the new mutation rate.

Experimental Evaluation (Decision-Based Attack)

Table 2: Decision-Based Attack Performance (100 Queries)

Classifier Type	Attack Effectiveness [%] Our Decision-Based Attack	Attack Effectiveness [%] BiRand, Dang et al. 2017 [24]	Attack Effectiveness [%] Rosenberg et al. 2018 [42]	Additional API Calls [%] Our Decision-Based Attack	Additional API Calls [%] BiRand, Dang et al. 2017 [24]	Additional API Calls [%] Rosenberg et al. 2018 [42]
LSTM	**62.21**	39.49	51.15	22.22	31.42	17.22
Deep LSTM	**63.62**	40.38	50.80	22.71	32.12	29.51
GRU	**63.35**	40.21	51.16	21.47	30.36	16.09
1D CNN	**63.63**	40.39	48.93	4.10	5.80	49.21
Logistic Regression	**41.47**	26.32	35.67	4.43	6.26	7.58
Random Forest	**63.24**	40.14	50.87	5.20	7.35	9.40
SVM	**42.59**	27.04	36.27	3.82	5.40	7.19
Gradient Boosted Tree	**41.62**	26.41	36.55	13.99	19.78	27.80

Experimental Evaluation (Score-Based Attack)

Table 3: Random *Perturbation Type* Attack Effectiveness Comparison for a Fixed Number of Queries (LSTM Model)

Number of Queries Logarithmic Backtracking (/BiRand)*Iteration Method*	100 Linear	200 Linear	2500 Linear	100 Log	200 Log	2500 Log
Our Score-Based Attack (Score-Based *Attacker Knowledge*)	**58.75**	**67.59**	**100.00**	**69.28**	**79.71**	**100.00**
Our Decision-Based Attack (Decision-Based *Attacker Knowledge*)	19.86	21.25	31.43	39.49	42.25	62.50
Rosenberg et al. [42]	51.15	67.11	99.99	51.15	67.11	99.99
Uesato et al. [43]	2.37	2.73	2.87	5.17	5.95	6.25
Ilyas et al. [30]	37.50	43.14	74.94	43.78	50.37	87.50
Alzantot et al. [17], Xu et al. [45]	54.68	62.91	100.00	62.06	71.40	100.00

Table 4: Benign *Perturbation Type* Attack Effectiveness Comparison for a Fixed Number of Queries (LSTM Model)

Number of Queries Logarithmic Backtracking (/BiRand)*Iteration Method*	100 Linear	200 Linear	2500 Linear	100 Log	200 Log	2500 Log
Our Score-Based Attack (Score-Based *Attacker Knowledge*)	**71.90**	**82.70**	**100.00**	**84.77**	**97.53**	**100.00**
Our Decision-Based Attack (Decision-Based *Attacker Knowledge*)	41.34	44.24	46.34	62.21	63.97	87.96
Rosenberg et al. [42]	51.15	67.11	99.99	51.15	67.11	99.99
Uesato et al. [43]	6.56	7.55	11.47	14.31	16.46	25.00
Ilyas et al. [30]	66.23	76.19	81.25	77.32	88.96	94.87
Alzantot et al. [17], Xu et al. [45]	68.49	81.09	100.00	79.93	91.96	100.00

Main Contributions

1. Novel query-efficient end-to-end black-box attack, preserving the malware functionality after perturbation
2. Classifier-agnostic attack: Effective against conventional RNN, LSTM, GRU, bidirectional and deep variants, fully-connected DNN, 1D- CNN, SVM, logistic regression, random forest, GBDT, etc.
3. First generalized decision-based attack for RNN variants, feed-forward networks and traditional machine learning classifiers
4. Online attack - No pre-deployment phase (e.g., surrogate model training).

Questions?