네트워크 패킷 레벨에서 알려진 실행 파일 식별 및 차단 연구

조용수*, 이희조*
*고려대학교 컴퓨터정보통신대학원 소프트웨어보안학과
e-mail:emptyskycho@korea.ac.kr

Research on the identification and blocking of known executalbe files at the network packet level

Yongsoo Jo*, heejo Lee*
*Dept. of Information Technology, Korea University

요 약

최근의 사이버 침해 사고는 공격 대상을 지정하여 지속적으로 공격을 시도하는 APT(Advanced Persistent Threat) 와 랜섬웨어(Ransomware) 공격이 주를 이룬다. APT 공격은 dirve by download 를 통하여 의도하지 않은 파일의 다운로드를 유도하고, 다운로드 된 파일은 역통신채널을 만들어 내부 데이터를 외부로 유출하는 방식으로, 공격에 사용되는 악성 파일이 사용자 모르게 다운로드 되어 실행된다. 랜섬웨어는 스피어 피싱(Spear-phishing) 과 같은 사회공학기법을 이용하여 신뢰 된 출처로 유장 된 파일을 실행하도록 하여 주요 파일들을 암호화 한다. 때문에 사용자와 공격자 사이 네트워크 중간에 위치한 패킷 기반의 보안 장비들은 사용자에 의해 다운로드 되는 파일들을 선제적으로 식별하고, 차단하여 침해 확산을 방지 할 수 있는 방안이 필요하다. 본 논문에서는 네트워크 패킷 레벨에서 알려진 악성파일을 식별하고 실시간 차단하는 방안에 대하여 연구하고자 한다.

1. 서 론[1)]

대부분의 기업들은 내부 네트워크를 외부의 불법적인 사용자의 침입으로부터 안전하게 보호하기 위한 네트워크 보안 장비, 방화벽(Network FireWall), IPS(Intrusion Prevention System) 등을 사용한다. 이러한 보안 장비들은 모두 패킷(packet)을 기반으로 검사하는 방식으로, 1Kbyte-2Kbyte (ethernet 기준 1518 byte) 패킷 정보로 보안 정책을 판단하여 차단 할 것인지, 전달 할 것인지를 판단하게 된다. 때문에, 1.5Kbyte 의 패킷 사이즈 보다 훨씬 큰 파일 기반 공격으로 이루어진 APT 와 랜섬웨어는 패킷 기반 보안 장비로 대응하기 어렵다. 시그니처 기반으로 각 파일들의 signature, pattern 를 찾아 등록 할 수 있지만, 모든 파일을 각각 분석하여 unique signature 찾아야 하기 때문에, 패킷 기반 보안 장비에서는 거의 사용하지 않는다. (패킷 fragment 에 의한 시그니처의 분리도 시그니처 탐지 방식 사용을 어렵게 한다.) 이러한 문제를 해결하기 위해 네트워크 패킷 단위를 기반으로 악성 파일을 식별할 수 있는 방안에 대해 연구하였으며, 제안 방식을 시험을 통하여 연구 결과를 살펴보고자 한다. 본 논문에서는 악성파일을 식별하고 탐지 및 차단하는 것을 목표로 하며, APT 공격와 랜섬웨어 이용되는 파일들이 EXE, DLL, SCR 같은 실행파일, 즉 PE(Portable Executable) 파일 형식을 사용하고 있으므로, PE(Portable Executable) 형태의 실행 가능한 파일을 대상으로 범위를 한정한다.

2. 관련 연구

알려진 악성 파일을 식별하기 위해 자주 사용되는 기술은 해시(hash) 이다. 파일 해시 기술은은 파일의 전체 내용을 SHA1이나 MD5와 같은 해시 연산을 통해 하나의 해시 값으로 생성하는 방식으로, 완전한 파일 전체를 대상으로 해쉬하여 나온 값을 통하여 식별한다. 해시를 이용한 다른 파일 식별 방법으로는 파일 부분 해시 기술로, 파일을 나누는 기준에 따라 FLC(Fixed Length Chunking)과 VLC(Variable Length Chunking)으로 나뉜다. 이러한 방식은 파일의 유사

성을 비교하기 위하여 주로 사용되고, 이 역시 완전한 파일 전체를 대상으로 block 단위와 위치에 따라 나누어 해시 한 후 대표값을 통하여 파일을 식별한다.

인터넷 사용자와 서버 사이에 위치하는 네트워크 보안 장비는 그 위치와 서비스를 지원하는 특성상 속도에 민감하다. 때문에 기존 전체 파일 해시 방식을 활용하여 악성파일을 식별한다면 네트워크 처리 성능에 많은 영향을 미치기게 된다. 다음은 악성파일 탐지를 위한 검사 시간 영향도이다.

$$\text{Scanning Time} = \frac{\text{Packet of the file recombination} \times \text{Size of the file objects} \times \text{Hash Time}}{\text{Speed of Processor}}$$

Fig 1. 파일전체 해시를 이용한 검사 시간

검사 대상 파일의 모든 패킷을 수집하여 완전한 파일로 재조합 한 후 해시 검사를 완료하여 탐지하고 차단한다면, 네트워크 중간의 보안 장비는 패킷을 수집 할 많은 메모리 공간과 파일 재조합 비용으로 많은 네트워크 딜레이가 발생하게 될 것이다.

3. 제안 방안

서론에서 언급한 것과 같이 악성 파일들이 EXE, DLL, SCR 와 같은 PE(Portable Executable) 파일 형식을 사용하고 있으며, PE(Portable Executable) 은 PE header 에 파일을 특정 지을 수 있는 많은 정보들이 들어있다. PE 파일의 대표적인 정보로는 TimeDate Stamp, Size of Code, Address of EntryPoint, CheckSum, Size Of File, Certificate address 등이 포함되며, 이 모든 정보는 파일의 첫 1Kbyte 안에 존재하게 된다.

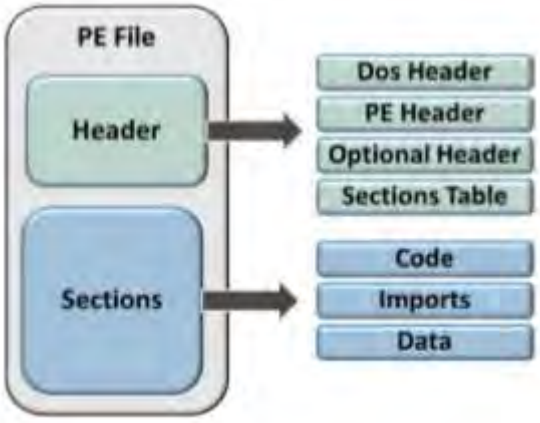

Fig 2. PE 파일 구조

즉, MS-DOS 2.0 Section, PE File Header, Windows Specific Fields, Data Diorectories, Certificate Table 정보 등 PE 파일을 특정 지을 수 있는 정보의 대부분이 파일의 첫 번째 패킷에 들어있게 된다. 이것은 PE 파일 전체 해시를 첫 번째 패킷 데이터로 대체 할 수도 있다는 가정을 하게 되었다.아래는 하나의 패킷 해시로 대응 했을 때 탐지를 위한 검사 시간 영향도 이다.

$$\text{Scanning Time} = \frac{\text{One piece of packet} \times \text{Hash Time}}{\text{Speed of Processor}}$$

Fig 3. PE헤더 해시를 이용한 검사 시간

4. 시험 및 결과

VirusTotal 에서 2020년 1월부터 3월기간 동안 악성파일로 탐지(20개 이상의 백신에 탐지) 된 1천개의 파일과 인터넷에서 무작위로 수집 된 인터넷 실행 파일(정상) 9천개를 대상으로 파이썬 스크립트를 통하여 네트워크로 전송하였다. 네트워크 중간의 보안 장비에서 패킷을 모니터하여, PE 파일의 경우 첫번째 패킷을 해쉬하여 로깅하였고, 이를 이용하여 충돌률을 비교 검사하였다.

Fig 4. 해시 충돌 시험 환경

시험 결과,

바이러스 토탈 악성 1000 파일 - 충돌파일 0 | 인식 1000
네트워크 임의 정상 9000 파일 - 충돌파일 3 | 인식 8994

총 1만개의 악성, 정상 파일을 분석 한 결과 악성파일에서 0건, 정상 파일에서 1건의 충돌이 있었다. 정상파일 1건의 경우 분석결과 빵집으로 압축 된 실행파일 형태였으며, 동일한 압축 포맷, 즉 동일한 빵집 PE 헤더를 통하여 내부 서로 다른 파일을 감싸고 있었다.

5. 결 론

시험을 통하여 PE (Portable Executable) 파일 기반 실행파을은 PE 헤더 정보를 해쉬하는 것만으로도 충분히 파일 인식이 가능하며, 이를 통하여 알려진 악성 파일의 확산을 FireWall 과 IPS 같은 패킷 기반 네트워크 보안 장비에서 선 차단 할 수 있을 것을 기대한다. 시험에서 예외로 발생하였던 실행파일 빵집 PE 포맷은 해당 프로그램의 특징으로 파일 패킹(file packing) 기술로 봐야 할 것이다. 다만, 빵

집의 경우 모두 동일 특징을 가지고 있어 예외처리가 가능하다. 이 후 시험에는 더 많은 파일들을 대상으로 시험하여 해시 충돌률을 면밀히 비교해 볼 예정이다.

참고문헌

[1] Breitinger, F. & Baggili, I. "File detection on network traffic using approximate matching.", Journal of Digital Forensics, Security and Law. Special Issue: 2014 ICDF2C / SADFE. 9(2): 23-36, 2014.

[2] Y. J. Cho, "Unknown Malware Detection Using File Reputation", Proceedings of the Korea Information Processing Society Conference, 376-379, 2015

[3] S. J. Oh, Ko and Y. W. Ko, "Triple Fixed Length Hashing Scheme for Similarity Search", Proceedings of KIIT Conference, 339-342, 2013

[4] Y. J. Yoo and S. J Kim and J. Kim, and Y. W. Ko, "File Similarity Evaluation System Using Rate-based Representative Hash Scheme", Korean Institute of Information Technology, 81-88, 2014

[5] Virustotal statistics, “virustotal malware statistics”, (https://www.virustotal.com/ko/statistics)