랜섬웨어 방어를 위한 딥러닝 기반의 데이터 암호화 탐지 기법

강민지○*, 원종훈*, 박지성, 김지홍
서울대학교 컴퓨터공학부
{kyang, barber}@snu.ac.kr, {jspark, jihong}@davinci.snu.ac.kr

A Deep Learning-Based Encrypted Data Detection Technique for Ransomware Defense

Minji Kang○, Jonghoon Won, Jisung Park, Jihong Kim
Department of Computer Science and Engineering, Seoul National University

요 약

최근 랜섬웨어에 의한 피해가 심각해짐에 따라, 랜섬웨어 공격을 실시간으로 감지하고 방어하는 기술 개발의 중요성이 높아지고 있다. 전통적인 랜섬웨어 탐지 기법이 갖는 근본적 한계를 극복하기 위해 저장장치 내부 수준의 데이터 보존 및 복구 기법이 제안되었으나, 무분별한 데이터 보존으로 인해 저장공간 부하를 크게 증가시킬 수 있다는 한계점이 존재한다. 본 논문에서는 랜섬웨어의 공격에 대비해 보존해야 하는 데이터 선정 정확도를 높일 수 있는 딥러닝 기반의 데이터 암호화 여부 판단 기법을 제시한다. 제안하는 기법은 컨볼루션 신경망 및 순환 신경망을 활용하여 저장장치 내부 수준에서 상위 계층의 정보 없이 높은 정확도로 데이터의 암호화 여부를 판단한다. 실험 결과, 컨볼루션 신경망과 순환 신경망 기반 기법은 각각 93.34%, 82.54%의 높은 정확도로 데이터의 암호화 유무를 판별하였으며, 0에 가까운 부정 오류율을 달성하였다.

1. 서 론

최근 랜섬웨어의 공격 및 이에 따른 피해가 나날이 심각해짐에 따라, 컴퓨터 보안 분야에서 랜섬웨어의 공격을 감지하고 이에 대처하기 위한 기술의 중요성이 커지고 있다. 기존 랜섬웨어의 공격 패턴을 미리 입력해두고 이를 기반으로 공격을 탐지해내는 시그니처 기반의 기법이 널리 활용되고 있으나 [1], 새로운 공격 패턴의 랜섬웨어 공격에 대한 취약점을 갖는다 [2]. 또한, 운영체제 수준에서 별도의 소프트웨어를 사용하는 랜섬웨어 탐지 및 차단 기법은 커널 권한을 탈취한 랜섬웨어가 탐지 프로그램 자체를 중단시킴으로써 랜섬웨어 방어를 쉽게 무력화할 수 있다는 근본적인 한계점을 지닌다 [3].

전통적인 랜섬웨어 방어 기법의 한계점을 극복하기 위해, 낸드 플래시 기반 저장장치의 특성을 이용한 저장 장치 내부 수준의 랜섬웨어 방어 기법(FlashGuard)이 제안되었다 [3]. 낸드 플래시 기반 저장장치는 쓰기 명령을 수행할 때 기존 페이지를 덮어쓰는 대신 데이터를 빈 페이지에 기록하고, 기존에 기록된 페이지는 추후에 가비지 컬렉션(Garbage Collection, GC)을 통해 재사용한다. 따라서, GC 수행 시 랜섬웨어에 의해 공격받은 것으로 의심되는 특정 페이지를 수집하지 않도록 설정함으로써, 피해 데이터를 저장장치 내부 수준에서 안전하게 보존하고 복원할 수 있다 [3]. FlashGuard는 랜섬웨어에 의해 공격받은 데이터를 호스트 시스템과 독립적인 저장장치 내부에 보존하므로, 호스트 시스템의 커널 권한이 랜섬웨어에 의해 탈취되더라도 랜섬웨어의 공격에 따른 피해를 사후에 복구해낼 수 있다 [3].

한편, FlashGuard와 같은 접근방식에서는 사후 피해 복구를 위해 보존하는 데이터가 저장장치의 성능과 수명에 악영향을 줄 수 있으므로, 보존할 데이터를 효과적으로 선정해야 한다. 이는 보존할 데이터의 크기가 증가할수록 GC 시 복사하는 페이지 또한 증가하여 저장장치의 쓰기 증폭률(Write Amplification Factor, WAF)을 악화시키기 때문이다. 기존 연구에서는 랜섬웨어가 공격할 때 대상 데이터를 읽는다는 사실에 기반하여, 한 번이라도 읽힌 적이 있는 데이터를 GC 대상에서 일정 시간 제외하는 정책을 이용하였다. 이와 같은 정책은 피해 데이터가 GC에 의해 삭제될 확률을 낮출 수 있으나, 사용자가 읽은 데이터까지 보존하여 심각한 오버헤드를 초래할 수 있다.

본 논문에서는 위와 같은 저장장치 내부 수준의 데이터 보존 및 복구 방식에서 보존할 데이터 선정의 정확도를 높일 수 있는 데이터 암호화 여부 판단 기법을 제시한다. 랜섬웨어는 공격 시 대상 데이터를 읽는 동작에 더해 데이터를 암호화하고 저장하는 동작 또한 필수적으로 수반한다. 따라서 보존할 데이터를 선정할 때 데이터의 접근 이력과 함께 데이터 접근 전후에 발생한 암호화된 데이터의 기록 여부를 고려한다면 피해 데이터를 선별적으로 보존하여 저장장치의 성능 및 공간 부하를 최소화할 수 있다. 제안하는 데이터 암호화 여부 판단 기법은 피해 데이터의 손실을 방지하기 위해 낮은 부정 오류율(False Negative Rate, FNR)을 보장해야 하며, 저장장치 내부 수준에서 파일 시스템의 정보 없이 수행되어야 한다.

이를 위해, 딥러닝 기법인 컨볼루션 신경망(Convolutional

* 공동 제1저자.

Neural Network, CNN)과 순환 신경망(Recurrent Neural Network, RNN)에 기반하여 주어진 데이터의 암호화 유무를 판단하는 모델을 설계하였다. 암호화된 데이터는 높은 엔트로피를 가지므로 [4], 모델이 데이터의 엔트로피를 파악하여 암호화 여부를 판단할 수 있을 것으로 예상하였다. CNN을 사용할 때, 데이터의 바이트 값 빈도 분포(Byte Frequency Distribution, BFD)를 2차원으로 변형하여 입력하기 때문에 데이터의 엔트로피를 다양한 바이트 조합의 측면에서 평가할 수 있을 것으로 기대하였다. 또한, RNN을 사용하여 데이터의 바이트 값을 순차적으로 입력하여, 데이터 전체의 엔트로피를 반영할 수 있을 것으로 기대하였다.

실험 결과, 제시한 CNN과 RNN 모델은 각각 93.34%, 82.54%의 높은 정확도로 데이터의 암호화 여부를 판단하였다. CNN과 RNN 모델의 FNR은 각각 1.815×10^{-2}%와 0%로 모든 모델은 매우 낮은 FNR을 보였다. 모델들의 긍정 오류율(False Positive Rate, FPR)과 FNR 사이에는 보상 관계(trade-off)가 존재했으며, 특히 CNN의 경우 결정 경곗값(threshold)을 조절하여 이를 통제할 수 있었다.

본 논문의 구성은 다음과 같다. 2장에서는 제시한 CNN과 RNN 기반의 데이터 암호화 여부 탐지 모델을 설명한다. 이어 3장에서 제안한 CNN과 RNN 기반 모델의 학습 결과를 평가하고 비교한 후, 4장에서 결론을 맺는다.

2. 딥러닝을 통한 데이터 암호화 여부 탐지

암호화 알고리즘은 복호화가 어렵게 설계되었기 때문에, 암호화된 데이터는 엔트로피가 높고, 바이트 값이 균일하게 분포한다는 특성을 가진다 [4]. 제안하는 기법에서는 암호화된 데이터가 갖는 특성을 학습시켜 데이터의 암호화 여부를 결정하고자 하였다. 이를 위해 CNN과 RNN의 신경망을 이용한 두 개의 데이터 암호화 결정 모델을 설계하고 비교·평가하였다.

2.1. CNN 기반의 데이터 암호화 여부 탐지

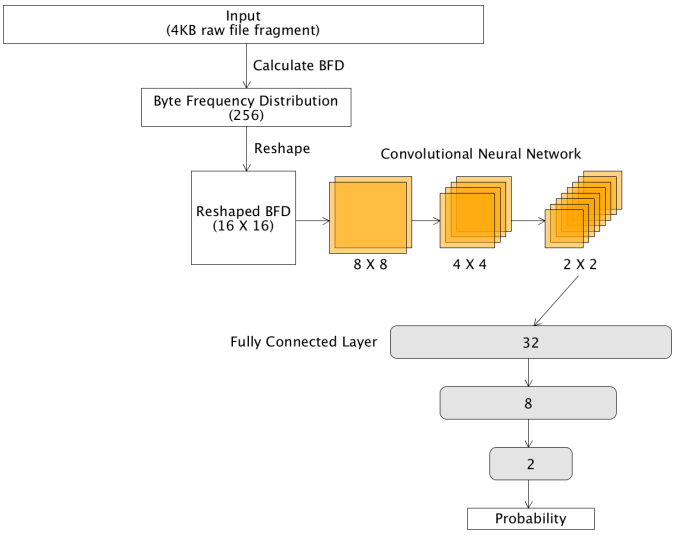

그림 1. 데이터 암호화 여부 탐지를 위한 CNN 구조

그림 1은 데이터 암호화 여부를 판단하기 위해 설계한 CNN의 구조를 보여준다. CNN은 다양한 채널에서의 필터 연산을 통해 이미지, 비디오 등의 고차원 데이터를 다양한 측면에서 분석하여, 높은 정확도를 보이는 신경망이다. 주어진 데이터의 엔트로피를 기반으로 암호화 유무를 분석하기 위해, CNN을 사용함으로써 바이트 별 빈도 분포(Byte Frequency Distribution, BFD)를 다양한 바이트 조합 측면에서 분석하여 암호화 유무를 판단할 수 있을 것으로 기대하였다.

그림 1에서 보이는 바와 같이, 암호화 유무를 판단해야 하는 4KB 데이터가 입력되면, 이를 BFD로 요약한다. CNN을 사용하여 BFD를 분석하기 위해, 1×256의 1차원 형태인 BFD를 16×16의 2차원 형태로 바꾼다. 2차원의 BFD는 3×3 크기의 필터 연산과 간격 2의 풀링 연산을 거치며 채널 수가 두 배씩 증가한다. 이 과정을 세 번 반복하여, BFD는 2×2 크기의 여덟 채널 데이터로 정리된다. 정리된 BFD 데이터는 세 층의 완전 연결 층을 거쳐 최종적으로 두 개의 값으로 정리되는데, 이는 각각 주어진 데이터가 암호화된 조각일 확률과 그렇지 않을 확률을 의미한다.

모델을 학습시키는 과정에서, 손실 함수로 식 1의 Cross Entropy 함수를 식 2와 같이 수정하여 이용한다. Cross Entropy 함수의 계수를 조절하여 FNR이 작아야 한다는 목표를 표현하고 그 강도를 조절할 수 있다. 즉, a_1을 a_2보다 크게 설정함으로써 부정 오류(False Negative)에 비중을 두어 처벌할 수 있다.

$$C(\hat{y}, y) = -y \log \hat{y} - (1-y) \log (1-\hat{y}) \quad (1)$$

$$C(\hat{y}, y) = -a_1 y \log \hat{y} - a_2 (1-y) \log (1-\hat{y}) \quad (2)$$

2.2. RNN 기반의 데이터 암호화 여부 탐지

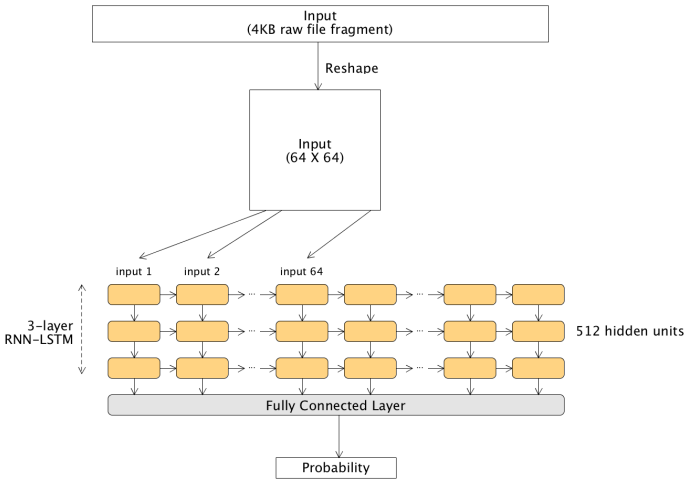

그림 2. 데이터 암호화 여부 탐지를 위한 LSTM-RNN의 구조

그림 2는 데이터 암호화 여부를 판단하기 위해 설계한 LSTM-RNN의 구조를 보여준다. RNN은 자연어, 음악 등의 시계열 데이터를 처리하기 위하여 도입된 신경망으로, 층의 출력을 해당 층의 입력으로 되먹임한다. RNN은 시계열 데이터가 긴 경우 학습을 진행함에 따라 기울기가 소실되는 문제를 가지는데, Long Short Term Memory(LSTM)는 RNN에 Forget Gate를 도입하여, 중요한 데이터만을 오래 유지함으로써 문제를 해결하였다. 본 연구에서는 LSTM-RNN을 적용하여, 신경망이 데이터의 바이트 값이 순차적으로 어떻게 변하는지 설명할 수 있을 것으로 기대하였다. 또한, RNN 모델이 별다른 요약 과정 없이 데이터를 입력받기 때문에, 하드웨어 상에서 온라인으로 구현할 때 CNN 모델에 비해 높은 성능을 달성할 수 있을 것으로 보았다.

그림 2에서 보이는 바와 같이, 먼저 4KB의 데이터가 64바이트 단위로 나뉘어 신경망에 예순네 번 입력된다. 그다음 세 개의 LSTM-RNN 층과 완전 연결 층을 지난 후, 두 개의 값으로 최종 출력된다. 최종 출력된 두 개의 값은 CNN과 마찬가지로 암호화된 파일일 확률과 암호화되지 않은 파일일 확률로 해석할 수 있다. LSTM-RNN 각 층은 512개 단위 요소로 구성된다. 손실 함수는 RNN 신경망에서 일반적으로 사용되는 평균 제곱 오차 (Mean Square Error)를 사용하였다.

3. 실험 결과

3.1. 실험 환경

실험에는 21GB, 21,000개의 파일로 구성된 govdocs-selected 데이터셋을 이용하였다 [5]. 모든 파일을 AES-256 알고리즘으로 암호화하여 암호화된 파일 21,000개를 얻었다. 42,000개의 모든 파일을 4KB 바이트로 조각내어 80%는 학습 데이터, 20%는 시험 데이터로 사용하였다. 학습과 시험에 사용된 4KB 데이터의 구성은 표 1과 같다.

	비암호화 데이터 (개)	암호화 데이터 (개)
학습 데이터	4,394,268	4,394,268
시험 데이터	1,098,567	1,098,567

표 1. 학습과 시험에 사용된 4KB 데이터의 구성

모든 모델은 TensorFlow 라이브러리로 구현하였다. 학습은 학습률 10^{-4}로 진행되었고, CNN의 완전 연결 층과 RNN에서 과적합을 방지하기 위하여 0.5 비율의 드롭아웃을 사용하였다.

3.2. 실험 결과

표 2는 CNN 및 RNN을 기반으로 하는 데이터 암호화 여부 탐지 기법의 정확도를 요약하여 보여준다.

	총 정확도 (%)	FPR (%)	FNR (%)
$\mathrm{CNN}_{\theta=0.5}$	93.34	13.32	1.815×10^{-2}
$\mathrm{CNN}_{\theta=0.1}$	91.84	16.37	2.335×10^{-4}
RNN	82.54	34.91	0

표 2. 실험한 모델들의 암호화 여부 결정 정확도

두 모델 모두 높은 암호화 결정 정확도와 낮은 FNR을 달성할 수 있음을 확인하였다. 기반 딥러닝 기법에 따라 다른 특성을 보였는데, CNN 기반 기법은 93.34%의 높은 정확도를 보였으나 FNR이 1.815×10^{-2}%로 RNN에 비해 높게 나타났다. RNN 기반 기법은 총 정확도가 82.54%로 CNN 기반 비법보다 낮은 대신 FNR을 0으로 만들 수 있었다.

CNN의 FNR을 더욱 낮추기 위하여, 데이터가 암호화되지 않았을 경우를 보다 보수적으로 판단하도록 했다. 이를 위해 입력된 데이터가 암호화되었을 확률이 0.1을 넘으면 데이터가 암호화되었다고 판단하도록 결정 경곗값 θ를 0.5에서 0.1로 낮추었다. 그 결과, 총 정확도는 91.84%로 다소 낮아졌으나, FNR을 기존 실험 결과의 1.29% 수준인 2.335×10^{-4}%로 줄일 수 있었다. 이 결과는 모델을 사용하는 용도에 따라 결정 경곗값을 조절함으로써, 같은 모델을 사용하면서도 FNR과 FPR의 보상 관계(trade-off)를 유동적으로 조절할 수 있음을 의미한다. FPR의 변화 폭이 FNR의 변화 폭보다 크기 때문에, 총 정확도에는 FNR보다 FPR이 미치는 영향이 더 크다. 이에 따라 데이터 보존의 안정성을 위해서는 정확도를 다소 희생하여 낮은 FNR을 갖도록 하는 한편, 데이터 보존의 부하를 최소화하기 위해서는 약간의 안정성을 희생하여 정확도를 높이는 식의 요구조건에 따른 모델 선정이 필요할 것으로 판단된다.

4. 결론 및 향후 연구

본 논문에서는 저장 장치 내부 수준에서 랜섬웨어 방어를 위한 데이터 암호화 여부 탐지 기법을 제안하였다. 파일 시스템의 정보 없이 높은 정확도를 달성하기 위하여 CNN 및 RNN 등의 딥러닝 기반 탐지 기법을 설계·평가하였으며, 모든 모델에서 80% 이상의 높은 정확도 및 1.815×10^{-2} % 이하의 낮은 FNR을 달성하였다. 특히 RNN 기반 기법에서는 0 FNR을 달성하였고, CNN 기반 기법의 경우 결정 경곗값을 조절함으로써 FNR의 크기를 사용자가 통제할 수 있었다. 또한, 모든 모델에서 FNR과 FPR 사이의 보상 관계가 존재함을 확인하였으며, 이는 모델의 사용자가 사용 용도에 따라 낮은 FNR과 높은 총 정확도 중 원하는 특성을 가지는 모델을 유동적으로 선택하여 이용할 수 있음을 의미한다.

본 연구에서 제시한 모델을 통해 저장 장치 내부 수준에서 랜섬웨어에 효율적으로 대처하기 위해서는 연산량 혹은 모델의 결정 시간을 정량화할 필요가 있다. 향후 연구로서, 저장장치 내부에서 제안된 탐지 기법을 실시간으로 활용할 수 있도록 효과적인 하드웨어 가속 모듈을 구현 및 평가할 예정이며, 펌웨어 수준에서 이를 기반으로 한 데이터 보존 및 GC 알고리즘을 개발할 예정이다.

감사의 글

이 연구를 위해 연구장비를 지원하고 공간을 제공한 서울대학교 컴퓨터연구소에 감사드립니다. 이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2018R1A2B6006878).

참 고 문 헌

[1] D. Xu, J. Ming, and D. Wu, "Cryptographic Function Detection in Obfuscated Binaries via Bit-Precise Symbolic Loop Mapping," 2017 IEEE Symposium on Security and Privacy (SP), pp. 921－937, 2017.

[2] A. Kharraz, S. Arshad, and C. Mulliner, "UNVEIL: A Large-Scale, Automated Approach to Detecting Ransomware," 25th USENIX Security Symposium, pp. 757－772, 2016.

[3] J. Huang, J. Xu, X. Xing, P. Liu, and M. K. Qureshi, "FlashGuard: Leveraging Intrinsic Flash Properties to Defend Against Encryption Ransomware," 2017 ACM SIGSAC Conference, pp. 2231－2244, 2017.

[4] Y. Dodis and A. Smith, "Entropic Security and the Encryption of High Entropy Messages," 2nd International Conference on Theory of Cryptography, pp. 556－577, 2005.

[5] N. S. Alamri and W. H. Allen, "A Comparative Study of File-Type Identification Techniques," IEEE SoutheastCon 2015, 2015