하이브리드 필터를 이용한 악성코드 탐지 장치

Malware Detection Device Using Hybrid Filter

저자
(Authors)

오동엽, 박재경
Dong-Yeob Oh, Jae-Kyung Park

출처
(Source)

한국컴퓨터정보학회 학술발표논문집 22(2), 2014.7, 67-70 (4 pages)
Proceedings of the Korean Society of Computer Information Conference 22(2), 2014.7, 67-70 (4 pages)

발행처
(Publisher)

한국컴퓨터정보학회
The Korean Society Of Computer And Information

URL

http://www.dbpia.co.kr/Article/NODE06603106

APA Style

오동엽, 박재경 (2014). 하이브리드 필터를 이용한 악성코드 탐지 장치. 한국컴퓨터정보학회 학술발표논문집 , 22(2), 67-70.

한국컴퓨터정보학회 하계학술대회 논문집 제22권 제2호 (2014. 7)

하이브리드 필터를 이용한 악성코드 탐지 장치

오동엽°, 박재경*
°*카이스트 사이버보안연구센터
e-mail:oh51dy@kaist.ac.kr°, wildcur@kaist.ac.kr*

Malware Detection Device Using Hybrid Filter

Dong-Yeob Oh°, Jae-Kyung Park*
°*Cyber Security Research Center, KAIST

● 요 약 ●

최근의 다양한 환경에서 악성코드나 의심 코드에 의한 피해가 날로 늘어나고 있는 추세이며 이를 종합적으로 대응할 수 있는 시스템에 대한 연구가 활발히 이루어지고 있는 상황이다. 이러한 악성코드는 사용자의 동의 없이도 PC에 설치되어 사용자가 인지하지 못하는 피해를 지속적으로 영산하고 있으며 그 심각성도 날로 심해지고 있는 실정이다. 또한 다양한 시스템으로부터 수집되는 방대한 양의 데이터를 실시간으로 처리하고 검증하는 기술 및 탐지 기법을 토대로 악성코드를 탐지하고 분석할 수 있는 대응기술로 고도화 되어야만 한다. 이러한 악성코드는 사용자의 PC에 설치되기 이전부터 검사 및 판단하여 사전 대응하는 것이 매우 중요하다. 본 논문에서는 이러한 악성코드가 실제 PC상에 설치되기 이전에 탐지할 수 있는 기법을 제시하며 이를 장치형태로 검증하였다. 본 논문에서 제시하는 기술을 토대로 악상코드 근절에 대한 근본적인 대안을 제시할 것이라 판단한다.

키워드: 악성코드(malware), 의심코드(suspicious code), 탐지(Detection), 검증(verification), 하이브리드 필터(Hybrid Filter)

I. 서 론

2013년 이후부터 보다 공격적이고 심각한 악성코드에 대한 피해는 날로 증가하는 추세이며 개인정보 유출 등의 문제와 연계되어 인터넷에 대한 불신이 가중되고 있다. 국내뿐만 아니라 미국 등의 선진국에서도 일어나는 현상으로 고객들의 정보 유출사건이 지속적으로 발생했다[1].

최근에 악성코드를 이용한 대표적인 공격 유형이 APT (Advanced Persistent Threat) 공격이며, 이 공격은 목표를 정하여 공격하는 방식으로 노골적인 공격을 감행하고 있다. 2013년 7월에 발표된 악성코드 유형별 비율을 보면 악성코드 유형 중 원격제어가 59%의 비율로 가장 높았다[2].

본 논문에는 악성코드 뿐만 아니라 사용자가 판단하여 피해를 줄 수 있는 코드는 모두 검사 대상에 포함하고자 한다. 즉, 내부자료 유출의 경우 악성코드는 아니나 사용자가 유출되는 것을 꺼려하는 것으로 볼 수 있으며 또한 파밍(Pharming)과 같이 인터넷 주소를 변조함으로써 사용자들로 하여금 진짜 사이트로 오인하여 접속하도록 유도한 뒤에 개인정보를 훔치는 범죄도 예방하는 방안을 강구하고자 한다.

본 논문의 연구를 바탕으로 설계된 프로토타입 시스템을 통해 실제 환경에서의 활용 가치와 성능적인 문제를 해결할 수 있는 방안을 함께 제안한다. 기존에 구성된 네트워크 환경에 패킷을 복사하는 형태로 구성하여 실제 트래픽을 실시간으로 처리할 수 있는 실험을 진행하였다.

II. 관련 연구

2013년 3.20 사고와 6.25 공격 사례를 살펴보면, 두 피해사례는 정부기관, 언론사, 금융기관들과 같이 사회 기반이 되는 산업들이 목표가 되었다는 점에서 공통점이 있다. 하지만 공격 기법 면에서는 서로 다른 형태를 보이고 있다. 3.20은 2011년과 그 이전에 발생했던 대규모 보안사고 사례와 유사하게 감염된 시스템의 정상적인 가동 방해 및 데이터 파괴를 위해 디스크의 MBR과 VBR을 특정 문자열로 강제 덮어쓰기를 수행하는 악성코드 유포가 목적이었다[3].

최근 국내 소프트웨어를 대상으로 한 취약점과 이를 악용한 악성코드가 증가하기 시작했고, 이러한 추세는 가속화하고 있다. 예를 들어 인터넷 뱅킹에 많이 쓰이는 소프트웨어의 취약점이 발견됐고(특히 Active-X), 이 외에도 곰플레이어와 같은 동영상 플레이어 프로그램에서 취약점이 보고되어 업데이트가 권고되기도 했다. 국내 문서작성 소프트웨어의 취약점을 이용한 경우도 발생하였는데 전 세계적으로 많이 사용하고 있는 문서관련 프로그램(워드 및 PDF)의 취약점을 이용한 공격도 꾸준히 발생하고 있으며

전자우편을 이용하여 자극적인 문구로 공격을 감행한다. 앞으로도 국내 소프트웨어를 대상으로 한 제로데이 취약점이 증가할 것으로 예상되는 만큼, 이를 사용하고 있는 기관이나 개인은 항상 최신의 보안패치를 확인하고 주의할 필요가 있다.

또한 드라이브 바이 다운로드 공격은 궁극적으로 사용자컴퓨터의 취약점을 이용하여 악성코드를 유포하는 방법이다. 공격자는 사용자 PC를 악성코드 유포사이트로 유도하기 위해 취약점을 가진 웹 서버에 SQL인젝션(SQL injection)과 같은 방법으로 침투한 후 웹 페이지에 직접 악성코드유포지로 연결 되는 코드를 삽입하거나, 광고 배너나 제3의 위젯과 연결되어 있는 링크를 변조하여 공격을 실행한다.

III. 본 론

본 논문에서 제안하고자 하는 하이브리드 필터를 이용한 의심코드 탐지 장치를 소개한다. 본 장치는 어플라이언스 타입의 장비를 통해 실시간으로 패킷을 처리하면서 의심코드를 분류하고 해당 코드가 악성코드인지를 파악하는 하이브리드 필터를 통해 탐지하는 과정을 수행한다.

3.1 하이브리드 필터 시스템 구성

본 논문에서 제안한 필터는 총 3가지의 종류로 커널 레벨의 필터와 크롤링 기반의 필터 그리고, 동적 실행 필터로 구성되어 있다. 각각의 레벨에서의 필터링을 통해 성능 지연을 최소화하여 실시간으로 처리할 수 있도록 설계하였다.

3.2 커널 URL 필터링

기존의 커널에서의 URL 필터링은 단순히 패턴 매칭에 의한 URL 필터링으로 해당 URL의 유무에 따라 추가 검사 여부를 결정하는 구조였으나 본 논문에서 제안하는 URL 필터링은 기존의 방식에 비해 검색 속도를 향상시켰다. 또한 웹 기반의 악성 코드에 감염되는 시나리오를 수집, 캐싱 하여 추후 분석 과정에서 재현 할 수 있는 시스템을 제안 하였다. 이러한 과정을 통해 향후 해당 URL을 크롤링하여 방문할 때 모든 페이지를 일일이 검사하는 불합리함을 개선하였다고 할 수 있다.

3.3 응용 실시간 필터링

기존 악성코드 탐지 연구 방법에서는 해당 웹 페이지를 다운로드하고 그 페이지 내에 숨김 iframe과 같은 비정상 태그의 존재 여부를 확인하여 처리하였다. 해당 방법은 스크립트 에뮬레이팅 모듈을 통해 페이지 내의 모든 URL을 추출하여 반복적으로 링크를 발견하지 못 할 때까지 반복하는 크롤링 기반의 탐지 방법이었다. 하지만, 이 방법의 경우는 실제 시스템을 적용하는데 있어서는 성능적인 한계가 있을 수 있다.

본 논문에서 사용하는 필터로는 다음과 같은 조건에 의해 실시간으로 해당 페이지만을 필터링하여 설계하였다.

- 시그니쳐 탐지
- 라인당 글자수 초과 탐지
- 비정상 문자열 포맷 스트링 탐지
- 숨김 iframe 탐지
- 숨김 applet 탐지, 숨김 object 탐지

이와 같은 두 가지의 필터링을 통해 페이지를 검사할 경우 해당 페이지가 악성코드를 내포하고 있는지 여부를 파악할 수 있지만 최근의 경우에는 현재 연구된 필터로도 탐지할 수 없는 유형들이 유포되고 있고 그 중 가장 보편화된 방법이 드라이브 바이 다운로드 기법이다. 이를 해결하기 위해 세 번째 필터를 제안하도록 한다.

3.4 PC 환경 파일 동적 검사

본 논문에서 제안하는 세 번째 필터로는 드라이브 바이 다운로드를 탐지하는 필터를 제안한다. 드라이브 바이 다운로드는 사용자의 개입 없이 웹 사이트를 방문만 해도 자동으로 파일이 로컬 PC에 설치되어 악성행위를 하는 방식으로 최근 들어 가장 많이 퍼져있는 공격 방법이라 할 수 있으나 마땅한 대응방안이 없는 공격이기도 하므로 본 논문에서는 실제 PC 환경을 도입하여 의심되는 사이트의 URL을 실제 브라우저로 접속하여 드라이브 바이 다운로드 공격이 이루어지는 지를 확인한다.

우선 실제 사용할 PC의 환경을 취약한 상태로 유지하는 것이 매우 중요한다. 취약점이 없는 PC의 경우에는 다운로드 되지 않는 형태도 있으므로 여러 환경을 결합하여 탐지한다. 그림 1에서와 같이 제안시스템을 통해 확정되지 않은 사이트가 별견되었을 경우 드라이브 바이 다운로드에 의한 공격인지를 확인하기 위해서 실제 수행을 진행한다.

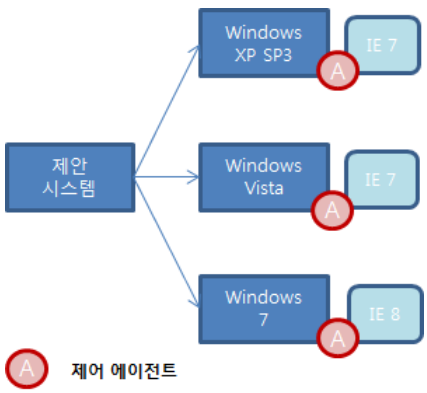

그림 106. PC 환경에서 에이전트를 이용한 Drive-by-Download 탐지
Fig. 1. Drive-by-Download detection on the PC environment using agent

이때 전체적인 프로세스 관리 및 파일이 다운로드 되는지를 모니터링 하기위해 에이전트를 개발하였다. 기존의 드라이브 바이 다운로드를 자동적으로 탐지하기 위해 오브젝트나 스트링을 트랙킹하거나 쉘코드의 스트링을 검사하는 방식을 통해 검출해 왔다. 그러나 본 논문에서는 스크립트를 통해 확인하는 것이 아니라 실

제 사용자의 개입 없이 파일이 다운로드 될 경우 이를 탐지하는 형태를 제안한다.

그림 2의 비정상적인 흐름은 어떠한 사용자의 개입이 없이도 프로그램이 다운로드 되는 것을 알 수 있으며 이를 정상적인 흐름과 비교하여 구별해 낼 수 있다. 정상적인 과정과 비교해보면 중간에 사용자에게 추가적인 파일을 설치하는 여부를 묻는 과정 자체가 생략되었고 초기에 응답을 통해 바로 실행가능 한 파일인 'hello.exe'를 요청하는 것을 알 수 있다.

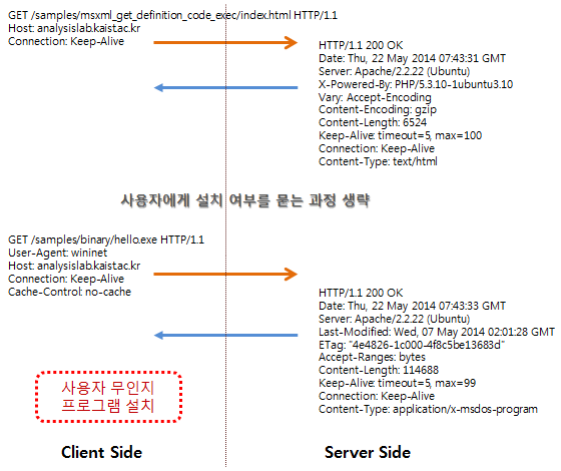

그림 107. 비정상적인 웹 통신 흐름
Fig. 2. Abnormal web traffic flow

본 논문은 위에서 제시한 3가지의 다중 필터링 시스템을 통하여 악성코드가 사용자 PC에 다운로드 되기 전에 이를 탐지하고 새로운 기법에 의한 방식으로 드라이브 바이 다운로드를 처리할 수 있는 방안을 제시하였다. 이러한 필터가 복합적으로 이루어질 경우 기존의 시스템의 처리 방식에 비해 매우 효과적이고 오탐이 없는 방안이라 할 수 있겠다.

Ⅳ. 실험 및 고찰

본 논문의 제안을 검증하기 위하여 본론에서 제시한 다중 필터를 프로토타입의 형태로 개발하여 실험하였다. 또한 드라이브 바이 다운로드의 지속적인 실험을 위한 실험 사이트를 직접 제작하여 운영하였다.

실험을 진행한 사이트 및 실험 운영 환경은 다음과 같다.

■실험 사이트

- http://analysislab.kaist.ac.kr/samples/ms13_037_svg_dashstyle/index.html
- 취약점 : CVE-2013-2551
- 테스트 환경 : Windows 7 / IE 8

■악성코드 파일 : calc.exe를 sample.exe로 변경 후 사용

위의 실험 환경을 통해 첫 번째 사이트를 방문하였을 경우 비정상적인 흐름을 감지하였고 이에 그림 3과 같은 결과를 출력 하였다.

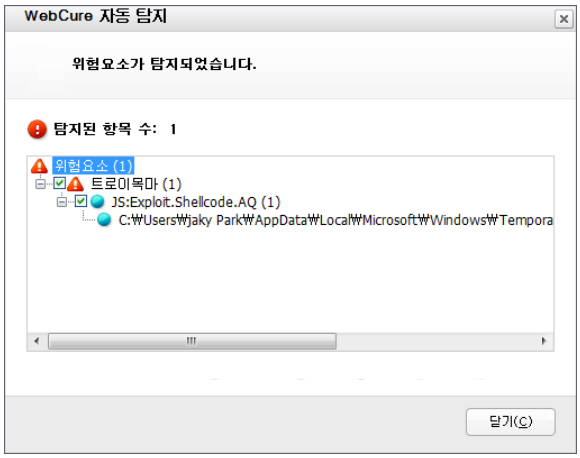

그림 108. 드라이브 바이 다운로드 자동 탐지
Fig. 3. Auto detection of drive-by-download

본 논문에서 제시한 다중 필터는 우선적으로 기존에 발견된 악성코드 URL을 커널 레벨에서 고속으로 탐지하고 또한 응용 레벨에서 크롤링을 통하여 탐지함으로써 패턴이 없는 사이트도 실시간으로 탐지할 수 있는 방안을 제시하였다. 마지막으로 드라이브 바이 다운로드를 근본적으로 차단함으로 인해 악성코드가 사용자의 허가 없이 직접 다운되는 것을 막는 방안을 제시하였다. 이러한 과정을 통해 악성코드 자체가 어떠한 행위를 하는지를 알기 전에 악성코드를 숨겨놓은 사이트를 찾아서 막음으로 인해 보다 효과적인 방어 방안을 제시하였다고 볼 수 있다.

Ⅴ. 결 론

본 논문에서는 기존의 악성코드를 직접 분석하거나 패턴에 의해 탐지하는 것이 아니라 악성코드를 웹 사이트에 숨겨 놓거나 해킹을 통해 사용자가 인지하지 못하는 방식으로 배포하는 것을 원천적으로 차단할 수 있는 방안을 제시하였다. 본 논문에서는 커널 레벨에서의 기존 URL을 차단하는 필터와 실시간으로 방문할 시점의 사이트를 크롤링 기법에 의해 추적하는 필터와 드라이브 바이 다운로드에 의한 악성코드 다운을 탐지할 수 있는 기법을 다중 필터 형태로 제안하였다. 이에 대한 실험을 위해 실제 드라이브 바이 다운로드 사이트를 제작하였고 객관적인 검증을 위해 VirusTotal과 같은 글로벌 사이트에서 추가 검증을 진행하였다. 본 제안된 기법을 통해 실제 네트워크에서 활용할 수 있는 시스템을 제작할 경우 매우 효과적인 기법이라고 할 수 있다. 본 연구를 통해 기업과 기관 및 개인에 적용할 수 있는 악성코드 탐지 기술을 보급함으로써 알려지지 않은 악성코드에 대한 예방 및 방어를 할 수 있을 것으로 기대한다.

참고문헌

[1] *http://www.ahnlab.com/kr/site/securityinfo/secunews /secuNewsView.do?menu_dist=2&seq=22325*

[2] Provos, N., McNamee, D., Mavrommatis, P., Wang, K., and Modadugu, N. *"The ghost in the browser analysis of web-based malware,"* Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets, pp. 4-4, Apr. 2007.

[3] Chen, K.Z., Gu, G., Zhuge, J., Nazario, J., and Han, X.., *"WebPatrol: Automated collection and replay of web-based malware scenarios,"* Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, pp.186-195, Mar. 2011.

[4] SpiderMonkey, *"https://developer.mozilla. org/en-US/ docs/Mozilla/Projects/SpiderMonkey"*

[5] Manuel Egele, Peter Wurzinger, Christopher Kruegel, and Engin Kirda, Defending Browsers against Drive-by Downloads: Mitigating Heap-spraying Code Injection Attacks, ACM New York, pp. 281-290, 2010.

[6] *https://www.VirusTotal.com*