리소스 점유 최적화를 이용한 고속 동적 크롤러 구현

High-speed Dynamic Crawler Implementation with Resource Occupancy Optimization

저자 (Authors)
조호묵, 이경석, 이정호
Ho-Mook Cho, Kyeong-Seok Lee, Jeong-Ho Lee

출처 (Source)
한국정보과학회 학술발표논문집 , 2019.6, 1138-1140(3 pages)

발행처 (Publisher)
한국정보과학회
The Korean Institute of Information Scientists and Engineers

URL
http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE08763447

APA Style
조호묵, 이경석, 이정호 (2019). 리소스 점유 최적화를 이용한 고속 동적 크롤러 구현. 한국정보과학회 학술발표논문집, 1138-1140

리소스 점유 최적화를 이용한 고속 동적 크롤러 구현

조호묵°, 이경석, 이정호
KAIST 사이버보안연구센터
chmook79@kaist.ac.kr, harvist@kaist.ac.kr, ddanzit@kaist.ac.kr

High-speed Dynamic Crawler Implementation with Resource Occupancy Optimization

Ho-Mook Cho°, Kyeong-Seok Lee, Jeong-Ho Lee
KAIST Cyber Security Research Center

요 약

최근 ICT 발전으로 인해 웹사이트가 폭발적으로 증가함과 동시에 웹을 통해 악성코드를 유포하는 사이버위협이 증가하는 추이를 보여 이를 효과적으로 탐지 및 분석할 수 있는 대응 연구가 활발히 이루어지고 있다. 하지만 공격 기법이 지속적으로 고도화되어 전통적인 분석 및 탐지 방식이 한계가 발생하였고, 방대한 웹사이트를 효과적으로 분석하기 위해 기존 탐지기술의 한계점을 극복해야 한다. 따라서 이러한 한계점을 극복하기 위해 리소스 점유 최적화 알고리즘을 적용한 고속 동적 크롤러를 제안한다. 제안방법은 브라우저가 점유하는 리소스를 최적화하여 고속 동적 크롤러의 병렬화를 구현하였고, 실험을 통해 크롤링 성능과 효율성을 검증하였다.

1. 서 론

최근 ICT의 발전으로 인해 웹사이트 수가 폭발적으로 증가함과 동시에 웹을 통해 악성코드를 유포하는 DBD(Drive-by download) 사이버 위협 또한 증가하였다. 이 공격은 웹사이트에 접속하였을 때 사용자 모르게 악성코드를 유포하는 사이트로 유도하여 취약한 환경의 컴퓨터를 감염시키는 기법을 사용한다[1].

DBD 공격에 대응하기 위한 기술은 크게 두 가지로 분류할 수 있다. 첫째, 웹 페이지의 소스 코드를 수집하여 분석하는 정적분석 기법으로 AV(Anti-Virus) 엔진과 같이 탐지 시그니처를 사용하거나 과거 탐지된 데이터와 비교하여 공격을 탐지한다. 시그니처 및 과거 탐지 데이터를 이용하기 때문에 동적분석에 비해 매우 빠르고 병렬화 구현이 비교적 쉽다는 장점은 있으나 탐지 시그니처의 지속적인 업데이트가 필요하고, 최근 공격기술에 사용되는 자바스크립트 난독화 기술의 고도화에 따른 미탐률 및 오탐률이 높다는 한계점이 있다. 둘째, 실제 브라우저나 에뮬레이터를 이용하여 웹사이트에 접속하고 시스템 변화를 분석하는 동적분석 기법으로 악성코드가 유포되는 시점에서의 파일, 레지스트리, 프로세스 등 시스템 변화를 분석하여 공격을 탐지한다. 정적분석 방법에 비해 미탐률 및 오탐률이 낮고, 난독화 등 Anti-Crawler에 효과적으로 대응할 수 있는 장점은 있으나 병렬화 구현이 어렵고 실제 브라우저나 에뮬레이터를 이용하기 때문에 성능이 매우 저조한 한계가 있다. 하지만 최근 난독화, Crawler bot 우회 등 Anti-Crawler 기술이 지속적으로 고도화됨에 따라 동적분석 기법을 이용하여 DBD 공격을 탐지하는 연구가 활발히 진행되고 있다[2,3].

본 논문에서는 동적분석 기법의 병렬화 및 낮은 성능 문제점을 개선하여 방대한 웹사이트를 효과적으로 분석할 수 있도록 리얼 브라우저 기반 고속 동적 크롤러(HDC, High-speed Dynamic Crawler)를 제안하며, 이 제안 크롤러를 이용하여 10만 개 국내·외 도메인을 대상으로 크롤링 성능 및 효율성을 측정하는 실험을 진행하였고, 기존의 동적 크롤러와의 비교를 통한 시스템 구현 측면의 경제성 및 웹사이트를 효과적으로 분석할 수 있는 성능의 효율성을 검증하였다.

2. 관련연구

2.1 악성코드 유포 네트워크

공격자는 PC에 악성코드를 감염시키기 위해 수 개의 웹사이트를 논리적으로 연결시켜 악성코드 유포지로 자동으로 유도되는 네트워크 즉, MDN(Malware Distribution Network)을 구축한다[3]. 공격자는 악성코드 유포 네트워크를 만들 때 사용자의 행위 없이 자동으로 연결될 수 있도록 javascript, iframe, redirection 등을 이용한다. 또한, MDN 분석을 어렵게 만들기 위해 링크 정보를 난독화 한다[2,3]. 이를 통해 취약한 PC가 MDN에 접속하는 것만으로 악성코드에 감염된다[2].

2.2 악성코드 유포 네트워크 분석 방법

악성코드 유포지 네트워크를 분석 및 탐지하기 위한 대표적인 연구는 정적분석과 동적분석이다.

정적분석은 웹사이트에 포함된 비정상 콘텐츠를 분석하는 시그니처 기반 분석 방법과 웹사이트의 메타정보를 분석하여 알려진 악성 메타정보와 비교하는 방법이 있다. 정적분석은 분석을 위해 난독화된 콘텐츠를 복호화해야 하는 과정이 필요하며, 신규 난독화 기술에 효과적으로 대응하기에 한계가 있다[2,3].

동적분석은 가상머신 또는 에뮬레이터를 사용하여 분

석 대상 웹사이트에 직접 방문한 후 시스템 변화를 분석하는 방법이다. 동적분석은 실제와 유사한 분석환경을 사용하여 정적분석의 한계점인 난독화에 대한 복호화 과정이 필요 없다는 장점이 있으나 하드웨어 감지, 실행환경 감지, 분석행위 감지 등 분석환경을 회피하는 기술이 적용된 악성코드 유포 네트워크에 대한 효율적인 대응이 제한되며, 분석환경이 악성코드에 감염될 수 있는 위험성 등이 상존한다[3]. 또한 병렬화의 어려움으로 인해 성능이 매우 저조하다는 가장 큰 단점이 있다.

2.3 크롤링 성능 향상 방법

Internet Live Stats에 따르면 전 세계 약 17억 개의 웹사이트가 존재한다고 보고되고 있으며, 이 많은 웹사이트 중에 섞여 있는 악성 웹사이트를 효과적으로 분석 및 탐지하기 위해 크롤링 성능은 매우 중요한 이슈 중에 하나다. 크롤링 성능을 효과적으로 향상하기 위해 대표적으로 병렬화 기술을 사용하는데, 이는 크롤러 에이전트를 병렬화하여 성능을 향상하거나, 장치 자원을 확장하는 방식으로 병렬화를 구현한다.

크롤러 에이전트의 병렬화 방법은 전통적으로 정적분석 기법에서 많이 사용되는 방법으로 브라우저나 에뮬레이터를 사용하지 않고 HTTP Request/Response 명령어를 이용하여 웹페이지의 소스 코드만을 수집하기 때문에 병렬화 기능을 구현하기 쉽다. 이에 비해 동적분석 기법은 브라우저나 에뮬레이터를 기반으로 구현되기 때문에 병렬화 기능을 구현하는 것이 어려울 뿐만 아니라 병렬화 기능을 구현하더라도 브라우저 또는 에뮬레이터의 동시 실행 수가 증가하면서 리소스 점유율도 증가하여 크롤링 성능 저하 및 효율성이 떨어지는 문제가 발생한다.

장치 자원 확장 병렬화 방법은 정적분석 및 동적분석 기법에 모두 사용할 수 있고, 최근 가상화 시스템의 발달로 인해 병렬화 구현이 더욱 쉬워졌다. 하지만 장치 자원을 무한대로 확장할 수 없기 때문에 근본적인 병렬화 대책으로는 적합하지 않다.

크롤링 성능 향상을 위해 표 1과 같이 수행 방식 개선을 알고리즘 적으로 접근하여 다양하게 연구되었으나 대부분이 정적분석 기법에 의존적인 방법으로 동적분석 기법의 성능 이슈를 개선하기에는 한계가 있다[4].

본 논문에서는 방대한 웹사이트를 효과적으로 분석하고 기존의 DBD 탐지 기반기술의 제한점을 극복하기 위해서 OS 수준의 리소스 점유 경량화, 가비지 메모리 최소화, 브라우저 리소스 점유 최적화를 통해 HDC를 제안하고자 한다.

표 1. 수행 방식별 크롤러 현황

크롤러	수행 방식
Focused	관련 있는 웹페이지들을 집중적 분석 크롤링
Parallel	여러 프로세스로 구성된 크롤링
Distributed	중앙서버를 통하여 분산된 노드의 통신과 동기화를 통한 크롤링
Incremental	AllUrl, Ranking Module에 따라 페이지의 변화 추정치를 기반으로 수행하는 크롤링
Breadth-first	Breadth first search 알고리즘을 기반으로, 처음 접근한 페이지 이외의 링크를 접근하는 크롤링

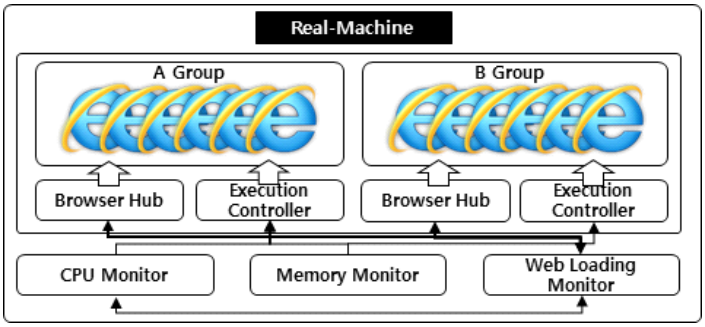

그림 1. HDC 구성도

3. HDC

본 논문에서 제안한 HDC는 그림 1과 같이 Browser Hub, Excution Controller, CPU Monitor, Memory Monitor, Web Loading Monitor로 구성되며, 크롤러 장치는 OS 수준에서 시스템 경량화를 표 2와 같이 진행한다.

3.1 Browser Hub

브라우저를 병렬화하여 실행하기 위해서 Browser Hub는 복수의 IE(Internet Explore) 프로세스를 실행시키는 기능과 장애가 발생한 IE 프로세스를 종료 및 재실행 시키는 기능, IE 동작 상태 인식 기능으로 구성된다. 또한 하나의 Browser Hub에 의해 제어되는 IE 프로세스 수는 PC 사양에 따라 다르지만 i7급(CPU:3.2Ghz, MEM:16GB, Disk:SSD256G)에서 15~17개로 한정한다. 이는 IE가 복수로 구동되면서 CPU, I/O 등 시스템 전체 리소스 점유율이 높아지면서 크롤링 성능과 효율성이 현저히 떨어지는 문제가 발생하는데 IE의 경우 16개 이상의 프로세스가 동시에 실행되면 리소스 점유율 증가 현상이 발생하기 때문이다. 따라서 두 개의 그룹으로 나눠 IE 프로세스 30개를 내비게이션 직전 단계인 상태로 실행한다.

3.2 Execution Controller

웹사이트를 분석하기 위해 Execution Controller는 대상 URL을 이용하여 웹사이트에 내비게이션 하는 기능과 CPU Monitor 및 Web Loading Monitor와 연계하여 다른 그룹의 리소스 점유율이 줄어드는 시점 즉, 웹 로딩이 거의 완료되는 시점을 판단하는 기능으로 구성된다. i7급 PC 1대 내에서 IE 프로세스 30개를 동시 구동하는 것은

표 2. 시스템 경량화

구분	내용
프로세스 경량화	Aduiodg.exe, Dwm.exe, SearchFilterHost.exe, SearchIndexer.exe 등 8종 종료
윈도우 서비스 경량화 및 환경설정	오디오 서비스 중지 시각효과 최소화 파일 검색 중지 프린터 스풀러 중지 윈도우 라이센스 확인 중지 윈도우 업데이트 중지 불필요한 윈도우 기능 제거 하드 드라이브 조각 모음 예약 실행 중지 관리센터 메시지 중지 윈도우 방화벽 및 Defender 중지 시스템보호 및 시스템 오류 기록 중지 등

성능 및 크롤링 기능 측면에서 매우 비효율적이다. 따라서 그림 1과 같이 A, B 그룹을 나눠 상대 그룹의 리소스 점유가 줄어드는 시점에서 내비게이션을 교차 실행하여 거의 동시에 30개가 구동되는 효과를 구현한다.

3.3 Monitor

CPU Monitor는 시스템 전체 CPU를 점유율을 초 단위로 모니터링하여 Execution Controller에 전달하며, Web Loading Monitor는 IE 프로세스별 로딩 상태를 상(2초)·중(1초)·하(0초)로 판단하여 내비게이션 대기시간을 조정하여 안정적인 크롤링이 수행될 수 있도록 한다.

반복적으로 IE 프로세스가 실행되고 내비게이션 되면서 가비지 메모리 점유 증가 및 여유 메모리 부족 현상이 발생됨에 따라 크롤링 성능 및 효율성에 악영향이 미친다. 이를 방지하기 위해 Memory Monitor는 시스템 전체 메모리를 모니터링하여 주기적으로 가비지 메모리를 없애는 작업을 수행한다.

4. 실험 및 분석

4.1 실험 방법

실험은 크롤링 수행 성능 및 효율성을 측정하기 위해 국내·외 10만 개 웹사이트와 네이버를 1,000회 반복하여 접속하는 방법으로 실험하였으며, 신뢰성 있는 검증을 위해 시스템 경량화 및 가비지 메모리의 초기화 기능은 모두 적용하여 실험하였다.

국내·외 10만 개 웹사이트와 네이버를 1,000회 반복해서 접속하여 크롤링 수행 성능 및 효율성을 측정한 결과 거의 유사한 결과를 확인할 수 있었다. 하지만 웹사이트의 특성에 따라 상이한 성능 결과가 나타나는 문제를 해결하기 위해 네이버를 1,000회 반복하여 접속하는 방법으로 IE 프로세스 동시 수행하는 방식과 브라우저 리소스 점유 최적화 기능이 적용된 방식에 대해 크롤링 수행 성능 및 효율성을 검증하였다.

4.2 실험 결과 및 분석

본 논문에서는 네이버를 1,000회 반복해서 접속하여 크롤링 수행 성능 및 효율성을 측정하는 실험을 진행하여 제안하는 HDC가 효과적으로 웹사이트를 분석할 수 있음을 검증하였다. 브라우저 리소스 점유 최적화 기능이 적용되지 않은 방식으로 크롤링(네이버 1,000회 반복 접속)할 경우 IE 프로세스 16개를 동시 실행하였을 때 성공률 99.9%, 2분 25초의 소요시간을 보였고 프로세스 개수가 증가할수록 성공률은 떨어지며 소요시간은 늘어나는 것을 알 수 있었다. 브라우저 리소스 점유 최적화 기능이 적용될 경우 32~34개 IE 프로세스를 동시 실행하여도 표 3과 같이 소요시간과 성공률이 안정적이며 효율적임을 알 수 있다.

표 3. 크롤링 수행 성능 및 효율성 비교

구분	일반 (32개)	HDC (32개)	일반 (33개)	HDC (33개)	일반 (34개)	HDC (34개)
성공률	26.%	100%	10%	100%	10%	99.5%
소요 시간	12분	3분	14분	3분	15분	3분

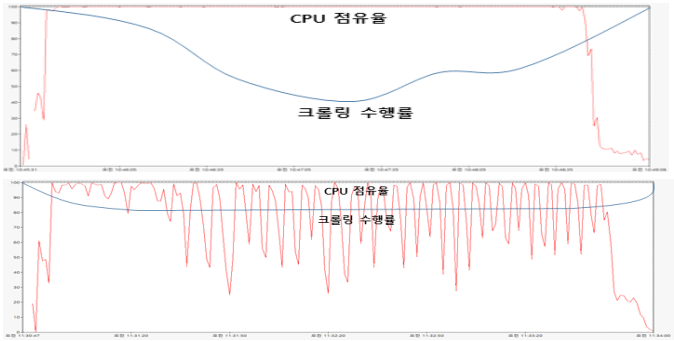

그림 2. 일반 크롤링(상), HDC 크롤링(하)

네이버를 1,000회 반복적으로 접속할 때 CPU 점유율 및 크롤링 수행률 분석 결과는 그림 2와 같다. 브라우저 리소스 점유 최적화 기능이 적용되지 않고 IE 프로세스 30개를 동시 실행하였을 때 CPU 점유율이 100%까지 증가하며 크롤링 수행률은 50% 이하고 떨어지는 것을 확인할 수 있다. 반면 브라우저 리소스 점유 최적화 기능이 적용된 HDC는 CPU 점유율이 간헐적으로 100%까지 증가하지만 지속적으로 유지되지 않는 것 확인할 수 있으며 그에 따라 크롤링 수행률이 80% 이상으로 유지되어 효율적인 크롤링 성능을 보증할 수 있음을 검증하였다. 또한 제안 HDC를 가상화 시스템으로 구현하기 위해서는 약 26배 시스템 구축비가 소요됨에 따라 경제적 효과도 검증하였다.

4. 결론 및 향후 연구

방대한 웹사이트에서 악성코드 유포 웹사이트를 효과적으로 탐지하기 위해 본 논문에서는 브라우저 리소스 점유 최적화 알고리즘이 적용된 HDC를 제안하였다. 제안 HDC은 크롤링의 성공률과 성능, 경제성에서 효율성을 입증하였고, 탐지 기능을 적용하면 악성 웹페이지 탐지 및 분석 도구로 될 수 있다. 향후 탐지 기능을 추가하여 악성 웹사이트의 탐지 연구를 지속할 계획이다.

[참고문헌]

[1] ENISA Threat Landscape Report 2016, https://www.enisa.europa.eu/publications/enisa-threat-landscape-report-2016, 2017, Feb. 8.

[2] Shindo, Yasutaka, et al. “Lightweight Approach to Detect Drive-by Download Attacks Based on File Type Transition.“ Proceedings of the 2014 CoNEXT on Student Workshop. ACM, 2014.

[3] 조호묵, 이경석 등, “리얼 브라우저 기반 웹 크롤러를 이용한 개선된 악성 웹사이트 탐지 기법”, 한국정보보호학회 하계학술대회 논문집, Vol. 26, No. 1, 2016, 6.

[4] Trupti V. Udapure, Ravindra D. Kale, Rajesh C. Dharmik, “Study of Web Crawler and its Different Types“, IOSR-JCE, e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 1, Ver. VI (Feb. 2014), PP 01-05