Exploring Adversarial Examples in Malware Detection

요약

- CNN, raw bytes 기반 악성 코드 분류
  - MalConv
  - 특별한 feature engineering 없이도 높은 성능
  - 하지만 AE attack에 대한 강건성(robustness)는 조사되지 않음
- 기존 evasive 목적의 AE 공격
  - 이미지 분야에서는 활발한 공격/방어 방법 연구
  - 하지만 binary executable은 임의로 고치는 것이 안되는 만큼 쉽게 적용되지는 않음
- 본 연구에서는
  - 악성코드 분야에서의 AE에 대해 알아본다.
  - 상용 scale의 대규모 데이터셋에서 검증
  - 어떤 공격법은 덜 효과적이고
  - 아키텍쳐적인 약점을 이용하는 효과적인 공격도 제안해 본다.

Introduction

- 기존 AE 에 대한 연구
  - evasion attack on test-time instance
  - SOTA focus mainly on image classifier
  - small perturbation to input pixels lead to large shift in feature space
- 반면 악성 코드에서는 도메인 특화된 특성이 AE의 응용을 제약
  - 임의로 binary를 고치면 악성 기능 저하/해제, 실행 불능 등
  - 공개된 데이터셋이 상대적으로 적다.
- 기존 연구(kolosnajaji, 2018)
  - 매우 작은 데이터셋이 약점
  - 단순히 바이너리 끝에 변화를 주는 제한된 변형 방식
- 우리의 데이터셋
  - Full
    - 16.3M PE : 12.5 train, 3.8 test
  - Mini
    - 1M
- Baseline
  - Full dataset, Malconv => acc=0.89, AUC=0.97
  - Mini dataset, small model => acc=0.73, AUC=0.82

Append Attack

- Random Append : append arbitrary bytes at end
- Gradient Append
  - append arbitrary bytes first
  - modify the bytes until simulated evasion success
  - guided by derivates of classification loss w.r.t input values

    - performed by each byte seperativly
    - prohibitively expensive
- Benign Append
    - append leading bytes extracted from begin instances with high confidence
- FGM Append
    - convergence time of gradient attack is very huge, linear by numBytes
    - update embeeding value instead of bytes itself
    - just move by learning rate to minimize loss

Slack Attack

- Malconv는 2MB까지만 다루므로, 2MB 이상 파일에 append attack은 소용없음
- 통계적으로보니, 초반 bytes들이 분류에 더 영향을 많이 끼친다.
    - 그러므로 append attack의 효용성이 떨어진다.
- 이는 append보다는 기존 binary에 대한 modify 를 하는 것이 필요함을 나타낸다.
- Slack FGM
    - PE를 깨트리지 않으면서도 어떻게 쉽게 AE를 얻을 수 있을까?
    - 기존 PE에서 가장 큰 Slack 공간을 찾는다.
    - PE의 각 섹션간 사이의 빈 공간이 있다고 한다. 이는 컴파일러나 다른 이유로 인해..
    - 이 중 가장 큰 Slack공간을 찾아서 기존 bytes를 FGM으로 변형

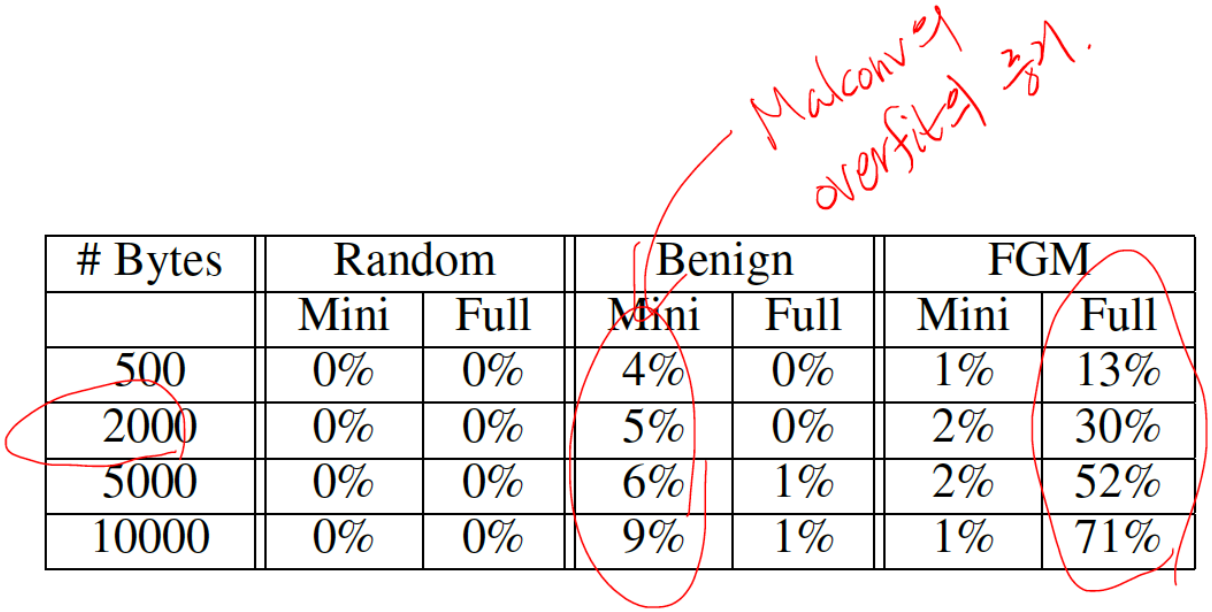
실험 결과

Malconv의 overfit의 증거.

# Bytes	Random		Benign		FGM	
	Mini	Full	Mini	Full	Mini	Full
500	0%	0%	4%	0%	1%	13%
2000	0%	0%	5%	0%	2%	30%
5000	0%	0%	6%	1%	2%	52%
10000	0%	0%	9%	1%	1%	71%

Table 1: Success Rate of the Append attacks for increasing number of bytes on both the Mini and Full datasets.

- Random Append 공격은 하나도 성공하지 못했다.
- Binign Append 공격은 Mini에서는 조금 성공했지만, Full에서는 실패했다.
- FGM Append 공격은 71% 정도로 매우 성공적이었다.
    - 반면 Mini set에서는 성공적이지 않았다? 실험때는 항상 큰 데이터셋을 쓰자
    - Mini model 자체가 not generalized well 모델이라서 이에 기반한 FGM도 효과적이지 않을 것이 당연할수도



# Bytes	Random		Benign		FGM	
	Mini	Full	Mini	Full	Mini	Full
500	0%	0%	4%	0%	1%	13%
2000	0%	0%	5%	0%	2%	30%
5000	0%	0%	6%	1%	2%	52%
10000	0%	0%	9%	1%	1%	71%

Table 1: Success Rate of the Append attacks for increasing number of bytes on both the Mini and Full datasets.

- FGM append가 효과적이기는 하지만 Slack FGM과 비교해 본다면 동일한 Success Rate를 얻기 위해서는 더 많은 bytes를 고쳐야 한다.
- Slack FGM은 약 1000bytes만 고쳐도 약 26%의 SR 달성

In [ ]: