다크웹 크롤러를 사용한 악성코드 탐지 및 분석

김아린*, 이은지*

*성신여자대학교 컴퓨터공학과

dkfls921@gmail.com, 20180993@gmail.com

Dark Web based Malicious Code Detection and Analysis

Ah-Lynne Kim*, Eun-Ji Lee*

*Dept. of Computer Engineering, Sungshin-Women's University

요 약

다크웹을 이용한 사이버 범죄율이 국내외에서 가파르게 상승 중이다. 그러나 다크웹의 특성상 숨겨져 있는 인터넷 영역에서 공유되는 악성코드들을 찾기란 어렵다. 특히 다크웹상 여러 서비스들은 크롤러 bot과 같은 정보 수집을 막고자 다양한 기법을 적용하고 있다.
따라서 우리는 기존의 연구 방법에 따라 다크웹 상의 URL을 수집한 후, 추가적으로 다운로더를 만들어 exe, zip과 같은 특정 형식의 파일을 수집하였다. 앞으로 해당 파일들은 통합 바이러스 스캔 엔진에서 검사하여 의심 파일들을 분별할 예정이다. 의심 파일들은 정적 / 동적 분석을 통해 상세한 보고서를 제출하여 향후 다크웹 내의 악성코드 분포 / 출처 분석에 유의미한 결과를 도출할 수 있다.

1. 서론

다크웹을 이용한 범죄가 급증하고 있으나 수사기관조차 대응법을 현실적으로 실현하기 어렵다. 다크웹 접속방법인 Tor 브라우저의 국내 사용자 수는 2017년에서 2018년 사이 약 2배의 사용자가 증가한 것으로 보인다. 그에 따른 범죄 발생 건수가 전년도 대비 약 20% 이상 증가했다.[1] 최근 다크웹 내의 악성코드/사이트로는 랜섬웨어와 같은 멀웨어, COVID-19와 관련된 피싱 사이트 등이 있다. 악성 코드 / 사이트를 이용한 공격은 다크웹 게시 정보 확대에 따라 웹 상의 악성코드 / 사이트 등이 전보다 빠르게 진화하고 있다.

2. 관련 적용 사례

2.1 Tor 네트워크

Mandeep Pannu[2]가 제안한 다크웹 크롤러에서 시스템 내 TOR 네트워크 웹 페이지에 포함될 수 있는 관련 링크 속성을 스크랩해서 의심스러운 악성 웹사이트의 DB를 생성한다. 이 데이터베이스는 자동으로 업데이트되며, 사용 가능한 작업 링크를 저장하면서 이전 버전의 TOR 사이트를 보관한다. 이를 통해, 사법기관은 이전, 현재 데이터베이스를 모두 검색하여 의심스럽고 악의적인 웹사이트를 탐지할 수 있다.

2.2 엔티티 지향 딥웹 사이트 크롤링

많은 딥웹 사이트들이 전통적으로 문서 지향적인 텍스트 콘텐츠(예: 위키피디아, 트위터 등)를 유지하고 있지만, 딥웹 사이트의 상당 부분이 기존과 반대로 구조화된 실체를 큐레이션하고 있다고 본다. 실체 중심의 심층 웹 사이트에 관해 구축한 프로토타입 시스템을 기술한다. 엔티티 지향 딥 웹 사이트의 특정 맥락에서의 쿼리 생성, 빈 페이지 필터링이나 URL 중복제거를 포함한 중요한 하위 문제에 대처하기 위해 효과적인 맞춤 기법을 제안한다. [3]

3. 분석 및 수행 방법

3.1 기존의 다크웹 크롤러

Trandoshan 크롤러는 스노우볼 샘플링을 활용하여 주어진 onion URL과 연결되는 외부 URL을 수집한다. 수집 이후 scheduler, persister, API와 같은 여러 프로세스들을 사용하여 URL의 완전무결성을 검증한다. Dashboard를 통해 데이터베이스에 저장된 URL 및 URL 제목을 검색할 수 있다.

이때 프로세스 간 통신 기법은 NATS로 알려진 메시징 프로토콜을 사용한다.

크롤러 내부 프로세스들은 크롤링 할 고유 URL을 받아서 프로세스 동시성이 가능하므로 성능이 향상된다.

그러나 Trandoshan 크롤러는 페이지 내에서의 특정 형식 파일 탐색 및 다운로드가 불가능하다. 악성코드 탐색 및 탐지를 해야 하는 본 연구에서 다크웹 URL 탐색만 가능한 해당 크롤러만 사용하여 진행할 수 없었다.

3.2 해결책 제안

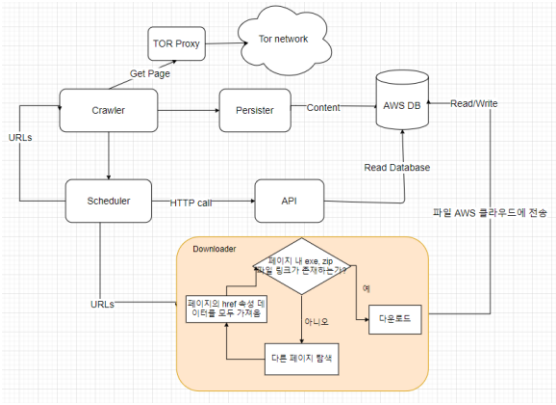
유명한 패키지인 Scrapy를 사용하여 다운로더를 만들었다.

크롤링 된 페이지 html 상에서 href 속성 텍스트를 추출했을 때 텍스트의 끝이 exe, zip로 끝나거나, 해당 클래스 이름에 download라는 텍스트가 포함되어 있으면 해당 링크를 사용하여 다운로드를 시도한다. 다운로드를 위해서는 절대 링크가 필요하므로 현재 사이트와 링크를 합쳐 절대 링크를 파일 URL로 연결한다.

파일을 다운로드할 때는 SHA1 해시 값으로 이름이 저장되므로 이름은 파일 경로로 설정하여 저장한다.

3.3 기존의 방법 대비 장점

기존 방법인 Trandoshan 크롤러는 API프로세스를 사용해 초기 URL에 연결된 모든 URL을 검색하고, 각 URL, title을 함께 보여준다.



<그림 1> Downloader 설계도

<그림1>은 본 논문에서 제시하는 D-Downloader(Darkweb Downloader)의 설계도이다. 전체적인 흐름은 다음과 같다.

1. Crawler가 초기 URL로부터 외부 링크를 수집한다.
2. 수집된 외부 링크는 AWS DB에 저장된다. 이때 Persister, API, Scheduler와 같은 함수를 통해 URL 내용을 검증하고 URL 내 내용을 DB에 삽입한다.
3. 저장된 URL을 바탕으로 D-Downloader는 URL 내의 href 속성 내용을 가져온다.
4. 속성 내용이 exe,zip 형식으로 끝나거나, 다운로드 링크가 첨부되어 있다면 해당 내용을 가져온다.
5. 해당 내용을 절대 링크화 시켜 다운로드를 수행한 후 AWS DB에 저장한다.
6. 다운로드가 수행될 동안 URL 내 속성 검색은 계속 되는데, 만일 페이지 내 관련 내용이 없다면 DB 내 다음 URL로 넘어간다.

URL만 수집해 오는 Trandoshan 크롤러와 달리 Downloader는 웹과 연결된 의심되는 모든 URL을 검색하고 exe 또는 zip 파일을 수집하고 탐지가 가능해 다크웹으로 인한 피해를 초기에 대응할 수 있는 가능성을 높여준다.

구분	Trandoshan	D-Downloader
개발 언어	Go	Python
검색 엔진	모든 URL을 검색하고 크롤링한 URL은 DB에 추가.	모든 URL을 검색하고 exe 또는 zip파일이 존재하는 URL은 DB에 추가.
장점	파일 수집 불가 URL분석 불가 확장 불가 URL 및 URL 내 header 수집	exe, zip 파일 수집 가능 URL 분석 가능 확장성 뛰어남 멀티 프로세스 사용으로 속도 향상 파일 다운로드 시 관련 링크 및 내용 수집 향후 악성코드 관련 통계 산출 가능

<표 1. Trandoshan 크롤러와 다운로더 차이점 비교>

4. 결론 및 향후 연구 방향성 제시

-Trandoshan 크롤러는 스노우볼 샘플링을 이용해서 외부 URL을 수집한다. 프로세스들을 거쳐서 URL의 고유성을 검증한다. 이는 Go 언어로 작성되어서 고성능 분산 시스템 구축이 가능하다.

-Downloader는 DB 내 URL을 크롤링하여 exe 혹은 zip 파일이 존재하는지 찾아주는 역할을 수행한다.

파일이 존재하는 URL은 별도로 저장하여 향후 악성코드 출처 분석에 쓰일 것이다.

URL을 수집해오는 기존 Trandoshan 크롤러의 기능에 덧붙여 웹 파일까지 수집할 수 있는 downloader를 구현하였다. 하지만 파일을 수집만 할 뿐, 유의미한 악성코드 파일 수집에 도달하지 못해 악성코드 의심 파일을 분별하는 데에는 한계가 존재했으므로 향후 추가 연구가 필요하다. 추가 연구를 통해 의심 파일을 정적/동적 분석하면 악성코드 현황, 출처와 관련된 통계 등을 알 수 있다.

향후 악성코드 의심 파일 분별 방식을 추가해서 악성 파일의 자동 분류, 출처 저장 등과 같은 악성 파일 수집 및 분류 기능을 보완한다면 체계적인 다크웹 악성코드 대응책이 될 것이다.

[본 논문은 과학기술정보통신부 정보통신창의인재양성사업의 지원을 통해 수행한 ICT멘토링 프로젝트 결과물입니다]

참고문헌

[1] 경찰청, 2019년 사이버위협 분석 보고서, 2019

[2] Pannu M., Kay I., Harris D.(2019) “Using Dark Web Crawler to Uncover Suspicious and Malicious Websites:”, Ahram T., Nicholson D. (eds) Advances in Human Factors in Cybersecurity. AHFE 2018. Advances in Intelligent Systems and Computing, vol 782. Springer, Charm.

[3]He, Y., Xin, D., Ganti, V., Rajaraman, S., & Shah, N. (2013, February). Crawling deep web entity pages. In *Proceedings of the sixth ACM international conference on Web search and data mining* (pp. 355-364).