https://doi.org/10.7840/kics.2023.48.9.1114

마이크로 세그먼테이션 기반 파일 암호화 랜섬웨어 탐지에 관한 연구

서 정 우◆

A Study on Micro-Segmentation Based File-Encrypting Ransomware Detection

Jung-woo Seo◆

요 약

기존 경계 중심의 보안 대책은 복잡·다양화하는 IT 인프라 환경에서 한계점을 노출하고 있으며, 경계 보안을 넘어 신뢰 영역 내부로 침입한 공격자를 효과적으로 탐지하지 못하고 있다. 공격 범위를 줄이기 위한 선제적 조치를 하더라도 정보 유출을 막는 것이 어려워 측면 이동을 탐지하고 방지하는 능력을 강화할 필요성이 있다. 마이크로 세그먼테이션은 IT 환경을 통제할 수 있는 구역으로 나누어 각 워크로드를 상호 안전하게 격리하는 동시에 보안 유출을 신속하게 탐지하고 억제할 수 있는 장점을 제공한다. 제안된 방법론은 마이크로 세그먼테이션 기반으로 디렉터리와 파일을 영역으로 분리한 후 블록으로 구성하고, 파일 암호화 랜섬웨어에 감염되어 파일 암호화가 발생하는 이벤트를 실시간으로 탐지한다. 파일 암호화 랜섬웨어 감염 여부를 판단하기 위하여 영역 블록의 경계를 넘어 파일 암호화가 발생하거나 파일 확장자가 변경되면 우선순위나 가중치를 계산하여 랜섬웨어 감염을 판단한다. 실험은 가상화 환경에서 실제 파일 암호형 랜섬웨어가 발현하도록 하였으며, 테스트 환경에서 파일 암호형 랜섬웨어 감염을 실시간으로 탐지하였다.

Key Words : Ransomware, Micro-segmentation, Anomaly detection, Malware

ABSTRACT

Traditional perimeter-focused security measures have limitations in a complex and diversifying IT infrastructure environment. Traditional perimeter firewalls cannot effectively detect attackers who have crossed the perimeter and entered the trust zone. While organizations take proactive steps to reduce the scope of attacks, it is difficult to stop data leaks. Therefore, organizations need to enhance their ability to detect and prevent lateral movement The proposed methodology is based on micro-segmentation to separate directories and files into regions, organize directories and files into blocks, and then detect anomalies. The experiment utilized real data from a virtualized environment and measured the time to detect file-encrypted ransomware by executing the malware.

◆ First Author : ICT Polytech Institude of Korea Department of Information Security, jwseo@ict.ac.kr, 정회원
논문번호 : 202304-078-B-RN, Received April 11, 2023; Revised May 24, 2023; Accepted June 20, 2023

Ⅰ. 서 론

최근 감소세에도 불구하고 랜섬웨어는 여전히 심각한 보안 위협 요소이며, 파일 암호형 랜섬웨어는 피해자의 파일을 암호화하는 악성코드의 일종이다. 공격자는 데이터에 대한 액세스를 복구하려는 피해자에게 몸값을 요구하며, 암호 해독 키를 얻기 위한 지침서에는 비트코인으로 몸값을 지불하는 방법을 사용자에게 표출한다[16]. 공격자는 제조, 운송, 통신, 금융 및 의료 서비스 등 경제적 이익이 높은 기업을 대상으로 랜섬웨어 공격을 수행한다[1,2]. 파일 암호형 랜섬웨어는 AES-256의 대칭키 알고리즘으로 파일을 빠르게 암호화하고, RSA-1024 또는 RSA-2048과 같은 비대칭 알고리즘으로 암호화키를 암호화하여 공격자의 복호화 키가 없이는 파일 복구를 어렵게 한다.

파일 암호형 랜섬웨어의 피해가 큰 이유는 네트워크 기반 공유 기능의 활성화로 기업에서 유통되는 중요 데이터들이 집중화되어 보관되기 때문이다. 경제적인 이유나 물리적인 한계로 별도의 백업 시스템 구축이 어려운 경우 여유 디스크 공간을 확보한 컴퓨터의 공유 폴더를 이용하여 중요 파일 및 백업 데이터를 보관하는 경우가 대표적인 사례이며, 공유 네트워크 환경의 컴퓨터가 랜섬웨어에 감염되면 모든 파일이 암호화되어 피해를 증폭시킬 수 있는 위험한 환경을 만들 수 있다. 실제로 랜섬웨어 피해자의 65%가 공유 네트워크 환경에서 피해가 발생하였다[3].

다양한 변종의 랜섬웨어가 확산되면서 학계와 보안 회사는 다양한 랜섬웨어 방어 도구들을 개발하였는데, 디스크 액세스 활동 모니터링을 통한 랜섬웨어 탐지 도구는 효과적인 랜섬웨어 탐지 도구 중 하나이다. 하지만, 정상적인 디스크 액세스 활동을 악의적인 행위로 판단할 수 있으며, 시스템의 성능에 따라 일관된 정책 기준을 적용하는 것이 어렵다[4,5].

본 연구에서는 통제할 수 있는 구역으로 나누어 각 워크로드를 상호 안전하게 격리하는 동시에 서비스 영역을 더욱 세분화하고 횡으로 이동하는 사이버 공격을 효율적으로 대처할 수 있는 랜섬웨어 탐지방안을 제안한다. 제안된 알고리즘은 디렉터리와 파일을 목록화한 후 디렉터리와 파일을 마이크로 세그먼테이션의 각 영역 속성으로 정의하며, 파일 암호화 랜섬웨어가 횡으로 이동하면서 디렉터리의 파일을 암호화하는 것을 탐지하기 위해 디렉터리와 파일을 마이크로 세그먼테이션의 각 영역으로 블록화한다. 실시간 모니터링을 통해 각 영역 블록의 파일이 암호화되면 이상 행위로 탐지하여 이벤트값을 증가시키고, 이벤트값이 기준값을 넘어서면 파일 암호형 랜섬웨어로 탐지한다.

본 논문의 나머지 세션은 2장에서 파일 암호형 랜섬웨어와 마이크로 세그먼테이션에 관한 관련 연구를 소개하고, 3장에서는 파일 암호형 랜섬웨어 탐지를 위한 시나리오를 설명한다. 4장에서는 파일 암호형 랜섬웨어 탐지를 위한 알고리즘을 설명하고, 5장에서는 제안된 방법론을 프로토타입으로 구현하고 실험 결과를 분석하여 성능을 평가한다. 마지막으로 6장에서는 결론에 관해 서술한다.

Ⅱ. 관련 연구

2.1 최근 랜섬웨어 현황

최근 랜섬웨어 현황을 살펴보면, 체이널리시스 보고서에 따르면 2022년 랜섬웨어 피해액은 4억 5,680만 달러로 2021년 랜섬웨어 피해액 7억6,500만 달러와 비교해 감소하였으나 가상자산을 제외한 피해액으로 실제 피해 금액은 더 높을 것으로 전망했다. 기업의 침해사고 경험에서도 랜섬웨어와 악성코드 감염은 높은 수준을 유지하며, 사회 인프라 운영기관 등을 대상으로 랜섬웨어 공격을 통한 직접적인 피해를 유발하고 있다. 랜섬웨어는 시스템 및 SW 취약점 등을 이용한 보안 취약점과 이용자를 속이는 사회공학적 공격기법 등 다양한 경로를 통해 전파 및 감염되고 있으며, 일부 공개된 도구를 활용해 복구를 시도하지만 대부분 복구가 불가능하다. 파일 암호화 랜섬웨어에 감염되면 파일을 암호화해 사용자의 시스템 이용을 제한하며, 공격자는 피해자에게 복구를 위한 돈을 요구하거나 암호화한 파일의 삭제 또는 다크웹 등을 통해 정보를 무단 유통, 판매하여 2차 피해를 유발한다[6]. 최근 러스트(Rust) 기반으로 작성된 네바다(Nevada) 랜섬웨어가 유포되고 있으며, 감염된 파일은 '.NEVADA' 확장자가 추가되어 암호화된다. 네바다 랜섬웨어의 특징은 숨겨진 파티션을 로드하여 암호화하거나 네트워크 공유 리소스에 접근해 암호화를 수행한다[7].

2.2 마이크로 세그먼테이션

마이크로 세그먼테이션은 네트워크 보안 영역 경계를 구성해 각 워크로드를 독립적으로 분리하여

보호함으로써 네트워크 보안을 더욱 세분화하는 방법이다. 소프트웨어 정의 네트워킹(SDN, Software defined networking)과 네트워크 기능 가상화(NFV, Network Function Virtualization)의 등장으로 마이크로 세그먼테이션을 소프트웨어에서 실현함으로써 배포와 관리가 쉬워졌다[14].

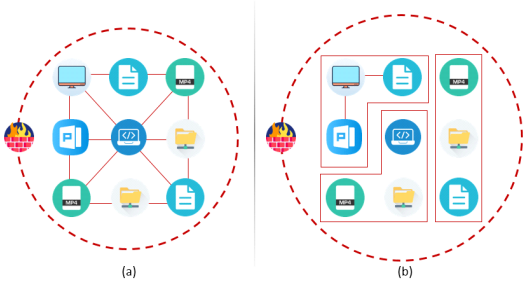

그림 1. (a) 일반 파일구조, (b) 마이크로 세그먼테이션
Fig. 1. (a) General file structure, (b) Microsegmentation

마이크로 세그먼테이션의 구현은 분산 방화벽에 보안 정책 관리와 엔드포인트 방어와 같은 기능을 추가한 확장이라고 볼 수 있다. 시스코의 블로그[8]에 따르면 마이크로 세그먼테이션은 엔드포인트 단위로 정책 설정을 의미하며, 기존 방화벽이 수직으로 데이터 플로우의 이동을 통제하지만, 마이크로 세그먼테이션은 내부 네트워크에서 수평으로 이동을 통제하여 모든 데이터 플로우의 통신 채널을 차단하고 최소 권한 정책을 적용해 필요한 통신만을 허용한다.

2.3 기존 연구의 한계

랜섬웨어 탐지 도구는 전통적으로 프로그램 바이너리를 실행하기 전에 정적 분석을 기반으로 하는 안티바이러스 도구와 유사한 기술을 사용한다[9,10]. 하지만, 새로운 유형의 바이너리가 발견되거나 난독화 기술이 사용되면 파일 암호형 랜섬웨어를 탐지하지 못하거나 오탐이 발생할 위험이 크고, 변종이 생성하는 수많은 시그니처에 대응하기 어려운 단점이 있다[12]. 정적 예방 기술은 사용자 호스트에서 실행 중인 모든 프로그램이 수행하는 작업을 모니터링하여 지표를 추출하는 아키텍처로 대체되고 있으며, 파일 암호형 랜섬웨어에 감염되어 파일을 암호화하고 있을 때 이상 행위를 조기 탐지하여 피해를 최소화하고 있다. 파일 암호형 랜섬웨어를 대처하는 최고의 방법은 모든 데이터를 백업하여 암호화된 파일을 복구하는 것이지만, 백업 대상 시스템의 규모가 커지거나 데이터 용량이 크면 자원 확보와 비용 증가에 따른 대책 마련이 필요하다.

Ⅲ. 시나리오

IT 인프라가 복잡해지면서 네트워크 환경에서 사이버 침해 위험성은 더욱 증가하고 있으며, 공격이 성공하면 공격자는 업무망과 같은 신뢰 영역을 방해받지 않고 이동할 수 있는 위험이 존재한다. 네트워크 영역을 마이크로 세그먼테이션으로 세분화하여 구현한다면 공격자의 공격 범위를 분할된 영역으로 제한할 수 있으며, 악성코드의 확산 및 횡으로 이동을 억제할 수 있다.

파일 암호형 랜섬웨어에 감염되면 대칭키(AES)와 비대칭키(RSA) 암호화 방식으로 시스템에 저장된 각종 문서와 파일이 암호화되어 피해자가 암호화 파일을 복구할 수 없도록 한다. 제안된 방법론은 파일 암호형 랜섬웨어에 감염되어 피해가 확산하는 것을 조기에 탐지하기 위해 디렉터리와 파일을 마이크로 세그먼테이션 영역으로 블록화하고 이상 행위가 탐지되면 이벤트를 발생시켜 기준값을 넘어가면 랜섬웨어로 탐지한다.

3.1 마이크로 세그먼테이션 영역 블록 설정

마이크로 세그먼테이션은 엔드포인트를 각 영역으로 분할하여 고립된 섬처럼 취급하며, 자원들은 통신이 가능한 알려진 장치로만 통신할 수 있으므로 각 영역 블록에 문제가 발생하면 주변으로 위험이 자유롭게 확산하는 것을 탐지할 수 있다. 마이크로 세그먼테이션 영역 블록의 설정은 시스템에 포함된 파일들을 각 영역으로 분류하고, 각 영역은 독립적인 블록으로 구성하여 파일 암호형 랜섬웨어의 발현을 탐지하는 역할을 한다. 마이크로 세그먼테이션의 영역을 분류하는 것은 파일들의 속성 정보를 이용하며, 파일명 및 파일 확장자, 수정일 등의 세부적인 속성 정보를 기록한다.

3.2 마이크로 세그먼테이션 기반 이상 행위 모니터링

디렉터리와 파일을 마이크로 세그먼테이션의 각 영역 블록으로 분할하는 것은 횡으로 이동하는 보안 위협을 효과적으로 대처할 수 있어 파일 암호형 랜섬웨어를 신속하게 탐지할 수 있다. 파일 암호형 랜섬웨어에 감염된 파일의 특징은 일반 파일을 대칭키 방식으로 암호화하고 비대칭키로 암호화

키를 암호화한 후 암호화 대상 파일의 확장자를 변경한다. 이상 행위 탐지는 마이크로 세그먼테이션이 적용된 디렉터리와 파일의 속성 정보를 실시간으로 모니터링하고 암호화 같은 이상 행위가 탐지되면, 변경사항에 대한 속성 정보(변경일시, 확장자명 등)를 저장한다.

3.3 파일 암호형 랜섬웨어 탐지

마이크로 세그먼테이션 기반으로 영역별 분류된 파일들이 파일 암호형 랜섬웨어에 감염되어 파일 암호화 및 확장자가 변경되면, 실시간 모니터링으로 이벤트를 탐지한다. 마이크로 세그먼테이션 각 영역 블록에서 이상 행위가 탐지되면 이벤트값을 증가시키고, 실시간으로 기준값과 비교한 후 파일 암호형 랜섬웨어 감염 여부를 판단한다.

Ⅳ. 파일 암호형 랜섬웨어 탐지 알고리즘

마이크로 세그먼테이션 기반 각 영역 블록을 분할하는 방법과 파일 암호형 랜섬웨어 탐지를 위한 알고리즘 및 구축 방안을 설명한다.

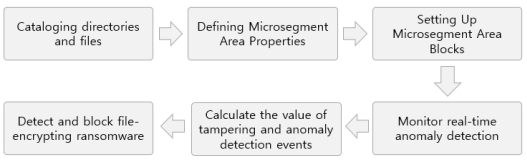

그림 2. 파일 암호형 랜섬웨어 탐지 절차도
Fig. 2. Flowchart for file encrypting ransomware detection

4.1 마이크로 세그먼테이션 영역 블록 설정

마이크로 세그먼테이션 정책은 디렉터리 및 파일을 통제할 수 있는 구역으로 나누어 랜섬웨어 공격을 신속하게 탐지함으로써 사용자 파일의 일부 또는 전부가 암호화되어 가용성이 제한되는 것을 방지한다.

마이크로 세그먼테이션 영역 및 블록 설정 절차를 살펴보면 첫째, 시스템의 디렉터리 목록을 추출하고 계층화하여 목록을 관리한다. 디렉터리에 파일이 존재하지 않거나 디렉터리만 존재하는 경우는 디렉터리 관리 목록에서 제외하고, 디렉터리 목록을 중요도와 최근 수정일에 따라 우선순위와 가중치를 부여한다. 예를 들어, 운영체제의 Windows 폴더는 운영체제 동작에 필요한 주요 시스템 파일들을 담고 있는 중요한 폴더이지만, 파일 암호형 랜섬웨어에 감염된 시스템은 Windows와 Program Files와 같은 시스템 폴더는 암호화 대상에서 제외하기 때문에 시스템 관련 폴더는 중요도와 가중치를 제외하거나 낮게 설정한다. 파일 유형이 다양하거나 중요 파일을 포함하는 디렉터리를 선택하면 이상 행위를 탐지하는 시간을 단축할 수 있다. 또한, 파일 수정일이 과거일수록 변경이 발생할 확률이 낮으므로 변경이 발생하는 경우 가중치를 높게 부여한다. 둘째, 시스템의 파일 목록을 추출하고, 파일을 디렉터리 목록과 연계한 후 계층화하여 구성한다. 파일은 디렉터리의 우선순위와 가중치를 상속하며, 파일의 수정일이 현재 일과 차이가 크고 과거일수록 가중치를 높게 설정한다. 예를 들어, 파일이 속한 디렉터리의 우선순위와 가중치가 <우선순위, 가중치>이면, <1, 5>와 같이 나타내고 디렉터리의 우선순위와 가중치를 파일에서도 같이 상속받는다. 디렉터리의 속성 정보에는 현재일에서 파일의 수정일을 뺀 값을 추가하여 <1, 5, 126>와 같이 생성한다.

표 1. 마이크로 세그먼테이션 영역 속성
Table 1. Micro-segmentation domain attribute

Item	Description
	Value
isDirectory	Directory or Files
	y:directory, n:file
Micro-segment ID(msid)	Micro-segment ID (Root directory is ‘ms1xxxx’, ‘ms2xxxx’, etc.)
	ex) ms100009
Parent MS ID (pid)	Parent micro-segment ID
	ex) ms100001
Priority	Directory priority
	ex) 1, 2, 3, …
Weight	Directory
	ex) 0.5, 03., 01, …
Create date	Directory or File create
	ex) 2020.6.15.
Modification date	Directory or File modification
	ex) 2021.3.4.

셋째, 마이크로 세그먼테이션 적용을 위하여 시스템의 모든 디렉터리 및 파일들을 마이크로 세그먼테이션의 각 영역으로 나누어 속성 정보를 생성한다. 마이크로 세그먼테이션 영역 속성은 [표 1]과 같이 나타내며, 새로운 디렉터리 및 파일이 생성되면 마이크로 세그먼테이션 영역 속성을 추가로 생성한다. 마이크로 세그먼테이션 영역 속성을 생성하는 경우는 Windows와 Programs Files 같은 시스템 디렉터리와 실행파일(.exe, .dll 등)은

제외하며, 각 드라이브(ex. C:\, D:\ 등)의 하위 디렉터리는 루트(Root) 디렉터리로 표시한다. 마이크로 세그먼테이션 영역 속성은 [그림 3]과 같은 연결 구조로 구성한다.

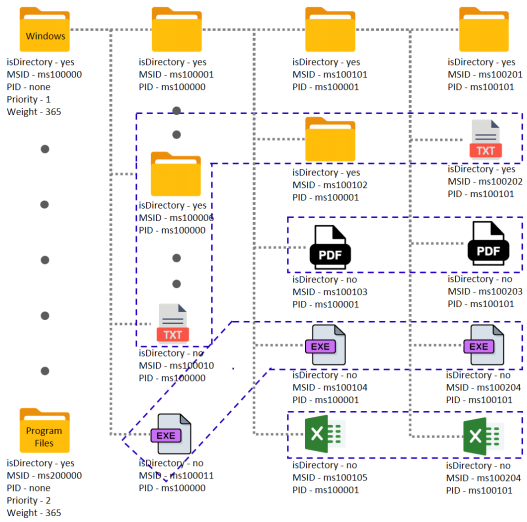

그림 3. 마이크로 세그먼테이션 영역 블록
Fig. 3. Micro-segmentation domain block

[표 2]는 디렉터리와 파일을 통제할 수 있는 영역으로 나누기 위한 마이크로 세그먼테이션 영역 설정 알고리즘을 나타낸다.

표 2. 마이크로 세그먼테이션 영역 설정 알고리즘
Table 2. Micro-segmentation domain algorithm

```
procedure Micro-segment Domain Attribute:
Folder_List=System
folder_Cnt=Folder_List.count
p=1, k=1
for i=1 to folder_Cnt
    mda.msid="ms_" + k + "_" + p
    mda.pid=null
    if Folder_List.name=="folder"
        isDirectory=true
        if Folder_List.name=="system folder"
            mda.priority=value
    else if Folder_List.name=="system file"
        mda.weight=value
        mda.cdate=date
        mda.mdate=date
Folder_List.nextItem
mda.pre_pid=mda.msid
Call MDA (Folder_List)
```

```
Call MDB (Folder_List)
Function MDA (Folder_List)
    for j=1 to Folder_List.count
        if Folder_List.type==Folder
            ArrayFolder_LIST=Folder_List.name
            isDirectory=true
        mda.msid="ms_" + k + "_" + p
        mda.pid=mda.pre_pid
        mda.pre_pid=mda.msid
        mda.cdate=date
        mda.mdate=date
        p += 1
        Folder_List.nextItem
if k != folder_Cnt
    k += 1
    Call MDA (ArrayFolder_List)
    Call MDB (ArrayFolder_List)
else
    return mda, mdb
```

넷째, 마이크로 세그먼테이션 영역 속성을 활용하여 마이크로 세그먼테이션을 구성하게 되며, 루트 디렉터리의 하위 디렉터리 및 파일을 영역 블록으로 묶어서 하나의 마이크로 세그먼테이션을 생성한다. [표 3]은 마이크로 세그먼테이션 영역 블록을 생성한 예제이며, 각 디렉터리에 존재하는 파일을 기준으로 확장자가 같으면 파일들을 마이크로 세그먼테이션을 위한 동일 영역으로 분류하여 블록을 구성한다.

표 3. 마이크로 세그먼테이션 영역 블록
Table 3. Micro-segment domain block

Item	Description
	Value
Micro-segment Block(mdb)	Micro-segment Block ID
	ex) mdb00009
Micro-segment ID(msid)	msid value of Micro-segment domain attribute
	ex) ms100009
File path	Encrypted file paths with changed file extensions
	ex) C:\Windows\system32
File name	Encrypted file name
	ex) file name
File extensions	Encrypted file extension
	ex) .cerber, .locky

[표 4]는 마이크로 세그먼테이션 영역 블록을 설정하기 위한 알고리즘을 나타낸다.

표 4. 마이크로 세그먼테이션 영역 블록 설정 및 이상 행위 탐지
Table 4. Micro-segmentation domain block and Anomaly detection

```
procedure Micro-segment Group Block:
Function MDB (Folder_List)
    for m=1 to Folder_List.count
        if Folder_List.type != Folder
            mdb.mdbid="mdb_" + k + "_" + q
            mdb.msid=Folder_List.msid
            mdb.filepath=Folder_List.filepath
            mdb.filename=Folder_List.filename
            mdb.filename_extension=Folder_
List.filename_extension
            if pre_mdb.filename_extension !=
mdb.filename_extension
                q += 1
            pre_mdb.filename_extension=
mdb.filename_extension

procedure Anomaly Behavior Detection Attribute:
    if event_value ≥ threshold
        return Ransomware_detection
```

4.2 실시간 이상 행위 탐지 모니터링

파일 암호화 랜섬웨어의 특징은 대칭키와 비대칭키 암호화 알고리즘(RSA 2048, AES 256, ECC 등)을 이용하여 피해자의 문서, 데이터베이스 등을 암호화하며, 복호화를 위한 키는 공격자가 소유하고 있다. 본 장에서는 파일 암호화 랜섬웨어에 의하여 파일이 암호화되는 것을 초기에 탐지하고 대응하기 위한 절차를 설명한다. 마이크로 세그먼테이션 영역 블록(Micro-segment ID, MDB)에 의하여 설정된 정보를 기반으로 실시간 영역 블록에 대한 파일 암호화를 검사하여 이상 행위가 탐지되면, [표 5]의 테이블 속성에 이벤트 정보를 생성한다.

표 5. 이상 행위 탐지 속성 테이블
Table 5. Anomaly behavior detection attribute table

Item	Description
	Value
Micro-segment ID(msid)	msid value of Micro-segmentation domain attribute
	ex) ms100009
File path	Encrypted file paths with changed file extensions
	ex) C:\Windows\system32
File name	Encrypted file name
	ex) file name
File extensions	Encrypted file extension
	ex) .cerber, .locky
Modification date	File modification date
	ex) 2021.3.4.
Modification time	File modification time
	ex) 14:31:20
Modification count	File modification count
	ex) 1,2,3, ..., 10

[그림 4]의 알고리즘과 같이 이상 행위 탐지를 위한 변경 행위 탐지(Changed Behavior Detection, *cbd*) 값은 마이크로 세그먼테이션 영역 속성의 우선순위와 가중치를 참조하여 파일에 변경이 탐지되면 *cbd*의 값을 증가시킨다. *cbd*는 마이크로 세그먼테이션 영역 블록에서 정의한 속성 정보에 변경이 발생하면 *cbd*의 값을 증가시키고, *cbd*의 값이 임계 설정값(Threshold)을 초과하면 이상 행위 탐지(Anomaly Behavior Detection, *abd*) 이벤트값을 증가시킨다. 변경 행위 탐지(*cbd*) 값은 우선순위 'P'와 가중치 'W'를 포함하여 누적값으로 저장하고, *abd*의 값은 다음 장에서 파일 암호형 랜섬웨어 여부를 탐지하기 위한 비교 값으로 사용한다.

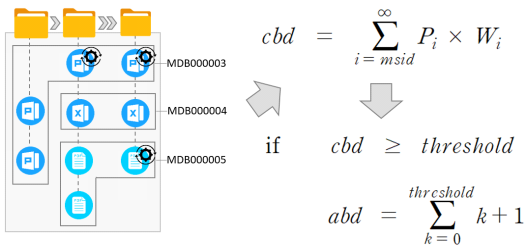

그림 4. 이상 행위 탐지
Fig. 4. Anomaly behavior detection

*abd*의 값은 초기에 '0'으로 세팅하며, 결과값이 10에 가까울수록 랜섬웨어 감염으로 판단한다. *abd*의 값은 시간 정보(Timestamps)를 활용하여 정해진 시간 동안 입력값이 없으면 *abd*의 값을 '0'으로 초기화한다.

4.3 파일 암호형 랜섬웨어 탐지 및 차단

*abd*의 값이 임계값(Threshold)을 넘어가면 파일

암호형 랜섬웨어로 의심할 수 있으며, 관리자에게 악성코드 감염 의심 알림을 전송하고 주요 파일들이 모두 암호화되는 것을 차단하기 위해 시스템을 강제 종료한다.

V. 실 험

실험 환경의 호스트 운영체제는 윈도우 10 Education 22H2 버전이며, 11th Gen Intel Core i7-11700 CPU 2.5GHz와 32GB의 메모리를 사용한다. 가상 컴퓨팅 환경의 구축을 위하여 VMware 16 버전을 사용하였으며, 디렉터리 및 파일 목록화, 마이크로 세그먼테이션의 속성 및 블록화 설정, 랜섬웨어 탐지를 위한 프로토타입 구현은 Python 언어(ver. 3.11)를 사용하였다.

디렉터리와 파일을 목록화하고, 마이크로 세그먼테이션의 생성을 위하여 각 디렉터리와 파일을 영역 속성으로 정의하고 블록화하여 구성한다. 실험을 위해 악성코드 분석 블로그[15]에서 제공하는 파일 암호형 랜섬웨어 악성코드를 활용하며, 실험 대상 컴퓨터의 파일들을 암호화한 후에 파일 확장자를 변경하고 암호 해독 키를 얻기 위한 지침서를 화면에 표출한다. 실험은 가상화 환경에서 진행하였으며, C:\ 드라이브의 시스템 폴더와 프로그램 폴더, 사용자 생성 폴더와 D:\ 드라이브의 사용자 폴더를 대상으로 파일 암호형 랜섬웨어를 실행하여 실험을 수행하였다.

실험에 사용한 디렉터리 및 파일은 실제로 사용중인 파일들이고, 파일의 유형은 이미지부터 문서 파일, 실행파일, 시스템 파일 등이 포함되어 있으며, 파일의 크기는 10KB에서 124MB까지 다양하게 구성되어 있다. 실험 환경의 디렉터리 및 파일의 구조는 [그림 5]와 같이 계층적 구조이며, 총 12,637개 파일과 12GB 크기의 용량으로 구성되어 있다. 실험에 사용된 악성코드를 가상화 환경에서 실행시키면 디렉터리의 파일들은 대칭키 알고리즘으로 암호화되고 비대칭키로 키를 암호화한 후 파일의 확장자를 랜섬웨어 유형에 따라 변경하여 저장한다.

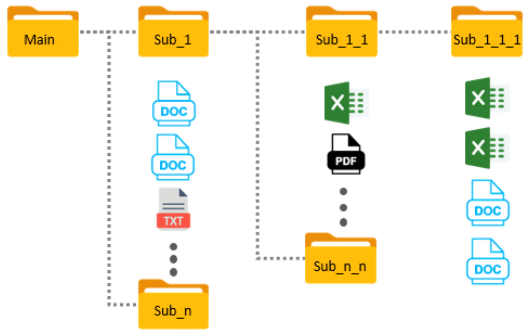

그림 5. 실험 환경 디렉터리 및 파일구조
Fig. 5. Experimental environment directory and file structure

제안된 방법론에 따라 파일 암호형 랜섬웨어를 실행한 후에 탐지까지 소요된 평균 시간은 6초가 소요되었으며, 파일 유형이 다양하거나 생성일, 수정일이 오래된 파일들이 많은 경우 탐지 시간이 단축되었다. 오탐이 발생하는 경우를 방지하기 위하여 파일이 복사되는 경우나 디렉터리를 넘나들어 압축을 실행하는 등 새롭게 파일이 생성되는 경우는 탐지 대상에서 제외한다.

[그림 6]은 실험 환경에서 파일 암호형 랜섬웨어를 실행한 후에 파일들이 암호화되는 시간을 측정한 결과이며, x축의 암호화되는 파일의 개수가 증가할수록 y축의 암호화에 걸리는 시간도 함께 증가하는 것을 확인할 수 있다. 시스템의 정상 파일 1,000개가 파일 암호형 랜섬웨어에 감염되어 암호화가 수행되는 시간은 약 1.97초가 걸렸으며, 이상 행위를 얼마나 신속하게 탐지하는 것이 시스템 운영과 복구에서 핵심 요소이다.

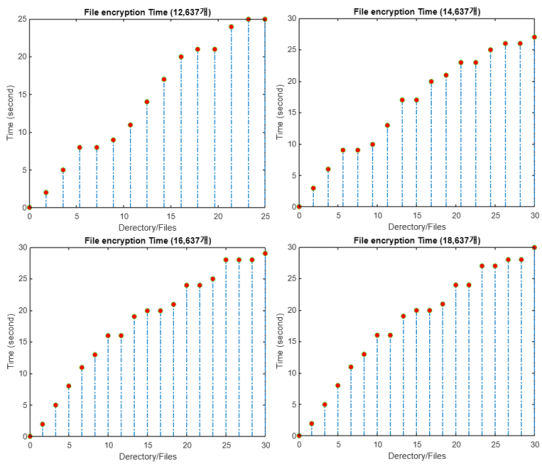

그림 6. 파일 암호형 랜섬웨어의 파일 암호화 시간
Fig. 6. File encryption time for file encrypting ransomware

실험 결과에서 제안된 방법론은 파일 생성일 또는 수정일이 오래된 경우 장기 미사용에 따른 백업 파일이거나 파일 소유자가 없는 무적파일이므로 파일 확장자 변경이 발생하면 높은 가중치를 부여했으며, 부모 디렉터리부터 하위 자식 디렉터리까지 동일 유형(.pdf, .doc, .hwp 등)의 파일들이 실시간으로 변경되는 경우도 이상 행위로 판단해 높은 가중치를 부여했다. 부여된 가중치에 따라 파일 암호형 랜섬웨어로 판단하는 기준값(Threshold)에 빠르게 도달했으며, 정확도도 증가하였다.

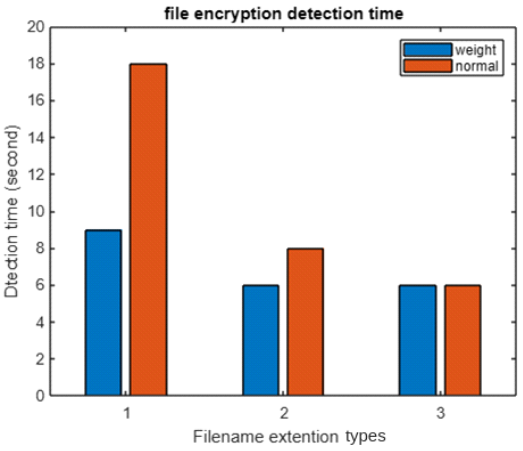

그림 7. 파일 암호형 랜섬웨어 탐지 시간
Fig. 7. Detection time for file encrypting ransomware

[그림 7]의 실험 결과에서 자식 디렉터리에 파일 유형이 한 종류면서 가중치 값이 없으면 18초의 탐지 시간이 걸렸으며, 파일 유형이 두 가지 종류이면서 가중치 값이 없으면 8초의 시간이 소요되었고, 가중치가 부여되면 6초의 시간이 걸렸다. 마지막으로 파일 유형이 세 가지 이상이면 가중치 여부와 관계없이 파일 암호형 랜섬웨어 탐지는 평균 6초의 시간이 소요되었다. 일정 시간 동안 이벤트가 발생하지 않으면, 탐지 결과에 가비지값(Garbage value)이 남아 있지 않도록 초기화 값을 '0'으로 설정했다.

Ⅵ. 결 론

본 연구에서는 파일 암호형 랜섬웨어에 대한 특징과 마이크로 세그먼테이션을 활용한 이상 행위 탐지방안을 살펴보고 실험을 통하여 결과를 분석하였다. 파일 암호형 랜섬웨어에 감염되면 문서, 동영상 및 이미지 파일들은 암호화되지만, 운영체제 설치 시 생성되는 시스템이나 프로그램 디렉터리는 파일 암호형 랜섬웨어에 의한 암호화가 수행되지 않는 특성을 갖는다. 실험에 사용된 파일의 개수는 12,637개 크기는 12GB이며, 동영상 및 이미지, 오피스 문서 등 다양한 유형의 파일을 포함하고 있다. 파일 암호화 랜섬웨어를 실행한 후 전체 파일이 암호화되기까지 소요된 시간을 측정한 결과 25초가 걸렸다. 사용자 컴퓨터나 서버 시스템이 파일 암호화 랜섬웨어에 감염되면 수분 이내에 모든 파일이 대칭키 및 비대칭키 알고리즘에 의하여 암호화되어 피해자가 액세스할 수 없을 것이다. 결과적으로 파일 암호화 랜섬웨어를 신속하게 탐지하고 차단하는 것이 악의적 행위로부터 안전하게 파일을 보호하는 최선의 방법이다. 파일 암호형 랜섬웨어에 감염되면 피해자의 파일들이 복호화가 불가능한 방식으로 암호화되고, 공격자는 피해자들의 파일을 인질 삼아 비트코인 같은 금전을 요구한다.

제안된 방법론은 파일 암호형 랜섬웨어에 감염되어 파일이 암호화되면, 마이크로 세그먼테이션의 영역과 블록화 속성 정보 그리고 이상 행위 탐지 알고리즘을 활용해 빠르게 랜섬웨어 감염 여부를 탐지하도록 설계하고 프로토타입을 구현하여 성능을 측정하였다. 파일 암호형 랜섬웨어가 탐지되면 사용자에게 알림을 제공하거나 암호화 프로세스가 더는 진행하지 못하도록 프로세스 강제 중지 및 시스템 종료를 수행할 수 있다. 본 논문의 연구 결과는 사용자 컴퓨터뿐 아니라 네트워크를 통해 중앙으로 관리 및 운영되는 공유 디렉터리에 활용될 수 있다. 사용자 컴퓨터는 개인적인 업무나 관심 파일들을 관리하지만, 기업의 공유 디렉터리는 주요 파일들을 집중적으로 관리하는 데이터베이스 역할을 하기 때문이다. 공유 디렉터리가 파일 암호화 랜섬웨어에 감염되어 조직 전체로 피해가 확산하는 것을 빠르게 차단할 수 있다면, 기업의 서비스가 중단되거나 중요 파일이 암호화되어 공격자로부터 금전적 피해가 발생하는 것을 최소화할 수 있다.

References

[1] S. Cobb, "*Ransomware vs printing press? US newspapers face foreign cyber-attack*," https://www.weiivesecurilty.com/2018/12/31/ransomware-printing-press-newspapers/, Last access:

Apr. 2023.

[2] TrendMicro, "*Report: Huge increase in ransomware attacks on businesses*," https://www.trendmicro.com/vinof/us/security/news/threat-landscape/report-huge-increase-in-ransomware-attacks-on-businesses, Last access: Apr. 2023.

[3] Sophos, "*The state of ransomware 2020*," Technical Report, https://www.sophos.com/en-us/medialibrary/Gated-Assets/white-papers/sophos-the-state-of-ransomware-2020-wp.pdf, Last access: Apr. 2023.

[4] A. Continella, A. Guagnelli, G. Zingaro, G. D. Pasquale, A. Barenghi, S. Zanero, and F. Maggi, "A self-healing, ransomware-aware filesystem," in *Proc. 32nd ACSAC 16*, ACM Press, 2016. (https://dx.doi.org/10.1145/2991079.2991110.)

[5] A. Kharraz, S. Arshad, C. Mulliner, W. K. Robertson, and E. Kirda, "A large-scale, automated approach to detecting ransomware," in *USENIX Security Symp.*, 2016.

[6] KISA, "Ransomware latest trend analysis and implications," *KISA Insight*, vol. 02, Aug. 2021.

[7] Boan news, "*Rust-based malware 'Nevada' ransomware in domestic circulation*," https://www.boannews.com/media/view.asp?idx=115325, 2023.

[8] *Cisco blog*, https://blogs.cisco.com.

[9] M. Hasan and M. Rahman, "RansHunt: A support vector machines based ransomware analysis framework with integrated feature set," in *20th ICCIT*, 2017.

[10] B. V. Reddy, G. J. Krishna, V. Ravi, and D. Dasgupta, "Machine learning and feature selection based ransomware detection using hexacodes," *Evolution in Computational Intelligence*, pp. 583-597, Singapore, 2021.

[11] S. K. Shaukat and V. J. Ribeiro, "RansomWall: A layered defense system against cryptographic ransomware attacks using machine learning," in *2018 10th COMSNETS*, pp. 356-363, 2018.

[12] Z. G. Chen, H. S. Kang, S. N. Yin, and S. R. Kim, "Automatic ransomware detection and analysis based on dynamic API calls flow graph," in *Proc. Int. Conf. Res. in Adaptive and Convergent Syst.*, pp. 196-201, 2017.

[13] R. Vinayakumar, K. Soman, K. S. Velan, and S. Ganorkar, "Evaluating shallow and deep networks for ransomware detection and classification," in *2017 ICACCI*, pp. 259-265, 2017.

[14] Wikipedia, "*Microsegmentation*," https://en.wikipedia.org/wiki/Microsegmentation_(network_security), Last access: Apr. 2023.

[15] Malware Traffic Analysis, *File Encryption Ransomware*, https://malware-traffic-analysis.net/2016/10/17/index.html, Last access: April 2023.

[16] IT WORKLD, *How file-encrypting malware ransomware works and how to remove it*, https://www.itworld.co.kr/news/156457, Last access: Apr. 2023.

서 정 우 (Jung-woo Seo)

2014년 2월 : 고려대학교 정보보호대학원 석사졸업

2018년 8월 : 고려대학교 정보보호대학원 박사졸업

2022년 1월~현재 : ICT폴리텍대학 정보보안학과

<관심분야> 네트워크포렌식, 랜섬웨어, 침해대응, UAM