랜섬웨어 대응 및 데이터 유출 보호를 위한 파일 접근 로그 기반 파일 접근 제어 시스템

A File Access Control System Based on File Access Logs for Ransomware Response and Data Loss Prevention System

저자 (Authors)
이한수, 김동주, 이혁준, 황동혁
Hansu Lee, Dongju Kim , Hyukjoon Lee, Donghyuk Hwang

출처 (Source)
한국정보과학회 학술발표논문집 , 2021.6, 2054-2056 (3 pages)

발행처 (Publisher)
한국정보과학회
The Korean Institute of Information Scientists and Engineers

URL
http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE10583527

APA Style
이한수, 김동주, 이혁준, 황동혁 (2021). 랜섬웨어 대응 및 데이터 유출 보호를 위한 파일 접근 로그 기반 파일 접근 제어 시스템. 한국정보과학회 학술발표논문집, 2054-2056.

랜섬웨어 대응 및 데이터 유출 보호를 위한 파일 접근 로그 기반 파일 접근 제어 시스템

이한수 O1 김동주 1 이혁준 1 황동혁 2

1 광운대학교 컴퓨터정보공학부, 2 테르텐

lhs1438@gmail.com, gggg8657@gmail.com, hlee@kw.ac.kr, mask@teruten.com

A File Access Control System Based on File Access Logs for Ransomware Response and Data Loss Prevention System

Hansu Lee O1 Dongju Kim1 Hyukjoon Lee1 Donghyuk Hwang2

1Department of Computer and Information Engineering, Kwangwoon University

2Teruten

요 약

본 논문은 엔드포인트에서 실행된 프로세스의 파일 접근을 제어함으로써 랜섬웨어 공격으로부터 사용자를 보호하고 데이터 유출을 차단하는 시스템을 제안한다. 다수의 PC에서 파일 접근 로그를 수집하고 이를 바탕으로 파일 접근 화이트리스트를 생성하여 파일 접근을 제어한다. 파일 접근 로그 수집과 제어를 위해 파일 접근 API를 후킹하며, 파일 접근 화이트리스트는 ABAC(Attribute- Based Access Control) 정책 형식으로 구현한다.

1. 서 론

코로나19의 영향으로 원격 근무 환경이 확산되면서 전 세계적으로 사이버 보안의 중요성이 커지고 있다. 2020년 글로벌 기업의 데이터 유출 사고 주요 원인은 유형별로 악의적 공격(52%), 시스템 결함(25%), 인적 오류(23%) 등의 순서이며[1], 최근에는 랜섬웨어 공격과 데이터 유출이 동시에 진행되는 경우가 증가하고 있다[2]. 또한 2021년에는 전 세계 대기업과 중소기업 모두에서 네트워크 및 엔드포인트 보안, 식별/접근관리 부문 시장이 더욱 성장할 것으로 예상한다는 점에서 이를 확인할 수 있다[3]. 프로세스의 파일 I/O를 제어해 랜섬웨어를 포함한 모든 프로세스의 파일 접근을 모니터링할 수 있다. 이를 바탕으로 짧은 시간 내에 한 프로세스에서 잦은 파일 I/O 요청을 보내면 비정상적인 파일접근으로 판단하여 차단하거나[4], 중요한 파일들만 파일의 읽기/쓰기 권한을 제한하는 방식으로 랜섬웨어로부터 파일을 보호하려는 연구가 있다[5]. 하지만 이들 연구는 I/O interval 기반이라 오탐할 수 있다는 점, 보호하고자 하는 파일을 직접 선택해야 한다는 한계점을 가진다. 랜섬웨어가 데이터 유출을 동시에 진행하는 경우도 완전히 대처할 수 없다는 부분에서도 한계를 띈다.

본 논문에서는 프로세스에서 파일 접근 로그를 수집하여 ABAC 정책 형식의 파일 접근 화이트리스트를 생성하고, 이를 통해 프로세스의 파일 접근을 제어하여 랜섬웨어 등을 통한 데이터 유출을 보호하는 시스템을 제안한다. 시스템은 1) 로그 수집 서브시스템, 2) 파일 접근 분류 서브시스템, 3) 파일 접근 제어 서브시스템의 총 3가지로 구성되며, 파일 접근 제어 서브시스템의 파일 접근 환경을 로그 수집 서브시스템과 유사하도록 만들어 다수의 엔드포인트 PC에서 파일 접근을 제어한다. ABAC 정책을 사용하기 때문에 I/O interval이 잦은 일반 프로세스를 오탐하지 않으며, 보호할 파일을 선택할 일도 없다. 또한, 랜섬웨어뿐만 아니라 인가되지 않은 프로세스의 데이터 유출과 일반적인 프로세스의 비정상적인 파일 접근 또한 차단할 수 있다.

2. 파일 접근 로그 기반 파일 접근 제어 시스템 구조

2.1 전체 시스템 구조

본 논문에서 제안하는 파일 접근 제어 시스템은 그림 1과 같이 로그 수집 서브시스템, 파일 접근 분류 학습 서브시스템, 파일 접근 제어 서브시스템의 총 3가지 서브시스템들로 구성된다. 로그 수집 서브시스템은 안전성이 확인된 다수의 PC들로 구성되어 있으며, 해당 서브시스템에 있는 PC들에서 실행된 프로세스들로부터 파일 접근 로그를 수집한다. 이렇게 수집한 파일 접근들은 모두 안전한 것으로 간주한다. 파일 접근 분류 서브시스템에서는 로그 수집 서브시스템과 파일 접근 제어 서브시스템에서 수집한 파일 접근 로그를 바탕으로 파일 접근에 대한 화이트리스트를 만든다. 파일 접근 제어 서브시스템에서 수집한 차단 로그들은 필요한 경우 제어자의 판단에 따라 허용해 줄 수 있도록 한다. 파일 접근 제어 서브시스템 또한 다수의 PC로 이루어지며, 이들 PC에서는 화이트리스트를

* 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW 중심대학 지원사업의 연구결과로 수행되었음 (2017- 0- 00096).

바탕으로 파일 접근을 허용 또는 차단하는 여부를 결정한다. 파일 접근 차단이 이루어진 경우, 파일 접근 차단 로그를 기록한다. 로그 수집 서브시스템의 로그들에 차단 로그들 중에서 추가로 관리자가 허용한 로그들을 바탕으로 화이트리스트를 생성하기 때문에 로그 수집 서브시스템의 파일 접근 환경과 파일 접근 제어 서브시스템의 파일 접근 환경이 서로 유사해진다. 각 서브시스템은 역할이 각자 다르고, 해당 환경을 구성하는 PC의 수를 필요에 따라 유동적으로 조절할 수 있도록 하기 위해 서로 분리하였다. 각 서브시스템 간 파일 전송은 네트워크를 이용한다.

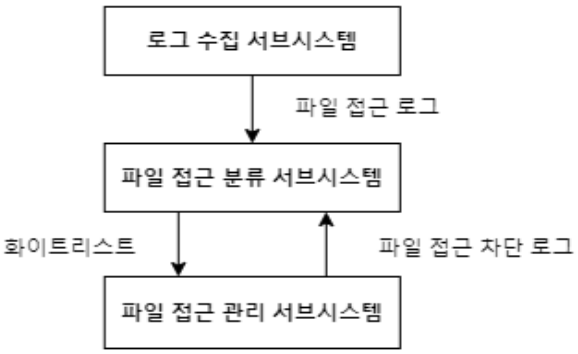

그림 1. 전체 시스템 구조

2.2 로그 수집 서브시스템

로그 수집 서브시스템은 그림 2와 같이 파일 접근을 로깅하는 DLL 파일, 프로세스의 실행을 감지해 해당 DLL을 주입하는 프로세스 모니터, 파일 접근 로그 파일들을 주기적으로 자동 전송하는 스크립트 3가지로 구성되어 있다. 우선 파일 접근을 로깅하는 DLL 파일은 ntdll.dll 내부의 ZwReadFile과 같은 파일 접근 API를 후킹하고 이를 로그로 남긴다. 로그에는 프로세스의 프로그램 이름, 파일 접근 함수 종류, 접근하려는 파일의 확장자 등의 정보가 존재한다. 로그는 JSON 형식을 사용한다. 프로세스 모니터는 운영체제가 시작되는 시점에 자동으로 실행되며, OS로부터 실행되는 프로세스를 탐지하여 미리 작성한 DLL을 주입한다. 마지막으로 수집한 파일 접근 로그 파일들을 작업 스케줄링 프로그램을 통해 주기적으로 파일 접근 분류 학습 서브시스템에 전송한다. 로그 파일들은 zip으로 압축되어 네트워크를 통해 전송된다.

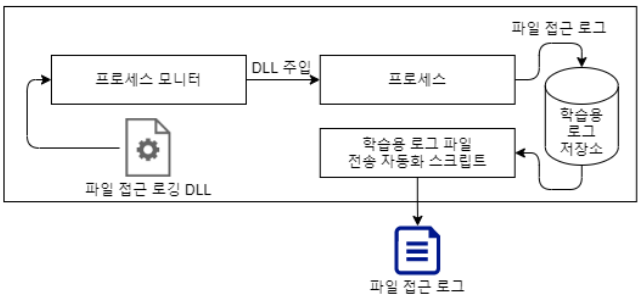

그림 2. 로그 수집 서브시스템 구조

프로세스 모니터에서는 프로세스를 런타임에 메모리 레벨에서 후킹하는데, 이러한 기술들 중 하나로 오픈소스인 Detours가 있다[6]. DLL 주입 후 그림 3과 같이 메모리가 수정되어 후킹 함수가 호출된다.

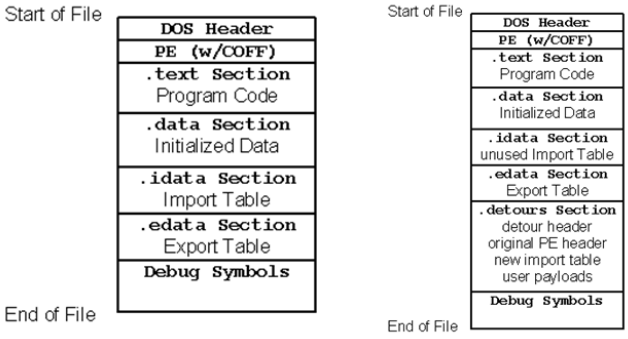

그림 3. DLL 주입 전후 프로세스 변화[6]

2.3 파일 접근 분류 서브시스템

본 시스템에서는 파일 접근 화이트리스트를 위해 ABAC(Attribute Based Access Control) 정책을 사용한다. ABAC란 어떤 대상들에게 동작을 요청하는 주체에게 주체와 대상, 그리고 환경의 속성 정보들을 바탕으로 구성된 정책을 따라 해당 동작을 승인하거나 거부하는 접근 제어 방법이다[7]. 파일 접근 분류 서브시스템은 두 가지 다른 서브시스템에서 수집한 파일 접근 로그들을 바탕으로 ABAC 정책 형태의 화이트리스트를 만든다. 주체는 파일 접근을 요청하는 NPE(Non- Person Entity)인 프로세스이고, 대상은 프로세스가 접근하고자 하는 파일이다. 해당 서브시스템에서는 우선 파일 접근 로그를 읽어 프로그램 이름과 파일 접근 함수 이름, 접근한 파일의 확장자 등의 속성 정보 튜플을 구성한다. 다음으로 이를 기존에 존재하는 화이트리스트에서 찾는다. 해당 튜플이 이미 화이트리스트에 존재하는 경우에는 중복하여 추가하지 않고 다음 로그를 읽는다. 만약 해당 규칙이 없었다면 새로운 규칙으로 추가한다.

2.4 파일 접근 제어 서브시스템

파일 접근 제어 서브시스템은 로그 수집 서브시스템과 유사하게 파일 접근 차단 여부를 정하는 DLL 파일, 프로세스 모니터, 파일 접근 차단 로그들을 주기적으로 전송하는 자동화 스크립트가 포함되며, 추가로 파일 접근을 차단한 경우 사용자에게 표시하는 팝업 UI까지 총 4가지 컴포넌트들로 구성된다. 파일 접근 허용 또는 차단은 프로세스 모니터에 의해 삽입된 DLL에서 결정되며, 흐름은 그림 4와 같이 이루어진다. OS에서 프로세스가 실행되면 프로세스 모니터가 이를 감지하여 DLL을 삽입한다. 프로세스가 파일 접근 API를 호출하면 DLL에서 미리 정의한 후킹 함수를 대신 호출한다. 후킹 함수에서는 화이트리스트에서 해당 프로세스가 요청하는 접근을 허용할지 확인한다.

화이트리스트에서 허용하는 접근인 경우, 실제 API 함수를 호출하여 그 결과를 반환한다. 허용하는 접근이 아닌 경우, 파일 접근 차단 로그를 기록하며 사용자에게 파일 접근이 차단되었음을 알려주는 팝업 UI를 표시한다. 마지막으로 API 함수 호출 실패 결과를 반환한다.

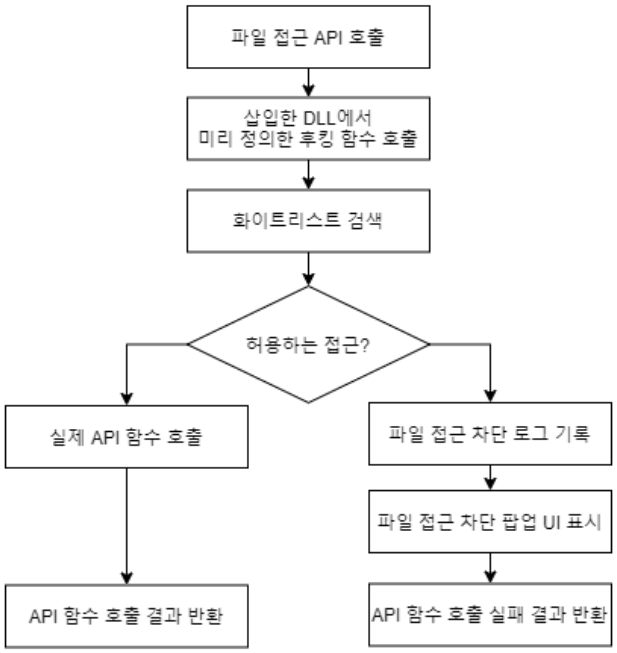

그림 4. 파일 접근 허용 및 차단 흐름

3. 시스템 구현 및 성능 분석

본 시스템은 Windows 10 환경에서 구현하였다. 프로세스 모니터에서 프로세스의 시작을 탐지하기 위해서 WMI(Windows Management Instrumentation)를 사용했고, 로그를 JSON 형태로 기록하는 데 오픈소스 JSON for Modern C++[8]를 사용하였다. 로그 파일들을 압축하는 데 tar를 사용하였고, 파일 접근 분류 서브시스템으로 로그를 전송하는 데 scp를 사용하였다. 화이트리스트는 sqlite3 데이터베이스로 구성하였다.

그림 5는 프로세스 모니터를 통해 실행한 프로세스에 DLL을 주입하는 모습이다. 그림 6은 로그 수집 서브시스템에서 실행된 프로세스로부터 수집한 로그들이며, 그림 7은 수집한 파일 접근 로그를 바탕으로 생성한 화이트리스트를 사용하여 음원 재생 애플리케이션인 foobar2000에서 flac 파일 음원 재생이 차단된 모습이다.

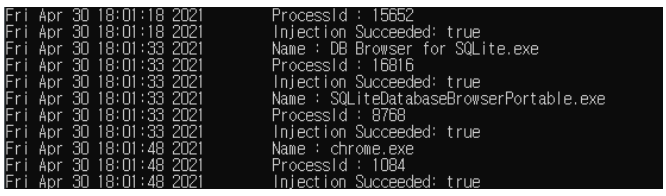

그림 5. 프로세스 모니터로 DLL을 주입하는 모습

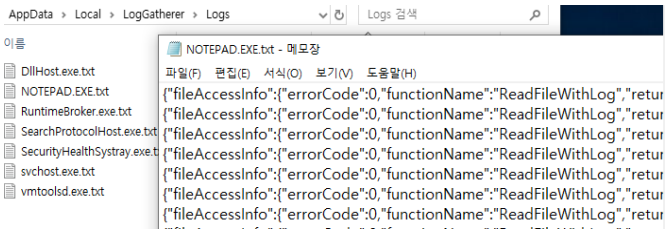

그림 6. 수집한 파일 접근 로그

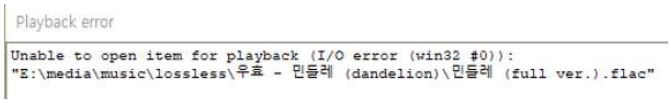

그림 7. foobar2000에서 flac 음원 재생 차단

4. 결론 및 향후 연구

본 논문에서는 파일 접근 로그를 수집하고 이를 바탕으로 파일 접근 화이트리스트를 만들어 파일 접근을 제어하는 시스템을 제안한다. 이를 통해 프로세스의 인가되지 않은 파일 접근을 제어함으로써 랜섬웨어 공격 및 데이터 유출을 방지하는 효과를 기대할 수 있다. 향후에는 보다 다양한 속성 정보들을 기반으로 파일 접근을 제어하여 차단 방식의 정확도와 안정성을 높일 필요가 있으며, 화이트리스트가 아닌 블랙리스트도 포함하는 정책으로 발전할 필요가 있다. 또한 관리자가 직접 판단하여 허용하는 부분 역시 자동화할 필요가 있다. 우리는 파일 접근 분류 서브시스템에 머신러닝 알고리즘을 적용하여 이러한 점들을 개선하는 연구를 진행할 계획이다.

참 고 문 헌

[1] 한국인터넷진흥원, “IBM 2020 글로벌 기업 데이터 유출 현황 주요 내용 분석,” 2020.

[2] Risk Based Security, “2020 Year End Data Breach QuickView,” 2021.

[3] 한국인터넷진흥원, “2020 글로벌 정보보호 산업시장 동향조사,” 2021.

[4] 윤정무, 조제경, 류재철, “파일 I/O Interval을 이용한 랜섬웨어 공격 차단 방법론,” 정보보호학회논문지, 26(3), 645- 653, 2016.

[5] 김재홍, 나중찬, "파일의 읽기/쓰기 권한 제한을 통한 암호화 랜섬웨어로부터 선택적 파일보호 연구," 한국정보처리학회 학술대회논문집, vol. 24, 234- 237, 2017.

[6] “Detours,” https://github.com/microsoft/Detours.

[7] V. C. Hu, D. Ferraiolo, R. Kuhn, A. R. Friedman, A. J. Lang, M. M. Cogdell, A. Schnitzer, K. Sandlin, R. Miller, K. Scarfone, et al., “Guide to Attribute Based Access Control (ABAC) Definition and Considerations,” NIST, vol. 800, no. 162, 2013.

[8] “JSON for Modern C++,” https://json.nlohmann.me.