Abstract

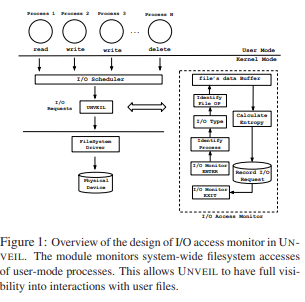
1. Introduction
2. Background
3. UNVEIL Design
   1. Detecting File Lockers
      1. Generating Artificial User Environments

최근의 영악한 악성코드들은 Fingerprinting 기법을 이용해서 현재 환경이 악성코드 분석환경임을 알아챌 수 있으므로 분석 환경을 악성코드가 알아챌 수 없도록 설계해야 한다. VM에서 자동으로 생성된 유저 데이터나 초기 환경같이 초기 상태(bare metal)인 시스템 환경은 악성코드가 분석되고 있음을 알아차릴 수 있는 중요한 증거가 된다.

랜덤하고, 최근에 수정, 생성, 삭제된 파일이 잔뜩 있는 분석 환경을 조성하는 것이 좋다.

* + 1. Filesystem Activity Monitor

UNVEIL에서는 IO request를 다루는 data buffer에 직접 연결된 파일시스템 감시모니터를 제안했다. 유저/커널 모드에서 파일시스템 생성, 수정 등이 일어날 수 있는 여러가지 방법이 있지만 결국 이런 여러 방법들을 통해 I/O request sequence가 발생한다. 유저 쓰레드에서 I/O API를 사용하면 I/O request가 발생해서 파일 시스템 드라이버에 전달된다.



UNVEIL의 감시모니터는 user-mode에서 생성되는 모든 I/O request에 대해 callback을 설정한다.

저자는 여러 샘플들을 분석한 결과, 샘플들 중 다수가 분명하고 반복적인 I/O trace를 남기는 걸 확인하였다. Trace를 남기는 이유에 대해 유저로부터 파일에 대한 접근을 막기 위해 단 하나의 전략만을 사용하기 때문이라고 밝혔다.

그 결과로 랜섬웨어가 공격을 수행할 때 I/O에 대한 Access Pattern을 남긴다.

* + 1. I/O Data Buffer Entropy

UNVEIL은 I/O trace에 남은 읽기/쓰기 요청에 대해서 data buffer의 Entropy를 계산한다. 동일한 파일 오프셋을 기준으로 읽기/쓰기 요청에 의해 발생한 Entropy를 비교해보는 방법으로 Crypto-ransomware의 동작을 알아낼 수 있다.

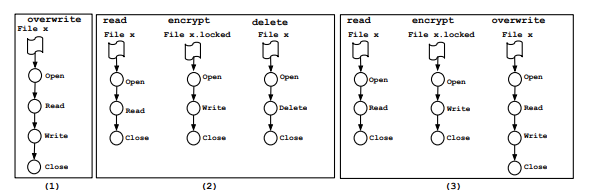
이것은 랜섬웨어의 동작 방식 – 오리지널 파일 읽기, 암호화, 기존 파일 암호화된 버전으로 덮어쓰기 – 때문이다.

* + 1. Constructing Access Patterns.

UNVEIL이 I/O trace를 생성한다음, 파일 이름과 Timestamp를 기준으로 정렬하여 몇 가지 정보를 얻을 수 있다.

파일 별로 I/O 접근 순서를 알아낼 수 있으며, 어떤 프로세스가 파일에 접근했는지 알아낼 수 있다. 중요한 점은 랜섬웨어의 프로세스가 파일에 접근한 순서와 동일한 순서를 얻을 수 있다는 점이다.

UNVEIL에서는 일련의 과정들 – 읽기, 암호화, 덮어쓰기 – 가 반복적으로 일어나는 경우 이를 인식해 알람을 보낸다.



위 그림은 실험동안 분류한 3가지 랜섬웨어 패밀리이다.

좌측은 Cryptolocker변종, CryptoWall 변종으로 같은 패턴을 갖는다. 이는 암호화과정에서Microsoft사의 CryptoAPI를 사용하기 때문이다. 기존의 파일을 암호화한 파일로 덮어씌운다.

가운데는 FileCoder Family에 속한다. 이 종류의 랜섬웨어는 기존 파일 이외에 암호화한 기존 파일을 따로 생성한다. 보통 랜섬웨어들이 기존 파일들을 storage에서 삭제하는데 반해, 이 종류의 랜섬웨어는 파일을 삭제하지 않고 단순히 link를 끊을 뿐이므로, 파일 복구 가능성이 높다.

우측은 기존 파일을 암호화한 버전으로 완전히 덮어씌운다. Window API를 이용하거나 직접 구현한다.

* 1. Detecting Screen Lockers

UNVEIL은 Screen Locker 계열의 랜섬웨어 또한 감지할 수 있다.

랜섬웨어 유포자들은 금전적인 이득을 위해 시스템을 감염시킨 후 Ransom Note를 띄우게 만드는데 이런 시도를 알아채면 Screen Locker 또한 감지할 수 있다.

1. UNVEIL Implementation
   1. Generating User Environments

랜섬웨어 에게 실제 사용자 환경인 것처럼 보이기 위해 미디어, 오디오, 사진, 문서 파일들을 적절히 배치되게 만든다.

파일들의 확장자는 실제 사용자가 사용하는 pptx, pdf, txt등이 혼재해야 하며, 열람 날짜, 수정 날짜, 생성 날짜가 최근이어야 한다.

* 1. Filesystem Activity Monitor

파일 시스템 활동 분석을 위해 몇 가지 방법들이 이전에 사용되었다. 그러나 API Hooking의 경우 이름을 변경하여 재사용하는 방법으로 회피가 가능하고, 알려진 API이외에 직접 작성한 알고리즘을 사용할 수 있기 때문에 SSDT 호출 또한 기술적인 한계가 존재한다. 그러므로 API나 System Call Hooking대신 Windows File System Mini filter Driver Framework을 통해 Filesystem 관련 I/O 활동을 모니터링 하는 방식을 사용한다.

Windows 환경에서 I/O request는 I/O request packet으로 표시된다.

UNVEIL의 감시 모니터는 유저모드 프로세스에서 생성된 모든 I/O request에 대해서 callback을 설정한다.

* 1. Desktop Lock Monitor

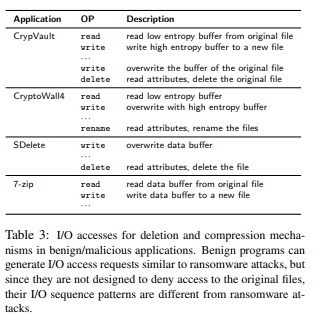
Desktop Locking ransomware가 출력하는 랜섬노트를 식별하기 위해서 외부에서 스크린샷을 찍고, 찍은 스크린 샷을 비교한다. 찍은 스크린 샷 간의 dissimilarity를 비교하기 위해서 Structural Similarity Image Metric을 구현한다.

1. Evaluation
   1. Experimental Setup

Cuckoo Sandbox에서 실험이 진행됨

* 1. Ground Truth (Labeled) Dataset
     1. Filesystem Activity of Benign Applications with Potential Ransomware-like Behavior

무해한 프로그램도 랜섬웨어-스러운 I/O trace를 남겨서 결과적으로 False Positive가 발생할 것이라는 우려가 있었다. 그러나 무해한 파일은 파일을 암호화 시키더라도 기존 파일을 삭제 시켜서 접근을 막지 않기 때문에 유저에 의해 기존 파일을 의도적으로 삭제 시키지 않는 한 FP는 나타나지 않는다.



위 그림을 보면 7-zip(압축프로그램)과 랜섬웨어 동작의 차이점을 확인할 수 있다.

7-zip은 기존파일에 대한 access를 막지 않는다. 따라서 랜섬웨어와 무해한 프로그램들의 I/O Trace는 구별이 가능하다.

* + 1. Similarity Threshold
  1. Detecting Zero-Day Ransomware
     1. Detection Results

다수의 읽기 및 쓰기 입력에 대한 I/O request가 발생하고 data buffer의 Entropy가 크게 증가한다면 True Positive로 분류한다.

특히 여러 개의 높은 엔트로피 파일을 생성하는 경우 신뢰할 말한 랜섬웨어의 지표가 될 수 있다. 그 이유는 낮은 Entropy로 암호화하더라도 결국 암호화된 버전의 파일을 생성해야 하기 때문이다.

* + 1. Early Warning
  1. Case Study: Automated Detection of a New Ransomware Family

1. Discussion and Limitations

I/O buffer를 감시하므로 비슷한 패턴(읽기 – 암호화 – 기존파일 삭제)을 따르는 모든 랜섬웨어는 탐지가 가능하다.

1. Related Work
2. Conclusions
3. 해석 불가

* Each I/O operation contains the process name, timestamp, operation type, filesystem path and the pointers to the data buffers with the corresponding entropy information in read/write requests
* In UNVEIL, user-mode process interactions with the filesystem are formalized as access patterns
* Comparing the entropy of read and write requests to and from the same file offset serves as an excellent indicator of crypto-ransomware behavior.
* Basing UNVEIL’s filesystem monitor on a minifilter driver allows it to be located at the closest possible layer to the filesystem with access to nearly all objects of the operating system.