개방형 OS에서 USB 개인정보 유출 방지 시스템*

현상엽O* 유준모 김규민 안종석
동국대학교 컴퓨터공학과
reserved-one@naver.com, whitesky118@gmail.com, zlarbals@gmail.com, jahn@dongguk.edu

A Prevention System of Personal Information Leakage Through USB for Open Source OS

SangYeop HyunO*, JunMo Yu, GyuMin Kim and JongSuk Ahn
Department of Computer Science and Engineering, Dongguk University

요 약

MS Windows 7 기술 지원 종료, 라이선스 비용 등의 이유로 MS Windows를 개방형 OS로 교체하려는 요구가 커지고 있다. 그러나 개방형 OS에서 기밀 정보 유출을 방지하는 적절한 DLP(Data Loss Prevention) 시스템을 지원하지 못한다는 점이 개방형 OS의 확산에 걸림돌이 되고 있다. 본 논문은 개방형 OS인 리눅스 환경에서 DLP 기능 중 하나인 USB로의 개인정보 유출을 방지하는 방안을 제시한다. 제안하는 방안은 USB 장치 인식과 파일 이동/복사를 감지 및 통제하는 USB 통제 모듈과 이동/복사하려는 파일을 검사하고 개인정보 통제 여부를 결정하는 개인정보 검출 모듈로 구성된다.

1. 서 론

우리나라 대부분의 공공기관과 기업에서 사용하는 업무용 PC는 마이크로소프트 (MS)의 Windows OS를 사용한다. 앞으로 Windows 7 기술 지원 종료, 라이선스 비용 등의 이유로 정부에서는 2026년까지 공공기관 PC 대부분에 개방형 OS를 사용한다는 계획이다[1]. 하지만 현재 개방형 OS를 위한 DLP 기술이 미흡하여 이에 대한 연구의 필요성이 대두되고 있다.

이러한 문제를 해결하기 위해서 본 논문에서는 개방형 OS 환경에서 조직 내부에서 외부로의 불법적인 자료 유출을 방지하는 시스템을 제안한다. 해당 모델의 개념을 설명하기 위해 자료 유출의 빈도가 높은 USB를 대상으로 한 개인정보 유출 방지 시스템을 제시한다. 본 시스템은 USB 통제 모듈과 개인정보 검출 모듈로 구성되어 있으며 오픈소스를 활용하여 쉽게 만들어 발전시킬 수 있게 설계하였다. USB 통제 모듈은 inotify와 fanotify 오픈소스를 활용하여 파일 시스템의 변화를 모니터링함으로써 USB의 장착과 파일 이동/복사 이벤트를 감시한다[2]. 해당 이벤트가 발생하면 개인 정보 검출 모듈에서 정규표현식과 Hyperscan[3] 오픈소스를 활용하여 개인정보를 검출한다. 검출 결과가 통제 정책에 해당되면 USB에 파일이 옮겨지지 않아 개인정보 유출을 방지한다.

2. 관련 연구

이 절에서는 USB 장착과 파일 복사/이동 이벤트를 감지하는 파일 시스템 이벤트 모니터링, 다양한 파일 형식에서 텍스트를 추출하는 구문 검출 기술, 그리고 개인정보 검출에 사용하는 정규표현식을 소개한다.

2.1 파일 시스템 이벤트 모니터링

윈도우의 파일 시스템과 파일 시스템 필터 드라이버를 개발할 수 있는 내장 프레임워크가 존재한다. 이 내장 프레임워크를 사용하는 드라이버로 미니 필터 드라이버가 있다[4]. 이를 활용하면 필터 관리자가 제공하는 API를 사용해 복잡한 I/O를 관리하지 않고도 간단히 파일 시스템 이벤트를 모니터링할 수 있다. 이와 비슷하게 리눅스(버전 3.8 이후)에는 파일 시스템 이벤트 모니터링 기능을 가진 오픈소스 라이브러리로 inotify와 fanotify가 있다.

inotify는 파일 시스템 이벤트를 모니터링하는 메커니즘으로 이를 이용하면 특정 디렉터리를 개별적으로 모니터링하여 해당 디렉터리에서의 파일 및 디렉터리의 생성/수정/삭제 이벤트를 감지할 수 있다[2].

fanotify도 inotify와 마찬가지로 파일 시스템 이벤트를 모니터링하는 메커니즘이지만 inotify와 다르게 감시하고자 하는 디렉터리의 하위 디렉터리에서의 이벤트 발생을 감지할 수 있다. 하지만 감시 디렉터리에서의 파일 및 디렉터리의 생성, 이동 및 삭제 이벤트를 감지할 수 없으며 대신 파일 읽기, 파일 열기, 파일 닫기 등의 이벤트를 감지할 수 있다[2].

2.2 구문 검출

구문 검출 기술은 파일에서 텍스트 혹은 메타데이터를 감지하고 추출하여 구문 분석을 한다. Apache TIKA[5] 오픈소스의 경우 파일에서 바이트의 특수 패턴을 찾음으로써 파일 형식을 탐지한다. 탐지한 파일 형식에 따라 다양한 파서를 사용하여 텍스트를 분석한다. 본 논문에서는 다양한 파일 형식에서 텍스트를 정확히 검출할 수 있어야 하므로 Apache TIKA[5] 오픈소스를 사용한다.

* "본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학지원사업의 연구결과로 수행되었음"(2016-0-00017)

2.3 정규표현식

정규표현식(Regular Expression)은 특정한 규칙을 가진 문자열의 집합을 표현하는데 사용하는 형식 언어이다. 이를 이용하여 개인정보를 추출하는 연구가 존재한다[6]. 본 논문에서는 유효성을 검증하기 이전에 정규표현식으로 개인정보를 분류한다. 기존 연구[6]와의 차이점은 고성능의 다중 정규표현식 매칭 기법을 사용한다는 점이다. 이를 위해 하이브리드 오토마타 기법을 사용하여 데이터 스트림에서 수만 개의 정규표현식들을 동시에 매칭시킬 수 있는 Hyperscan[3] 오픈소스를 사용한다.

3. 개인정보 유출 차단 시스템 구조 및 설계

3.1 개인정보 유출차단 시스템의 구조

본 연구에서 제안하는 개인정보 유출 차단 시스템은 그림 1과 같이 USB를 통제하는 USB 통제 모듈과 파일을 검사하는 개인정보 검출 모듈로 구성된다. USB 통제 모듈은 USB의 장착을 인식하고 장착된 USB로의 파일 복사/이동 이벤트를 감시하는 기능을 한다. 개인정보 검출 모듈은 복사/이동된 파일에 대하여 검사를 진행하고 통제 여부를 결정하는 기능을 한다.

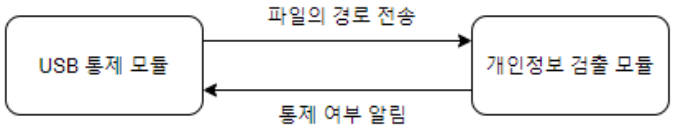

그림 1. 시스템 구조

시스템 프로세스로 돌아가는 USB 통제 모듈 안에서 시간이 오래 걸릴 수 있는 검사를 수행하는 것은 응답 속도 지연, 특정 모듈에서 오류가 났을 때 복구 등과 같은 시스템의 불안정성이 증가될 수 있다는 점과 추후 유지 보수 및 확장성을 고려하여 USB 통제 모듈과 개인정보 검출 모듈로 분리하였다.

3.2 USB 통제 모듈

USB 통제 모듈은 클라이언트의 USB 장착을 인식하는 기능과 장착된 USB로의 파일 이동/복사를 감지 및 통제하는 기능을 수행하며 그림 2와 같은 흐름을 갖는다.

우선 USB가 장착되면 이를 감지한다. USB 장착은 '/media/사용자명'에서의 디렉터리 생성 이벤트 감지로 인식한다. 이는 inotify API[2]를 활용하여 구현한다. 그리고 인식된 USB 디렉터리는 해당 경로로 파일 이동/복사가 이루어지는지 감시되어야 한다. inotify는 인식한 USB 디렉터리를 기점으로 하위 디렉터리에서의 파일 시스템 변경 이벤트를 감지할 수 없기 때문에 USB 디렉터리에 대한 감시는 fanotify API[2]를 활용하여 구현한다. fanotify가 감시하는 경로로 파일 복사/이동이 될 때 발생하는 file close event를 감시하여 파일 복사/이동을 감지한다.

위 과정을 통해 감지된 파일은 로그 데이터로서 활용될 수 있게 로그 디렉터리로 이동된다. 해당 파일의 경로를 개인정보 검출 모듈로 전송하고 통제 여부 값을 전달받는다. 안전하지 않은 파일이라면 USB로 이동시키지 않고 로그 디렉터리에 남김으로써 파일 복사/이동을 통제한다. 안전한 파일이었다면 다시 USB 디렉터리로 옮겨온다.

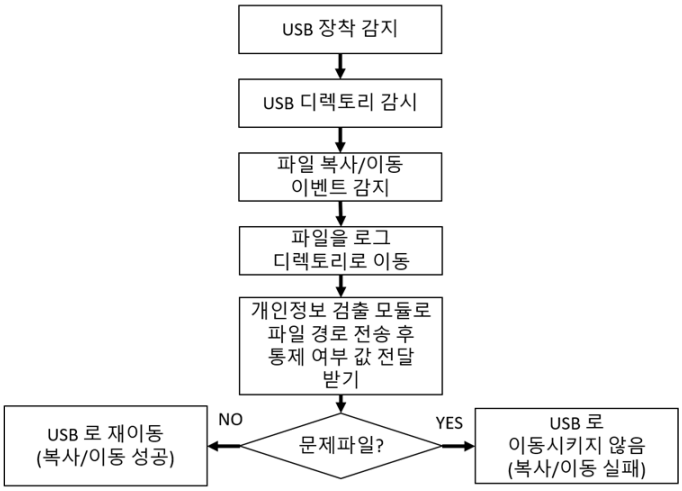

그림 2. USB 통제 모듈 수행 절차

3.3 개인정보 검출 모듈

USB 통제 모듈에서 감지한 파일을 대상으로 개인정보 검사를 실시한다. 개인정보 검출 모듈은 그림 3과 같은 흐름을 갖는다.

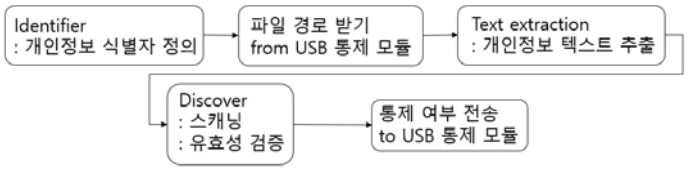

그림 3. 개인정보 검출 모듈 수행 절차

우선 개인정보별로 식별자(Identifier)를 정의한다. 각 식별자들은 해당하는 개인정보의 이름, 정규표현식, 유효성 검증 알고리즘을 갖는다. USB 통제 모듈로부터 파일 경로를 받으면 Text extraction에서 해당 파일 내용을 추출한다. Apache TIKA[5] 오픈소스를 사용하여 다양한 형식의 파일을 인식할 수 있게 한다.

Discover에서는 추출한 파일 내용을 가지고 스캐닝과 유효성 검증 작업으로 나누어 개인정보를 검출한다. 스캐닝에서는 각 개인정보 정규표현식을 파일 내용에 매칭 시켜 각 패턴에 맞는 데이터를 추출한다. 이 때 개인정보나 기밀정보의 확장, 즉 정규표현식의 개수가 많아질 경우를 고려하여 Hyperscan[3] 오픈소스를 사용한다. 각 개인정보 패턴에 맞는 데이터를 추출하면 유효성 검증 작업을 실시한다. 존재할 수 없는 값이 있거나 체크섬 오류가 발생하면 해당 개인정보가 아닌 것이다.

이러한 작업이 완료되면 개인정보 별로 몇 건이 검출되었는지 알 수 있다. 이후 검출 결과를 통제 정책에 대조하여 파일 이동/복사 통제 여부를 USB 통제 모듈에 알린다.

4. 시스템 동작 및 실험

그림 4. 통제/검출 결과 작동 화면

그림 4는 제안된 시스템이 동작하는 과정을 보여준다. 본 프로그램은 엔드 포인트 유저 측의 컴퓨터가 부팅이 되면서 실행된다. USB 통제 모듈은 '/media/사용자명' 디렉터리들을 감시하고 있으며 사용자명 디렉터리 밑에 USB가 장착되면 이를 감지하고 마크한다. 마크는 감시가 시작됨을 의미한다. 그림 4에서 '/media/reserved1'이 감시되고 FROZEN USB 장착을 감지하고 마크한다. 아래 그림 5는 이 과정을 코드로 보여준다. i_fd와 f_fd는 각각 inotify와 fanotify의 파일 디스크립터이다. i_fd로부터 이벤트 데이터를 (void*)event에 읽어 들인다. 만약 (void*)event가 USB 디렉터리 생성 이벤트라면 f_fd에 usb_path를 마크한다.

```
void handle_inotify_event(int i_fd, int f_fd) {
    ... // 변수 선언 등 수행
    length = read(i_fd, (void *)event, EVENT_BUF_SIZE);
    while (i < length) {
        if (event[i]->len) {
            if (event[i]->mask & IN_ISDIR) {
             sprintf(usb_path, "/media/%s/%s",userID, event[i]->name);
             fanotify_mark(f_fd, FAN_MARK_ADD | FAN_MARK_MOUNT,
                                FAN_CLOSE_WRITE, AT_FDCWD, usb_path);
            }
            ... // 다른 event type 처리
```

그림 5. 파일 복사/이동 감지 코드

USB로의 파일 이동/복사를 감지하면 로그 디렉터리로 파일이 이동되고 개인정보 검출 모듈이 검사를 진행한다. 그림 4에서 USB 통제 모듈은 개인정보검출1.pdf가 USB로 이동하는 동작을 감지하고 이 파일을 로그 디렉터리로 이동시킨다.

개인정보 검출 모듈은 해당 파일의 내용을 대상으로 개인정보를 검출한다. 그림 4에서 개인정보검출1.pdf의 경우 주민번호 11건, 휴대폰 번호 16건, 전화번호 19건, 건강보험증 번호 10건이 검출되었다. 검출된 개인정보 중 주민번호의 검출 과정을 살펴보면 우선 다음과 같은 정규표현식을 사용하여 대상 문서를 스캐닝 한다: [0-9][0-9][0-1][0-9][0-3][0-9][-₩₩x20]?[1234]₩₩d{6} 월은 01부터 12까지 존재하므로 3번째 자리는 [0-1]을 사용하였고 일은 01부터 31까지 존재하므로 5번째 자리는 [0-3]을 사용하였다. 주민번호 뒷자리 중 1번째 자리는 00년생 이전이면 1,2, 00년생 이후이면 3,4를 사용하므로 [1234]를 사용했다. 그리고 검출된 문자열에 대해 다음과 같이 유효성을 검증한다:

1. 추출된 데이터로 생년월일이 과거에 실재했던 날짜인지 확인한다.
2. 7번째 자리가 2000년생 이후면 3 또는 4인지 확인한다.
3. 체크섬이 맞는지 유효성을 검증한다.

유효성이 입증되면 주민번호가 검출된다. 이러한 과정을 각 개인정보에 대하여 파일 내용 전체에 반복하면 그림 4와 같이 검출된 개인정보의 종류와 개수가 나온다.

개인정보 검출 모듈은 검출 결과와 통제 정책을 비교하여 파일 이동/복사 통제 여부를 USB 통제 모듈에 알린다. USB 통제 모듈은 통제 여부에 따라 USB로의 파일 이동/복사를 통제하거나 허용한다. 그림 4에서 개인정보검출1.pdf는 통제되었다. USB를 제거하면 탈착이 감지된다. 그림 4 마지막에 FROZEN USB가 탈착된다.

5. 결론 및 향후 연구 과제

정부에서는 2026년까지 공공기관 PC 대부분을 개방형 OS로 대체할 계획이다. 본 논문에서는 개방형 OS로 전환 시 공공기관이나 기업에서 필요한 DLP, 특히 엔드 포인트 유저단에서 USB로의 특정 개인정보 유출 방지 시스템을 제시한다.

향후에는 USB 이외의 매체와 네트워크를 포함한 여러 유출 경로에 대응하고 다양한 개인정보와 기밀정보로 확장하는 연구를 진행할 계획이다.

REFERENCES

[1] 행정안전부 보도자료(2020.02.05.), 「공공기관 PC, MS 윈도우 대신 개방형OS 사용한다.」

[2] 전우진, 박기웅, PaaS 클라우드 컴퓨팅을 위한 컨테이너 친화적인 파일 시스템 이벤트 탐지 시스템, 한국차세대컴퓨팅학회 논문지, vol.15, no.1, pp.88-89, 2019.

[3] Xiang Wang, Yang Hong, Harry Chang, KyoungSoo Park, Geoff Langdale, Jiayu Hu, and Heqing Zhu, Hyperscan: A Fast Multi-pattern Regex Matcher for Modern CPUs. NSDI'19: Proceedings of the 16th USENIX Conference on Networked Systems Design and Implementation, pp.631-648, 2019.

[4] Mohammed I. Al-Saleh, Hanan M. Hamdan, Precise Performance Characterization of Antivirus on the File System Operations. Journal of Universal Computer Science, vol. 25, no. 9, p.1093, 2019.

[5] Apache TIKA, [Online], Available http://tika.apache.org

[6] 이민석, 김숙현, 윤지애, 원유재, 이미지파일에 포함된 개인정보추출에 관한 연구, 한국컴퓨터정보학회 학술발표논문집, 25(1), p. 210, 2017.