쿼리를 사용하지 않는 딥러닝 모델 탈취 공격 연구

조윤기*, 이영한*, 전소희*, 백윤흥*
*서울대학교 전기정보공학과, 반도체 공동연구소
ygcho@sor.snu.ac.kr, yhlee@sor.snu.ac.kr, shjun@sor.snu.ac.kr, ypaek@snu.ac.kr

A Study on Non-query Based Model Extraction Attacks

Yungi Cho*, Younghan Lee*, Sohee Jun*, Yunheung Paek*
* *Department of Electrical and Computer Engineering and Inter-university Semiconductor Research Center, Seoul National University

요 약

인공지능 기술은 모든 분야에서 혁신을 이뤄내고 있다. 이와 동시에 인공지능 모델에 대한 여러 보안적인 문제점이 야기되고 있다. 그 중 대표적인 문제는 많은 인적/물적 자원을 통해 개발한 모델을 악의적인 사용자가 탈취하는 것이다. 모델 탈취가 발생할 경우, 경제적인 문제뿐만 아니라 모델 자체의 취약성을 드러낼 수 있다. 현재 많은 연구가 쿼리를 통해 얻는 모델의 입력과 출력을 분석하여 모델의 의사경계면 또는 모델의 기능성을 탈취하고 있다. 하지만 쿼리 기반의 탈취 공격은 획득할 수 있는 정보가 제한적이기 때문에 완벽한 탈취가 어렵다. 이에 따라 딥러닝 모델 연산 과정에서 데이터 스니핑 또는 캐시 부채널 공격을 통해 추가적인 정보 또는 완전한 모델을 탈취하려는 연구가 진행되고 있다. 본 논문에서는 최근 연구 동향과 쿼리 기반 공격과의 차이점을 분석하고 연구한다.

1. 서론

최근 인공지능 모델들은 광범위한 분야에서 활용되고 있다. 예를 들어 다양한 기업에서 API 를 통해 모델의 의사결정 서비스를 제공하고 학계에서도 각광받는 주제이다. 이와 같이 인공지능 모델, 특히 딥러닝 분야의 부흥과 함께 모델에 대한 여러 보안적 이슈가 대두되고 있다.

현재에 와서는 AE(Adversarial Example) 생성[1], MIA(Membership Inference Attack)[2], Model Poisoning[3]과 같은 문제들이 이슈가 되고 있다. 그 중 Model Extraction[4,5,6,7] 공격의 경우, 굉장히 적은 자원/비용을 사용하여 모델 기술을 훔칠 수 있을 뿐만 아니라, 다른 공격들의 Black-Box 인 threat model 을 White-box 상황으로 완화시킬 수 있다. 소프트웨어의 경우에는 Binary Obfuscation[8], Anti-reverse Engineering[9], Code Encryption[10]등 소프트웨어의 불법적인 복제와 공격을 위한 분석 방어에 대한 연구가 많이 진행되어 왔지만, 딥러닝 모델에 대해서는 아직 부족하다. 왜냐하면 기존 소프트웨어와 달리 딥러닝 모델의 핵심은 입력과 출력을 매핑시키는 것이고 이러한 매핑 관계는 기존 소프트웨어 작동 과정이 아닌 모델의 의사결정 결과에서 드러나기 때문이다.

최근 모델 탈취 공격[11, 12]는 최소한의 쿼리를 통한 모델의 의사결정 결과에서 최대한의 정보를 뽑는 것을 목표로 한다. 실제로 앞의 연구에서는 기존 모델과 유사한 정확도를 보이고 있으며 이정도의 결과를 가지고도 다른 인공지능 모델 공격의 initial step 으로 충분히 활용될 수 있음을 보여주고 있다. 하지만 모델의 실제 가중치 값들은 알아낼 수 없고, 모델의 구조에 대한 정보도 알 수 없다. 단지 정확도를 기반으로 한 탈취 여부와 결과를 분석하기 때문에 타겟 모델과 위조 모델이 정확하게 동일하다고 말하기는 어렵다.

이에 따라 많은 연구[4,5,13]에서는 쿼리가 아닌 딥러닝 연산과정 또는 이에 발생하는 부채널 공격 벡터들을 활용한 모델 탈취 공격들이 활발히 연구 진행중이다.

2. 모델 탈취 공격

초기의 모델 탈취 공격은 모델의 의사결정경계면 즉, 입력에 따라 매핑된 출력이 바뀌는 지점에 집중하였다. 왜냐하면 의사결정경계면을 잘 위조할수록 타겟 모델과 유사한 성능을 보일 수 있기 때문이다. Orekondy et al[6]의 경우, 모델의 Confidence Score 가 낮은 입력이 의사경계면과 가깝다는 가정을 사용하여 강화학습을 통해 타겟 모델에게서 Confidence Score 가

Algorithm 1: gemm_nn in OpenBLAS.

Input : Matrix A, B, C; Scalar α, β; Block size P, Q, R; *UNROLL*
Output : $C := \alpha A \cdot B + \beta C$

```
1  for j = 0, n, R do // Loop 1
2      for l = 0, k, Q do // Loop 2
           // Loop 3, 1st iteration
3          itcopy(A[0, l], buf_A, P, Q)
4          for jj = j, j + R, 3UNROLL do // Loop 4
5              oncopy(B[l, jj], buf_B + (jj − j) × Q, Q, 3UNROLL)
6              kernel(buf_A, buf_B + (jj − j) × Q, C[l, j], P, Q, 3UNROLL)
7          end
           // Loop 3, rest iterations
8          for i = P, m, P do
9              itcopy(A[i, l], buf_A, P, Q)
10             kernel(buf_A, buf_B, C[l, j], P, Q, R)
11         end
12     end
13 end
```

(알고리즘 1) OpenBLAS 에서의 GEMM[4]

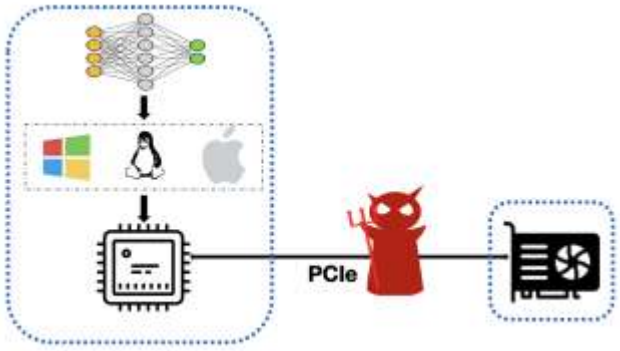
낮을 것으로 추측되는 입력을 샘플링하여 쿼리를 보냈고, 해당 쿼리 결과를 위조 모델에 재학습 시키는 방식을 통하여 모델을 탈취하였다. 이와 반대로 Goodfellow et al[7]은 AE 생성 알고리즘과 유사한 알고리즘을 활용하여 의사경계면에 가까운 입력을 생성해내는 방식을 사용하였다. AE 생성 알고리즘은 이미지에 변조를 주어 해당 이미지에 대한 모델의 출력을 변화시키기 때문에 의사경계면에 대한 정보를 나타낼 수 있기 때문이다.

하지만 위와 같은 연구들은 의사결정경계면을 찾기 위해 많은 쿼리수가 요구되고, 이에 비해 정확한 모델 탈취가 불가능하기 때문에 Active learning 의 라벨링 코스트 최적화 알고리즘을 활용하여 더 적은 쿼리로 정확한 모델 탈취를 시도하는 연구들이 진행되었다[11,12].

그럼에도 불구하고 여전히 실제 타겟 모델의 의사결정을 정확하게 위조할 수 없고, 모델의 구조 또한 동일하다는 강한 가정을 가지고 있다. 그렇기 때문에 더 많은 정보를 탈취할 수 있는 공격 방법들이 요구되었고, 캐쉬 사이드 채널 공격, 데이터 스니핑과 같은 공격들을 활용되기 시작하였다.

3. 캐시 부채널 공격을 활용한 모델 탈취 공격

캐시 부채널 공격은 데이터 또는 코드가 캐시에 존재하는 지 여부에 따라 액세스 속도에 차이가 있는 것을 이용한 공격이다. 예를 들어 타겟 프로세스가 A 라는 메모리 주소를 액세스하고 있다면, 공격 프로세스가 같은 칩에 존재할 때 A 라는 메모리 주소를 보다 빨리 액세스할 수 있다. 반대로 타겟 프로세스에서 A 라는 주소를 사용하지 않고 있는 경우라면, 메모리에서 CPU 로 데이터를 가져오는 과정으로 인해 로드 과정이 느려진다. 이를 활용한 Flush+Reload[14], Prime+Probe[15]가 가장 많이 사용되며, 두 공격의 핵



(그림 1) 데이터 스니핑을 활용한 모델 탈취 공격[14]

심은 타겟 프로세스가 접근한 코드, 즉 실행된 코드를 추적하여 콜그래프를 뽑아낼 수 있다는 것이다. Menghjia et al[4]는 딥러닝 모델이 사용하는 행렬 곱셈 연산이 GEMM 알고리즘에 의해 계산되며 인공지능 모델의 연산 프레임워크로 사용되는 것에 착안하여 모델 구조를 추출해 내는 방법을 제안하였다.

GEMM(Generalized Matrix Multiply)은 OpenBLAS, MKL 등 프레임워크에서 동작하게 된다. 이때 GEMM 은 행렬을 더 작은 행렬 쪼개는 과정을 반복하는 최적화를 진행한다. 최적화 과정은 4 개의 loop 로 이루어져 있고 각 loop 마다 특정 연산과 관련된 함수가 실행된다. 예를 들면 알고리즘 1 의 loop 4 는 oncopy, kernel 두 함수의 반복 횟수와 동일하게 된다. 이를 캐시 부채널 공격에 활용하여 itcopy, oncopy, kernel 함수의 실행 순서를 추적하면 GEMM 연산에서 각 루프의 반복 횟수를 알 수 있다. 이를 통해 행렬 연산의 크기를 구하고 행렬 연산의 크기를 이용하면, 각 레이어의 크기를 알 수 있으므로 모델의 구조를 파악할 수 있게 된다.

4. 데이터 스니핑을 통한 모델 탈취 공격

딥러닝 모델의 연산은 CPU 에서 이루어지는 경우도 있지만 연산량이 많아질 경우 GPU 를 더 많이 사용한다. GPU 가 연산되는 과정은 CPU 에서 GPU 에 PCIe 패킷을 활용하여 필요한 연산과 연산자를 전달하면 이를 GPU 에서 연산한 뒤 다시 메인 메모리로 전달해주는 과정으로 이루어진다. 하지만 대부분의 시스템 환경에서 CPU 에서 GPU 또는 GPU 에서 CPU 로 데이터가 전달되는 과정에서 암호화 또는 아무런 방어 장치가 적용되어 있지 않다. 따라서 이 부분의 패킷을 데이터 스니핑 할 수 있다.

Zhu et al[14]는 모델 연산을 하나씩 실행시켜 이에 따른 PCIe 패킷의 패턴과 페이로드를 분석하고, 해당하는 연산과 데이터가 패킷의 페이로드의 어느 부분이 의미하는 지를 리버스 엔지니어링하였다. 이를 통

하여 어떠한 모델이든지 해당 PCIe 프로토콜을 사용하는 시스템의 경우 해당 패킷을 스니핑하여 모델을 복구할 수 있다. 이때 앞선 캐시 부채널 공격의 경우, 모델의 구조만을 복원할 수 있었지만, 해당 공격은 데이터 자체를 스니핑하기 때문에 모델의 가중치까지 추출할 수 있어 타켓 모델과 완벽하게 동일한 모델을 위조할 수 있다.

해당 공격은 모델을 완벽하게 복원할 수 있지만 엄청난 엔지니어링 노력과 시간이 필요하다.

5. 최신 쿼리 기반 모델 탈취 공격과의 비교

최신 쿼리 기반 모델 탈취 공격의 경우[11,12] 최대한 적은 쿼리로 높은 정확도를 가지는 위조 모델을 목표로 한다. Pal, Soham, et al[12]은 Active Learning 의 학습 방식을 사용하여 적은 쿼리로 높은 정확도를 달성하였다. Active Learning 은 학습을 하는 과정에서 Non-labeled dataset 에 대해서 정보량이 많은 데이터를 골라내려 한다. 이때 Uncertainty, K-centering, Adversarial Example 등 다양한 방법이 사용된다. Orekondy et al[6]와의 차이점은 타겟 모델에 대해 데이터를 샘플링하는 것이 아니라 위조 모델에 대해 데이터를 샘플링하는 것이다. 따라서 타겟 모델에 대한 데이터를 확인하기 위한 일련의 과정이 필요없기 때문에 쿼리 수가 훨씬 적게 필요하다.

Yu, Honggang, et al[11]은 위조 모델에 대해 AE 를 생성하고 해당 AE 를 재학습하여 변화한 위조 모델에 대해 다시 AE 를 생성해 재학습하는 과정을 반복하여 모델 탈취 공격을 진행한다. 이는 타겟 모델에 의사결정면에 점점 다가가는 방향으로 AE 가 생성되기 때문에 쿼리 수를 효과적으로 줄일 수 있다.

이와 같이 쿼리 기반 모델 탈취 공격은 재학습을 통해 모델의 기능성을 탈취하는데 초점이 맞추어 있다. 그렇기에 쿼리 기반이 아닌 모델 탈취 공격과 정량적인 비교는 어렵다. 따라서 두 공격 방법의 장단점 비교를 통해 어떻게 활용할 수 있는 지에 대해 설명한다.

기존 쿼리 기반의 탈취 공격은 모델의 구조, 모델의 가중치에 대한 분석은 안 이루어졌다. 데이터 스니핑을 활용한 공격[14]의 경우에는 가중치를 바로 뽑을 수 있기 때문에 쿼리 기반 공격을 활용하기는 어렵다. 하지만 캐시 부채널 공격을 활용한 모델 탈취 공격[4]은 모델의 구조를 뽑기 때문에 쿼리 기반 공격에 활용할 수 있다. 왜냐하면 구조가 동일할 때를 가정한 공격들이 많기 때문에 실제 공격에 대한 제한사항들을 완화할 수 있기 때문이다.

6. 결론

본 논문에서는 쿼리를 활용하지 않는 모델 탈취 공격에 대해 조사하고 쿼리 기반 공격과 비교, 분석하였다. 쿼리 기반 공격과 쿼리를 활용하지 않는 공격은 각 공격마다 공격 벡터가 다르고 얻어낼 수 있는 정보에 차이가 있기 때문에 정량적으로 어떠한 공격이 실제 환경에서 더욱 강력하다고 말하기 어렵다.

따라서 각 공격이 가지고 있는 공격 벡터와 얻어낼 수 있는 정보들을 분석하여 혼합적으로 사용될 때 강력한 위력을 보일 것으로 예상된다.

7. ACKNOWLEGEMENT

이 논문은 2021 년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원(NRF-2020R1A2B5B03095204)과 2021 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(No.2018-0-00230, (IoT 총괄/1 세부) IoT 디바이스 자율 신뢰보장 기술 및 글로벌 표준기반 IoT 통합보안 오픈 플랫폼 기술개발 [TrusThingz 프로젝트]). 그리고 2021 년도 BK21 FOUR 정보기술 미래인재 교육연구단에 의하여 지원되었음. 또한 본 연구는 반도체 공동연구소 지원의 결과물임을 밝힙니다.

참고문헌

[1] Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014).

[2] Shokri, Reza, et al. "Membership inference attacks against machine learning models." 2017 IEEE Symposium on Security and Privacy (SP). IEEE, 2017.

[3] Jagielski, Matthew, et al. "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning." 2018 IEEE Symposium on Security and Privacy (SP). IEEE, 2018.

[4] Yan, Mengjia, Christopher W. Fletcher, and Josep Torrellas. "Cache telepathy: Leveraging shared resource attacks to learn {DNN} architectures." 29th {USENIX} Security Symposium ({USENIX} Security 20). 2020.

[5] Hu, Xing, et al. "Deepsniffer: A dnn model extraction framework based on learning architectural hints." Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems. 2020.

[6] Orekondy, Tribhuvanesh, Bernt Schiele, and Mario Fritz. "Knockoff nets: Stealing functionality of black-box models." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[7] Papernot, Nicolas, et al. "Practical black-box attacks against machine learning." Proceedings of the 2017 ACM on Asia conference on computer and communications

security. 2017.
[8] Popov, Igor V., Saumya K. Debray, and Gregory R. Andrews. "Binary Obfuscation Using Signals." USENIX Security Symposium. 2007.
[9] Chen, Shuai, et al. "Chip-level anti-reverse engineering using transformable interconnects." 2015 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS). IEEE, 2015.
[10] Geethanjali, D., et al. "AEON: android encryption based obfuscation." Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy. 2018.
[11] Yu, Honggang, et al. "Cloudleak: Large-scale deep learning models stealing through adversarial examples." Proceedings of Network and Distributed Systems Security Symposium (NDSS). 2020.
[12] Pal, Soham, et al. "Activethief: Model extraction using active learning and unannotated public data." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34. No. 01. 2020.
[13] Zhu, Yuankun, et al. "Hermes Attack: Steal DNN Models with Lossless Inference Accuracy." arXiv preprint arXiv:2006.12784 (2020).
[14] Yarom, Yuval, and Katrina Falkner. "FLUSH+ RELOAD: A high resolution, low noise, L3 cache side-channel attack." 23rd {USENIX} Security Symposium ({USENIX} Security 14). 2014.
[15] Liu, Fangfei, et al. "Last-level cache side-channel attacks are practical." 2015 IEEE symposium on security and privacy. IEEE, 2015.