목차

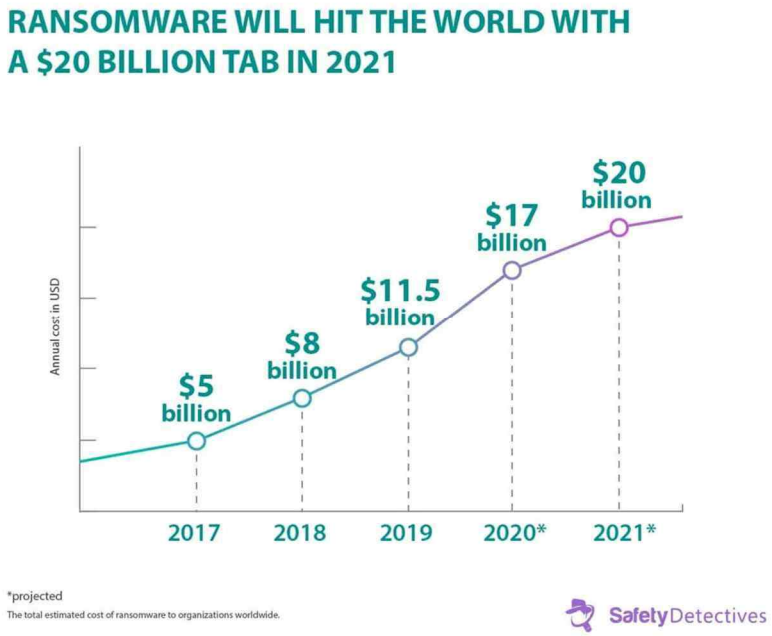
1. 과제의 배경 및 목표

① 과제 배경

랜섬웨어는 컴퓨터 시스템을 감염시켜 접근을 제한하고 일종의 몸값을 요구하는 악성 소프트웨어의 한 종류입니다. 주로 Email 을 통해 전파되지만 네트워크를 통해 직접 감염되기도 합니다. 사용자 운영체제의 취약점을 파고들어 사용자 모르게 정보를 탈취하는 것을 목표로 했던 이전의 악성코드들과 다르게

랜섬웨어는 사용자의 파일 시스템, 데이터베이스에 대한 암호화(Crypto) 혹은 락킹(Locker) 공격을 감행하여 중요 문서, 데이터에 대해 접근하지 못하게 합니다. 인질로 잡은 정보에 접근할 수 있는 권한을 복구해주는 대가로 사용자에게 직접 금전적인 요구를 하는 일반적인 패턴을 갖고 있습니다.

2021 년 현재 랜섬웨어는 해커 집단에서 가장 높은 인기를 구가하는 악성 소프트웨어입니다. 많은 상용 서비스들과 사용자 PC 에 대한 공격이 성공하면서 해커들이 대가로 요구하는 가격이 높아져 범죄 행위의 수익성이 높아진 탓으로 알려져 있습니다. [그림 1]은 2017 년 이후 랜섬웨어로 인한 피해액의 증가추세를 보여줍니다.

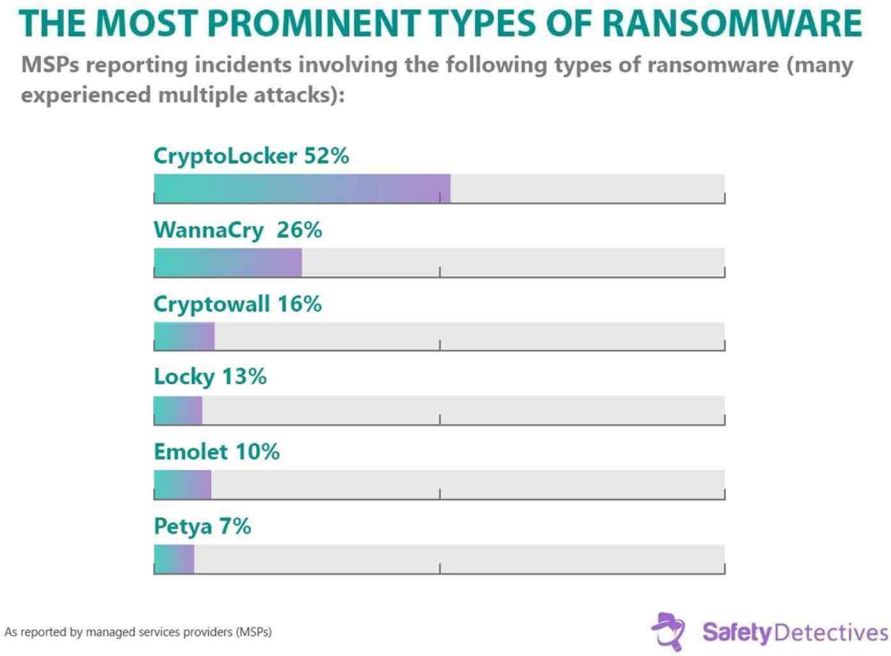


[그림 1]

랜섬웨어로 인한 피해액은 수백억 달러를 뛰어넘는 규모로 해가 지날 수록 가파르게 증가하고 있습니다. 해커들은 피해자에게 협박금을 요구할 뿐만 아니라 협박급을 기다리는 동안 위협이 실제라는 것을 보여주기 위하여 피해자의 파일을 실제로 손상하거나 삭제하기도 합니다.

사이버 범죄자의 궁극적인 행위와 상관없이, 실제적인 랜섬웨어로 인해 치르게 되는 비용은 단순 협박금을 훨씬 넘어섭니다.비교적 보안에 취약한 공공기관을 노린 공격으로 인해 사회 시스템이 정지되기도 합니다. 2017 년, WannaCry 는 세계 정부 조직, 대중교통, 국내 통신 회사, 글로벌 물류 기업, 다수의 대학과

관련된 시스템을 중단시키며 세계적인 주목을 받았습니다. [그림 2] 에서 볼 수 있듯 WannaCry 는 미국에서만 보고된 사건의 거의 절반을 차지하고 있습니다.



[그림 2]

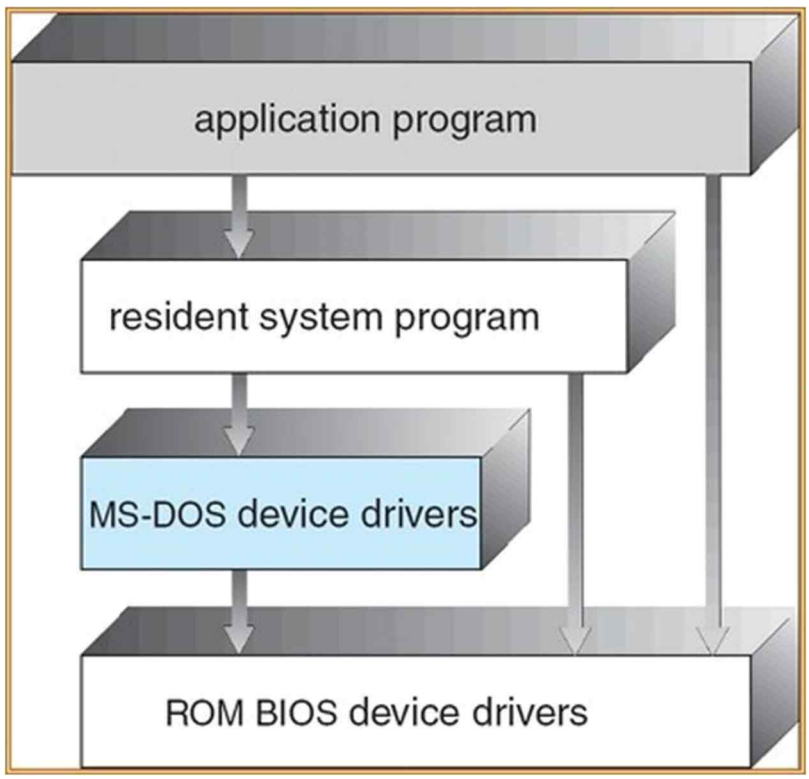
이전부터 다양화, 고도화되는 랜섬웨어 위협을 막을 수 있는 대응책들이 여럿 제시되었습니다. 대표적으로 정적 분석, 동적 분석등을 통해 악성코드를 분석하여 대응책을 마련하는 방법이 자주 쓰입니다.

정적 분석은 실행시키지 않고 실행파일의 코드를 분석하여 악성 소프트웨어를 탐지해내는 방법입니다. 이전에 알려진 악성 소프트웨어에 대해서는 정확도가 높고 편리하지만 알려지지 않은 새로운 유형의 악성 소프트웨어에는 취약한 특성을 보입니다. 특히 악성 코드 제작자가 이를 우려하여 사용자가 코드를 알아볼 수 없도록 난독화 과정(Code Obfuscation)를 수행할 경우 대응이 어려워집니다.

동적 분석은 사용자에 의해 제어되는 가상환경을 생성하여 의심스러운 실행 파일을 직접 실행시켜서 동작을 관찰하여 악성 소프트웨어를 탐지해내는 방법입니다. 실행파일을 직접 실행시키기 때문에 정적분석보다 직관적으로 알 수 있습니다. 그러나 악성 소프트웨어가 가상환경에서 동작됨을 알아차리고 숨어버리는 기능같이 지능적인 방식을 사용한다면 분석에 많은 시간과 자원이 소모됩니다.

기존의 방식들 또한 각각의 장단점과 특징을 갖고 있습니다. 그러나 알려지지 않은 새로운 종류의 랜섬웨어에 취약하다는 공통점을 갖고 있으며 응용 프로그램의 계층에서 접근을 시도하기 때문에 API 탐지에 의존적입니다.

기존 방식들이 갖는 한계점들을 해결하기 위해 하위 Layer, 즉 Kernel Level 에서 랜섬웨어를 탐지 및 차단할 수 있는 시스템이 필요합니다. [그림 3]은 컴퓨터 시스템을 계층별로 간략화한 그림입니다.



[그림 3]

② 과제 목표

- Windows 운영체제에 적합한 Crypto-Ransomware Detecting Service 개발을 목표로 한다.
- 랜섬웨어가 암호화를 시도할 때 커널에 파일 I/O 를 요청하는 점을 이용하여, 커널 레벨에서 비정상적인 접근을 차단하는 시스템을 만든다.

1. 시스템을 구현하기 위해 랜섬웨어의 일반적인 Payload 에 대한 분석이 선행되어야 한다. 따라서 '분석용 가상환경을 생성'하여 '샘플 랜섬웨어에 대한 분석'을 진행한다.

2. 파일 시스템 I/O request 를 감시하고 차단할 수 있는 '커널을 이용한 파일 I/O 모니터링 및 제어 모델(Monitor)'을 개발한다.
3. 감시 시스템으로부터 전달받은 I/O request 에 대해 정상 / 비정상을 판단하는 '비정상적 파일접근 탐지 모델(Detector)'을 개발한다.

2. 구현 사항 분석

① 분석용 가상환경 생성 및 랜섬웨어 분석

- Sandbox, Emulator 혹은 Virtual Machine 을 통해 샘플 랜섬웨어를 실행하고 분석할 수 있는 환경을 조성한다.
- 일부 지능적인 랜섬웨어는 에뮬레이터 환경에서 동작하지 않으므로 실제 사용자 환경에 가깝게 가상환경을 조성해야 한다.
- 중요 파일들은 지역적으로 밀집해야 하고, 최근 문서에 많은 파일이 등록되어 있어야 함. 사용자 스토리지에는 문서 파일뿐만 아니라 동영상 파일, 오디오 파일 등 다양한 타입이 존재해야 함
- 랜섬웨어의 일반적인 Payload 에 대한 가설을 세우고 (Read - Write - Delete), 동적 분석을 통해 가설을 검증한다.
- 커널에 전달되는 I/O request 에 대한 패턴을 작성한다.

② 커널을 이용한 파일 I/O 모니터링 및 제어 모델(Monitor)

- 파일 I/O 모니터링 및 제어 모델 생성
  - Pid 값을 통하여 각각의 프로세스를 식별 가능하며, 이를 이용하여 특정 프로세스의 파일접근을 제어할 수 있다.
  - 정보 송수신 모델로부터 받은 정보를 이용해 랜섬웨어의 프로세스로 인지하고 해당 프로세스의 파일 접근권한을 차단할 수 있다.
- 파일 접근정보 송수신 모델
  - 차단해야 할 프로세스에 대한 정보와 파일 처리 요청에 대한 정보를 파일 I/O 모니터링 및 제어모델과 공유할 수 있다.
  - 이 정보는 소켓을 이용해 유저영역에서 동작하는 Detector 와 주고받을 수 있다.

- 소켓으로 송수신하는 메세지의 내용은 파일 I/O 모니터링 및 제어모델이 특정한 프로세스가 파일에 접근할 경우 탐지했던 pid 와 시간, 파일경로가 있다. 소켓은 해당 정보를 보관하고 있다가 Detector 와의 연결이 확인되면 Detector 에게 메시지를 송신할 수 있다.

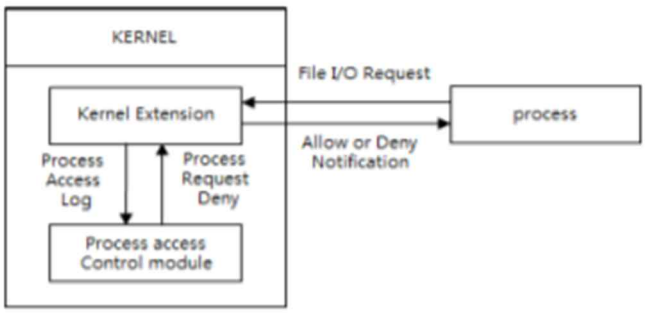


Fig. 1. File I/O Monitor and Control model

[그림 4]

③ 비정상적 파일접근 탐지 모델(Detector)

- 파일 접근 정보 수신 및 파일 접근 명령 송신 모델
  - 특정 프로세스가 커널에게 파일 접근을 요청할 경우 파일 I/O 모니터링 및 제어모델은 관련 정보를 담은 메시지를 Detector 에게 송신할 수 있다.
  - Detector 의 정보 송수신 모듈이 수신한 메시지는 추후 에러가 발생할 경우 더 능동적으로 대처할 수 있도록 텍스트파일에 수신한 내용을 저장할 수 있다.
  - 커널의 정보 송수신 모델이 메시지를 송신하는 시점을 모르기 때문에 Detector 의 정보 송수신 모델은 계속 대기하면서 메세지를 수신할 때마다 텍스트파일에 메세지를 추가해야 한다.
- 이상징후 탐지 모델
  - 탐지 정확도에 영향을 미치는 변수 M, N, T 등은 반복된 실험을 통해 알아내야 한다.
  - PID, 파일 접근 시간, 파일의 전체 경로에 대한 정보를 구조체 형태로 저장해 빠른 검색 및 비교가 가능하다.

- M 개의 메시지 중에서 동일한 프로세스가 N 개 이상이면 동일한 프로세스의 메시지 들을 가져온다. 가져온 메시지 중에서 가장 최근의 메시지와 가장 오래된 메시지의 시간차이가 t 초 이하일 경우 비정상적인 접근으로 판단해야 한다.
- 비정상적인 파일접근으로 판단할 경우, 정보 송수신 모델과 정보를 공유하고, 이 정보를 담은 메시지를 커널의 정보 송수신 모델에 전송한다. 메시지를 수신하면 해당 정보를 파일 I/O 모니터링 및 제어 모델이 확인하고, 해당 프로세스의 파일 접근권한을 차단할 수 있다.
- pid 를 기준으로 비정상적으로 파일을 접근한 프로세스의 명령어 또는 경로를 확인하고 pid 와 관련 정보를 사용자에게 알려줄 수 있다.
- 비정상적으로 파일에 접근하는 프로세스를 사용자에게 알리고 차단여부를 사용자가 결정할 수 있다.

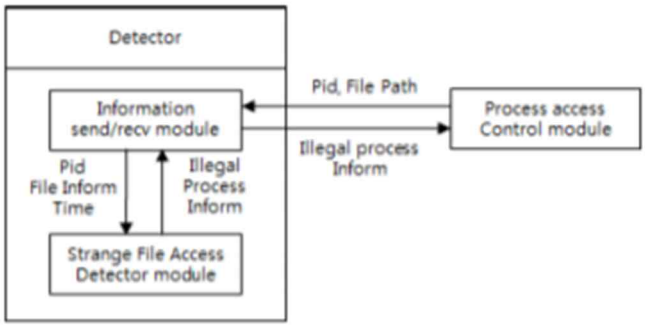


Fig. 2. Strange File I/O Detector model

[그림 5]

3. 현실적 제약 사항 및 대책

① 제약사항

- 테스트에 필요한 실제 랜섬웨어를 구하기 어려움
- 실제 랜섬웨어를 이용한 테스트 단계에서 False Positive 를 줄이기 위한 방안이 필요함

② 해결방안

- 랜섬웨어를 직접 제작하거나 교육용 랜섬웨어를 배포 받아서 사용

- False Positive 가 낮은 패턴을 찾기 위해 머신러닝을 이용해 시스템의 전체적인 Accuracy, Precision 을 높일 수 있음

4. 설계 문서

① 개발환경

- 실행 환경 : Windows 10, Virtualized Windows
- 개발 도구 : Cuckoo Sandbox, WDK(Windows Driver Kit), IFS(Installable File System) Kit, Visual Studio 2019, DDK
- 개발 언어 : C

② 사용 기술

File system Filter driver Framework

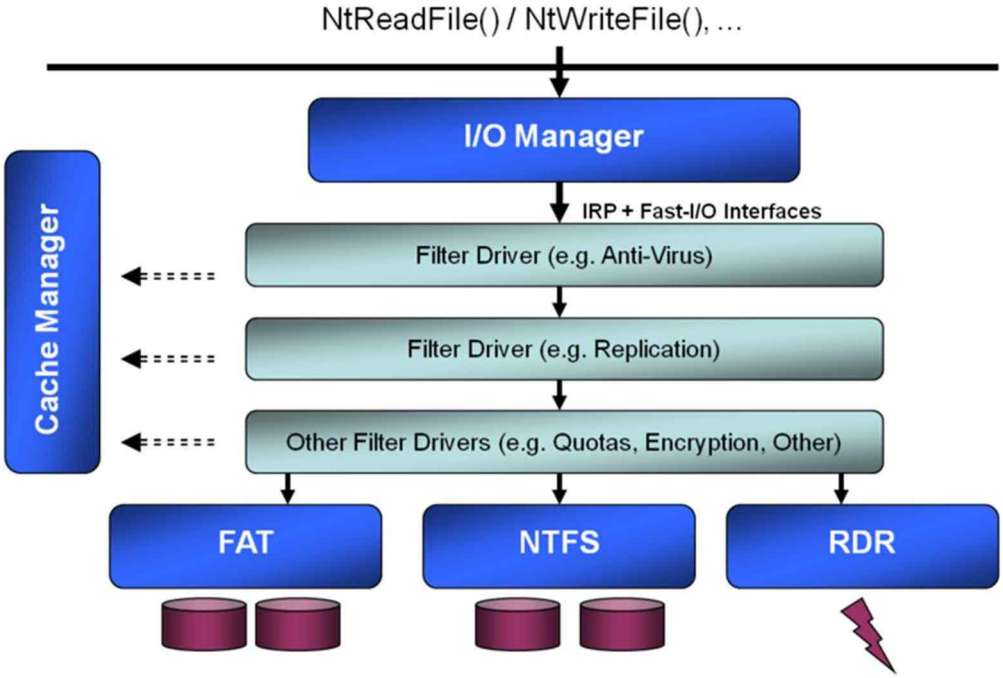
- 파일 시스템 필터 드라이버는 파일 시스템에 대한 요청을 가로챌 수 있음.

  -> API 에 의존하지 않고 응용프로그램의 I/O Operation 을 감시할 수 있다.

- 원래 의도했던 대상에 도달하기 전에 가로챔으로써 이 대상이 제공하는 기능을 확장하거나 교체할 수 있다.

  -> 유저 모드 프로세스의 파일 수정 요청에 대해 감시할 수 있다.

- 바이러스 백신 필터, 백업 에이전트, 암호화 제품 등 파일 시스템 및 파일 시스템 필터 드라이버를 개발할 때는 WDK(Windows Driver Kit)와 함께 제공되는 IFS(Installable File System)를 사용

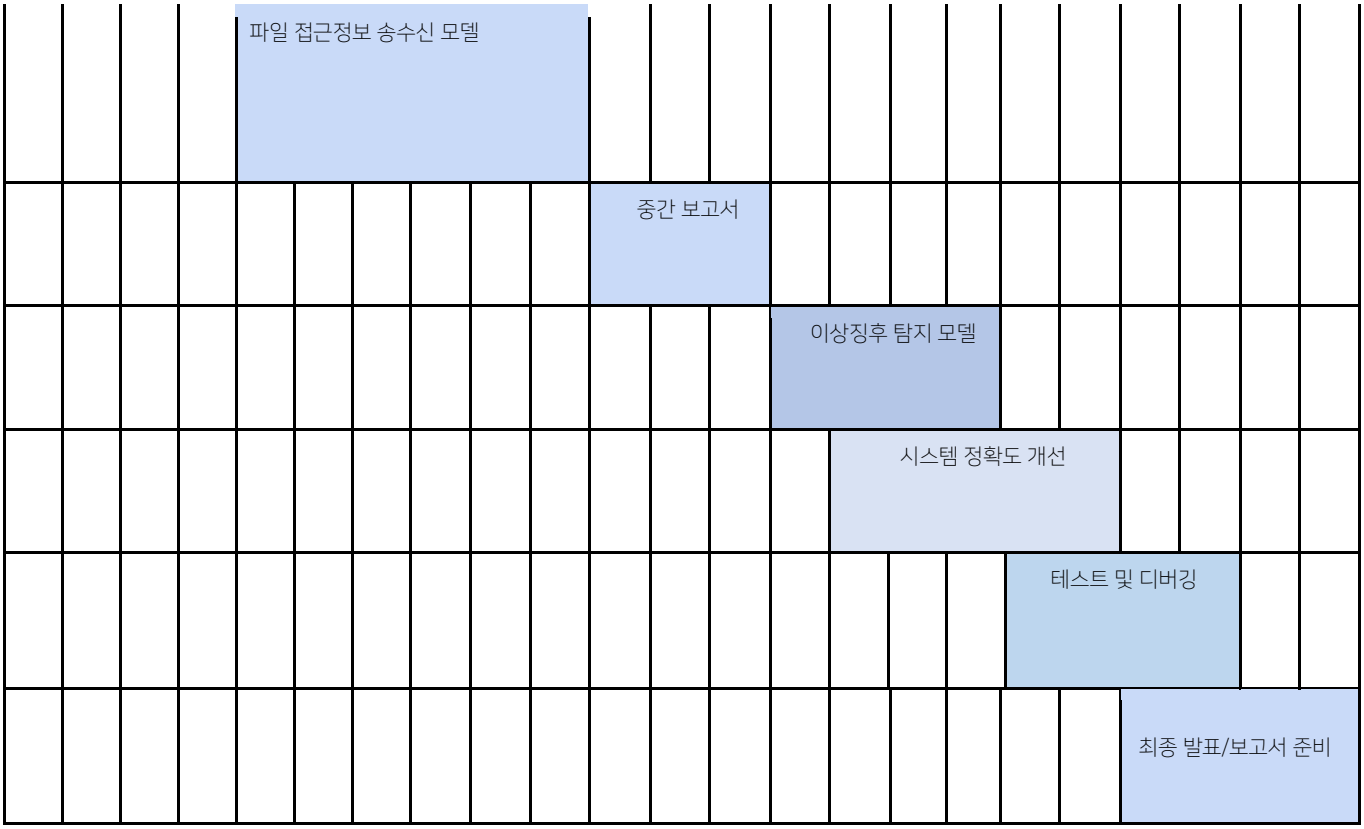


[그림 6]

5. 개발 일정 및 역할 분담

① 개발 일정

5 월			6 월					7 월					8 월					9 월				
3 주	4 주	5 주	1 주	2 주	3 주	4 주	5 주	1 주	2 주	3 주	4 주	5 주	1 주	2 주	3 주	4 주	5 주	1 주	2 주	3 주	4 주	5 주
관련 기술 공부																						
				파일 I/O 모니터링 및 제어 모델 생성																		



② 역할분배

이름	담당 영역
이재원	- 랜섬웨어 메커니즘 분석 및 패턴화 - 탐지 모니터 구현 - Github 저장소 관리
서희재	- 샘플 랜섬웨어 수집 - 랜섬웨어 분석용 가상환경 생성 - 차단 시스템 구현

공통	- 회의록 작성 - 보고서 작성 - 드라이버 개발

③ 참고 논문 / 사이트

2021 랜섬웨어에 대한 사실, 경향 및 통계

(https://ko.safetydetectives.com/blog/ransomware-statistics-ko/)

Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions

(https://www.sciencedirect.com/science/article/pii/S016740481830004X)

파일 I/O Interval 을 이용한 랜섬웨어 공격 차단 방법론

(https://doi.org/10.13089/JKIISC.2016.26.3.645)

UNVEIL: A Large-Scale, Automated Approach to Detecting Ransomware

(https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/kharaz)