韓國컴퓨터情報學會 論文誌
第14卷 第9號, 2009. 9.

2009-14-9-3-3

HTTP Outbound Traffic을 이용한 개선된 악성코드 탐지 기법

최 병 하*, 조 경 산**

An Improved Detecting Scheme of Malicious Codes using HTTP Outbound Traffic

ByungHa Choi *, Kyungsan Cho **

요 약

웹을 통해 유포되는 악성코드는 다양한 해킹 기법과 혼합되어 진화되고 있지만, 이의 탐지 기법은 해킹 기술의 발전과 신종 악성코드에 제대로 대응하지 못하고 있는 실정이다. 본 논문에서는 악성코드와 이의 유포 특성의 분석에 따라 탐지 시스템이 갖추어야 할 요구 사항을 정의하고, 이를 기반으로 HTTP Outbound Traffic을 감시하여 악성코드의 유포를 실시간으로 탐지하는 개선된 탐지 기법을 제안한다. 제안 기법에서는 악성코드를 유포하는 것으로 입증된 HTML 태그와 자바스크립트 코드를 시그니쳐로 IDS에 설정한다. 실제 침입된 환경에서의 검증 분석을 통해 제안 기법이 기존 기법에 비해 요구 사항의 만족에 우수하고 악성코드에 대한 높은 탐지율을 보임을 제시한다.

Abstract

Malicious codes, which are spread through WWW, are now evolved with various hacking technologies. However, detecting technologies for them are seemingly not able to keep up with the improvement of hacking and newly generated malicious codes. In this paper, we define the requirements of detecting systems based on the analysis of malicious codes and their spreading characteristics, and propose an improved detection scheme which monitors HTTP Outbound traffic and detects spreading malicious codes in real time. Our proposed scheme sets up signatures in IDS with confirmed HTML tags and Java scripts which spread malicious codes. Through the verification analysis under the real-attacked environment, we show that our scheme is superior to the existing schemes in satisfying the defined requirements and has a higher detection rate for malicious codes.

• 제1저자 : 최병하 교신저자 : 조경산
• 투고일 : 2009. 08. 10, 심사일 : 2009. 08. 14, 게재확정일 : 2009. 09. 04.
* 단국대학교 정보통신대학원 ** 단국대학교 컴퓨터학부 교수
※ 이 연구는 단국대학교 2009학년도 대학연구비 지원으로 연구되었음.

▸ Keyword : 악성코드(malicious code), Outbound Traffic, 탐지(detection), 시그너처(signature), HTML 태그(HTML tag)

Ⅰ. 서 론

컴퓨터 또는 네트워크의 취약점을 찾아내어 불법적인 목적으로 이용되는 해킹은 OS 해킹, 원격 사이트의 취약한 데몬 해킹, TCP/IP 취약점 해킹의 단계를 거쳐 현재는 웹 응용 프로그램 해킹으로 발달하였다[1]. 웹 응용 프로그램 해킹은 웹의 대중화, 정보의 다양성, 그리고 용이한 접근성으로 인해 XSS(Cross site scripting), SQL(Structured Query Language) Injection, WEB SHELL 등의 기법을 단독 또는 혼합으로 적용하여 다양한 피해가 발생되고 있다.

한국정보보호진흥원에 의하면 2009년 1월에 발생한 해킹 피해는 메일 서버를 해킹하여 스팸메일을 보내는 39.1%의 스팸릴레이에 이어 홈페이지 변조가 27%로 2위를 차지하고 있으며, 홈페이지 변조는 전월에 비하여 140%의 높은 증가추세를 보이고 있다[2]. 홈페이지 변조는 악성코드 유포와 개인정보 유출 등으로 2차 피해를 일으키는데 그 중 악성코드 유포는 2008년 12월 6만9964개에서 2009년 1월과 2월에는 각각 13만8505개, 19만 2433개로 급속히 증가하고 있다[3].

신종 악성코드와 새로운 해킹 기법들은 계속 개발되어, 기존 기법과 장비로 악성코드를 방어하거나 탐지하기 어렵다. 또한 침입 후에 해커는 웹서버에서 침입 흔적과 이상 징후를 삭제하여 서버 관리자가 인지 못하게 한 후 다량의 악성코드를 유포하게 한다.

본 논문은 이러한 문제점을 해결하기 위해 HTTP Outbound Traffic을 감시하여 사용자 컴퓨터가 악성코드에 감염되기 이전에 악성코드의 유포를 실시간으로 탐지하는 기법을 제안한다.

해커가 악성코드를 유포하기 위해 웹서버의 홈페이지를 변조했다면, 웹서버는 HTML(Hyper Text Markup Language) 문서에 악성코드를 유포시킬 수 있는 HTML 태그와 자바스크립트 코드 등의 특정 문자열을 포함하여 응답하는 것으로 분석된다. 이러한 특성을 기반으로 서버측의 IDS(Intrusion Detection System)에서 악성코드의 유포 시에 사용되는 특정 문자열을 포괄적인 시그너처로 설정하면, 예측 가능한 서버의 한정된 HTML 문서를 감시하며 악성코드 유포를 탐지할 수 있다.

신종 악성코드도 유포시에 동일한 패턴의 특정 문자열이 적용된다는 분석에 기반한 제안 기법은 신종 악성코드와 새로운 해킹 기법에 대응책이 될 수 있다. 또한 악성코드의 유포를 실시간으로 탐지할 수 있고, 숙주서버를 신속히 추적하여 악성코드의 확산을 막을 수 있다.

본 논문의 구성은 다음과 같다. 2장에서 악성코드의 정의 및 악성코드의 유포 기법 및 탐지 기법을 분석하고, 악성코드 탐지 시스템의 요구 사항을 제시한다. 3장에서는 HTTP Outbound Traffic으로 악성코드를 탐지하는 개선된 기법을 제안한다. 그리고 4장에서는 제안 기법을 검증 분석하고 5장의 결론으로 본 논문을 마무리 짓는다.

Ⅱ. 악성코드 유포 기법과 탐지 기법의 분석

1. 악성코드 정의와 유형

"malicious code" 또는 "malware" 등으로 불리는 악성코드는 "운영체제 커널이나 보안에 민감한 애플리케이션의 동작을 변화시키는 코드 부분으로 이는 사용자의 동의 없이 이루어지거나 운영체제나 애플리케이션의 문서화된 기능(예:API)을 사용하여 그러한 변화를 탐지할 수 없도록 행해진다."라고 정의된다[4].

악성코드는 기능에 따라 유용한 프로그램으로 가장하여 불법적으로 정보나 파일을 외부로 전송하는 트로이목마(Trojan), 다른 파일에 삽입되어 자기복제를 하는 바이러스, 독립적으로 자기복제를 하기 위해 네트워크 스캐닝을 일으키며 CodRed나 Slammer의 원인이 된 웜(worm) 등으로 나눌 수 있다[5].

이들 악성코드는 해킹과 결합되어 발전하는 추세로, 악성코드가 자체적으로 증식하는 것뿐만 아니라, 해킹 기법을 이용하여 웹을 통해 유포되기도 하고, 악성코드에 해킹 기법이 삽입되어 DDos 같은 공격을 발생시키기도 한다.

2. 홈페이지 변조를 통한 악성코드 유포 기법

본 절은 악성코드의 유포기능을 가진 HTML 태그와 자바스크립트 코드를 분석하고 이들 태그와 코드를 불법적으로 입력하는 해킹 기법을 제시한다.

2.1 악성코드의 유포 패턴 분석

홈페이지를 통해 악성코드를 유포하는 HTML 태그와 자

바스크립트 코드는 다음과 같은 종류가 많이 사용되는 것으로 분석된다.

1) IFRAME 태그는 브라우저 화면의 일부분을 다른 문서로 할당할 때 쓴다. 브라우저의 화면에 보이지 않으면서 외부문서를 가져올 때 악성코드를 유포한다[6].

2) EMBED 태그와 OBJECT 태그는 FLASH 파일이나 동영상 파일 등 브라우저에서 실행 가능한 Active-X 등의 이진 파일을 브라우저에 실행할 때 사용하는 태그다. 이들 태그로 악성코드를 유포할 때는 IFRAME처럼 화면에 보이지 않으면서 외부 파일을 참조한다[7].

3) LINK 태그는 웹 사이트의 스타일쉬트(Stylesheet)를 정의하는 태그로써 하나의 파일로 여러 웹 페이지에서 참조하는 형식으로 대부분 사용한다. 이를 이용하여 다른 외부의 도메인이나 IP에 있는 스타일쉬트 파일을 참조케 하여 악성코드를 유포한다[8].

4) SCRIPT 태그는 자바스크립트를 정의하는 태그로써 자바스크립트 함수들을 독립된 파일로 만들어 SRC 속성으로 호출할 수 있다. 이를 이용하여 해커들은 SRC 속성이 외부 도메인의 파일을 참조케 하여 악성코드를 유포한다. 또한 LANGUAGE 속성에 "JScript.encode"를 넣고 자바스크립트 코드를 암호화하여 무슨 의미인지 알 수 없게 관리자를 속여 악성코드를 유포할 수도 있다[9].

5) 자바스크립트 중 decodeURIComponent(), decodeURI(), charCodeAt(), unescape(), String.fromCharCode() 함수는 특정 코드 문자를 ASCII 코드 문자로 변환해주는 함수로써 서버 관리자나 사용자가 직관적으로 무슨 내용인지 알 수 없는 내용을 ASCII 코드로 변환해서 실행한다. 즉, 앞서 제시한 HTML 태그를 다른 코드로 변환된 상태에서 실행한다[10].

2.2 SQL Injection을 이용한 악성코드 유포

홈페이지에서 SQL로 해석될 수 있는 입력을 시도하여 데이터베이스에 접근하도록 하는 기법이 SQL Injection이다. 악성코드를 유포하는 2.1의 코드와 태그를 아래와 같은 방법으로 DB에 입력한다.

첫째, HTTP Cookie 또는 REFERER라는 HTTP 헤더 인자를 이용하는 것으로, 이들 속에 악성코드를 유포하는 태그와 코드를 SQL문과 조합하여 데이터베이스와 연동 후 DB에 입력하는 해킹 방식이다[11].

둘째, 브라우저의 주소창에 입력하는 URL의 일부분을 SQL문으로 입력하는 기법이다. 입력요소가 데이터베이스와 연동되는 부분이라면 SQL문으로 실행될 수 있는데 이때 유포 태그와 코드를 DB에 입력한다[8].

셋째, 최근 중국에서 클릭 몇 번으로 초보자도 SQL Injection을 쉽게 할 수 있는 툴인 HDSI, D-SQL, NBSI, WIS 등이 제작되었다. 이를 이용하여 데이터베이스의 내용을 조작하여 유포한다[8].

2.3 웹쉘을 통한 악성코드 유포

웹쉘(WEB SHELL)은 해커들이 해킹된 서버에서 사용할 웹 프로그램으로 이를 이용하면 웹서버의 대부분 자원을 통제할 수 있다[12].

가능한 침투 기법들은 다음과 같다.

1) 파일 업로드가 되는 게시판을 통해 침투한다.

2) SQL Injection으로 파일 생성 권한을 DB를 통해 획득한 후 웹서버에 웹쉘을 생성한다.

3) "ASP목마2006"과 같은 생성기로 쉽게 침투할 수 있는 웹쉘을 작성하여 해킹한다.

웹쉘을 통한 침투가 성공하면, 파일의 조작 기능을 이용해 홈페이지를 변조하여 악성코드를 유포한다.

3. 기존 악성코드 탐지 기법 분석

앞 절에서 제시한 문제의 해결을 위해 침입 탐지 및 대응을 위한 통합 보안 관리 시스템의 연구와 악성코드 탐지 기법들이 제안되었다[5, 13-17]. 대표적인 악성코드 탐지 기법을 비교 분석한 결과는 표 1과 같다. 이들 기법들은 악성코드 자체를 탐지하거나 이상 행위 또는 실행 후 결과를 탐지하는데, 각각 다음과 같은 제약을 가진다.

1) 신종 악성코드를 탐지 못한다.

2) 탐지를 위해 시스템 자원을 너무 많이 소모한다.

3) 오탐율이 상당히 존재한다.

그러므로 기존의 탐지 기법의 제약점을 해결하는 개선된 탐지 기법이 요구된다.

4. 악성코드 탐지 시스템의 요구 사항

앞 절에서 분석한 최근의 악성코드의 증가추세는 컴퓨터와 네트워크 안에서 활동하던 악성코드가 해킹 기법과 결합되어 홈페이지 변조 등의 기법으로 웹을 통해 다량의 악성코드를 장기간 유포하도록 진화하고 있다. 하지만, 3 절에서 제시한 바와 같이 기존의 탐지 기법들은 한계가 있어 2 절의 다양한 해킹 기법을 통한 악성코드 유포는 탐지하기가 어렵다.

이러한 분석을 토대로, 본 논문에서는 기존의 문제점을 해결하고 새로운 탐지 시스템의 필요를 만족하기 위해 표 2와 같은 요구 사항을 제시한다.

표 1. 악성코드 탐지 기법 비교
Table 1. Comparison of the Existing Detecting Schemes

기법	내용	탐지대상	비고
시그너처기반의 탐지 기법	악성코드로 판단할 수 있는 특정 문자열을 시그너쳐로 데이터베이스화 하여 해당 파일을 검사하는 기법으로 대부분은 안티 바이러스 소프트웨어(백신)의 기법	악성코드 파일	빠르고 정확하게 진단할 수 있으나 신종 악성코드는 탐지하지 못한다.
휴리스틱 검사법	시그너쳐기반 탐지의 한계를 넘기 위한 기법으로 악성코드가 실행하는 명령이나 행위를 찾으므로 변종의 악성코드도 탐지 가능	악성코드의 특별한 명령과 행위	변종 악성코드까지 탐지 가능하나 오탐이 가능하다.
가상환경을 이용한 기법	가상환경에서 파일 감염, 파일 삭제, IRC 서버 연결, 이메일 송신 및 리스닝 포트 오픈까지 다양하게 실행하며 탐지하는 기술	가상환경에서 실행하고 수집된 결과를 통해 악성코드 탐지	감염 피해가 없고 알려지지 않은 악성코드까지 탐지 가능하나, 검사할 때 마다 시스템 자원의 소모가 많다.
전자메일 콘텐츠 필터링 방식	전자메일의 제목(Subject), 본문(Body) 그리고 첨부파일(Attachment File) 명칭과 같이 특징적인 문구로 탐지하는 기법	매스 메일러 웜(Mass Mailer Worm)	탐지 기법이 쉽고 적용하기가 편리하나 전자메일의 형태가 변경되면 탐지 못 한다
행동기반 악성코드 탐지	시스템내의 행위를 탐지하는 시스템 기반 탐지 기법과 네트워크의 행위를 탐지하는 네트워크 기반 탐지 기법으로 나뉘며 비정상 행동을 하는 악성코드를 탐지	악성코드의 이상 행위	알려지지 않은 악성코드까지 탐지하나 오탐이 많다.

표 2. 악성코드 탐지 시스템의 요구 사항
Table 2. Requirements for the detection systems of malicious codes

요구사항	**웹서버의 해킹으로 컴퓨터에 유포되는 악성코드 탐지**
	컴퓨터가 악성코드에 감염되기 이전에 탐지 이미 컴퓨터에 감염된 악성코드는 2009년 7월의 DDos 공격으로 나타난 것처럼 피해는 심각하다. 이들은 감염되기전에 차단할 필요가 있다.
	실시간 탐지 악성코드가 유포되는 상황을 실시간으로 탐지 가능해야한다.
	새로운 악성코드 탐지 매일 새롭게 출현하는 악성코드는 기존의 anti-virus 제품으로는 탐지하기 어렵다. 이러한 새로운 악성코드를 실시간으로 탐지하는 기법이 필요하다.
	탐지범위의 제한 과도하게 모든 트래픽을 검사하여 네트워크에 체증(congestion)을 유발하지 않고 일부의 트래픽(HTTP)만 검사하여도 악성코드가 탐지 가능해야한다.

III. HTTP Outbound Traffic을 이용한 악성코드 탐지 기법 제안

본 장에서는 앞 장의 4 절에서 제시된 요구 사항을 만족시키기 위해 HTTP Outbound Traffic을 감시하여 악성코드를 유포하는 HTML 태그와 자바스크립트 코드를 실시간으로 탐지하는 개선된 탐지 기법을 제안한다.

1. 악성코드 유포와 탐지 위치

일반적인 악성코드의 유포 과정은 그림 1과 같다.

Ⓐ 해커는 악성코드 숙주서버를 먼저 구성한다.
Ⓑ 숙주서버에서 악성코드가 유포되도록 유명 웹 사이트를 해킹한다.
Ⓒ 컴퓨터가 유명 웹 사이트에 접속한다.
Ⓓ 해킹된 웹 사이트는 응답을 한다.
Ⓔ 응답된 HTML 문서 속에 악성코드 숙주서버에서 악성코드를 다운로드해서 감염된다.

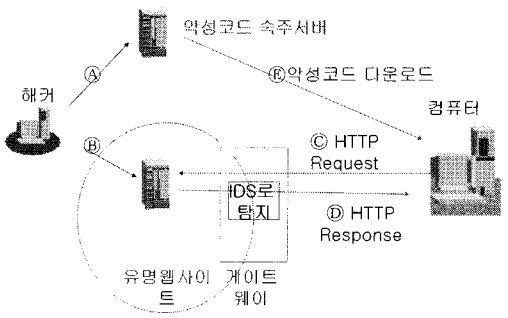

그림 1. IDS의 위치
Fig 1. Location of IDS

본 연구에서 제안하는 악성코드의 탐지는 Ⓓ의 단계에서 수행되며, 그림 1의 IDS와 같은 위치에서 탐지하여 서버 쪽의 HTTP Outbound Traffic의 패킷 속에 악성코드를 유포하는 코드가 존재하는지 확인한다.

2. 악성코드 탐지 기법에 이용하는 시그너쳐

앞 장의 2.1에서 악성코드를 유포하는 HTML 태그와 자바스크립트 코드를 분석하였다. 분석 결과를 기반으로 악성코드를 유포할 수 있는 대표적인 시그너쳐를 표 3과 표 4와 같이 제안한다.

표 3에 대한 의미는 HTML 문서에 〈LINK HREF="http://다른도메인/css.css"〉라는 문자열이 있다면, LINK 태그의 HREF가 (1)에 해당하므로 시그너쳐에 탐지된다는 것이다.

3. 시그너쳐의 IDS 적용

제안 시스템에서는 앞 절에서 제안한 시그너쳐를 Snort(version 2.8.3.2) IDS에 다음과 같이 적용한다. Snort는 침입 탐지를 위해 시그너쳐와 규칙을 가진다. 그 예로, 그림 2는 포트 80으로 송수신되는 TCP 세그먼트를 감시하며 시그너쳐인 "iframe"이라는 문자열을 탐지하라는 규칙이다. 또한 pcre:정규표현식이라는 형태로 정규표현식을 추가할 수도 있다. 이와 같이, Snort에서 2 절의 시그너쳐들을 규칙모음으로 설정한다.

```
alert TCP any any <> any 80 (msg:"iframe attack";
flow:established; content : "iframe"; nocase; sid:5555)
```

그림 2. 규칙과 시그너쳐
Fig. 2. A Rule and A Signature

4. 제안 탐지 시스템의 동작

본 절에서 제안한 탐지 시스템은 그림 3과 같이 동작하며, 수행단계는 다음과 같다.

1) 해킹당한 서버에서 전송되는 HTTP Outbound Traffic의 패킷을 IDS가 캡쳐한다.
2) 앞 절에서 제시된 시그너쳐들로 구성된 규칙모음을 근거로, Snort로 이루어진 IDS 엔진이 패킷들을 하나씩 살펴본다.
3) IDS 엔진이 패킷들을 비교하고, 분석하여 시그너쳐에 해당하는 문자열을 가진 패킷을 만나면 경고를 일으킨다.
4) 웹사이트 특성에 따라 TCP 연결을 차단하거나 관리자에게 통계 자료 등을 제공한다.

표 3. HTML 태그의 시그너
Table 3. Signatures of HTML tags

태그명	속성명	속성값
IFRAME	HEIGHT	0
	FRAMEBORDER	0
	SRC	(1)에 해당
IFRAME	WIDTH	0
	SRC	(1)에 해당
IFRAME	STYLE	display:none 이 포함
	SRC	(1)에 해당
IFRAME	STYLE	visibility:hidden 이 포함
	SRC	(1)에 해당
LINK	HREF	(1)에 해당
EMBED	HEIGHT, WIDTH	0
	SRC	(1)
OBJECT	HEIGHT, WIDTH	0, (2)
OBJECT	STYLE	display:none 포함, (2)
SCRIPT	SRC	(1)에 해당
SCRIPT	LANGUAGE	JScript.encode
	SRC	(1)에 해당
SCRIPT	LANGUAGE	JScript.encode
	SRC	해당 속성이 없음

(1) "./", "/", "문서명", "http://서버의 아이피 또는 도메인"으로 시작하지 않음
(2) src 속성에 해당하는 속성이 다른 이름으로 쓰일 수 있으므로 상황에 따라 예외처리 해야 함

표 4. 자바스크립트 코드의 시그너쳐
Table 4. Signatures of javascript codes

시그너쳐	내용
String.fromCharCode(	이 함수들은 표 4에 HTML 태그를 다른 문자코드에서 ASCII 코드로 변환하므로 직관적으로 악성코드를 유포하는 태그를 찾기 힘들게 한다. 흔치 않지만 만약 정상적인 unescape("%31")라는 부분이 있다면 정규표현식으로 예외 처리하고 나머지 "unescape("를 탐지하게 한다.
decodeURIComponent(	
unescape(	
decodeURI(	
charCodeAt(	
document.write("〈OBJ..... .width="0" (.....은 임의 문자열)	OBJECT 태그를 자바스크립트로 변환한 것으로 IDS가 탐지 못하게 할 목적으로 사용한다.

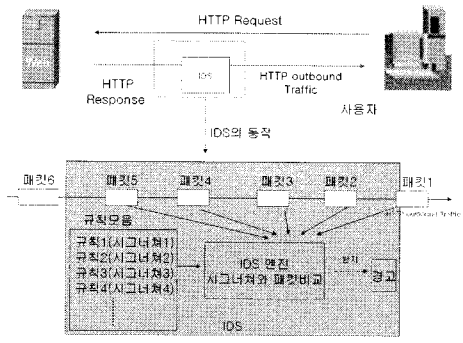

그림 3. 제안 탐지 시스템의 동작
Fig 3. Operation of the Proposed Detecting System

5. 본 연구의 요구 사항과 제안 탐지 시스템

표 5. 제안 탐지 시스템의 기능
Table 5. Functions of the Proposed Detecting System

항목	제안 탐지 시스템의 기능
웹서버의 해킹으로 컴퓨터에 유포되는 악성코드 탐지	악성코드를 유포하는 HTML 태그와 자바스크립트코드로 탐지 가능
컴퓨터가 악성코드에 감염되기 이전에 탐지	서버에서 유포되어 컴퓨터에 감염되기 전에 탐지 가능
실시간 탐지	가능
탐지범위의 제한	HTTP 탐지만으로 가능
새로운 악성코드 탐지	신종 악성코드도 기존의 태그와 코드를 사용함으로 탐지 가능

표 6. 부가적인 기능
Table 6. Additional Fuctions

악성코드 숙주서버의 위치를 파악 유포하는 태그와 코드를 파악하여 숙주서버까지 추적가능
악성코드 유포에 관련된 서버의 해킹 탐지 서버 관리자는 악성코드와 관련된 해킹 사실을 인지 못하는 경우가 많은데 제안 기법으로 탐지 가능
악성코드 유포에 관련된 새로운 해킹 기법에 대한 대응 새로운 기법으로 해킹하더라도 유포하는 태그와 코드는 동일하므로 그에 대한 대응 기법이 될 수 있음
오탐율(False Positive) 최소화 정해진 서버의 한정된 HTML 문서를 탐지하므로 오탐을 예측할 수 있으며, 오탐 발생시 상황에 따라 시그너쳐를 정규표현식으로 오탐 부분만 예외처리 가능

실제 IDS에 적용해 검증한 결과, 제안 탐지 시스템은 앞서 2장에서 제시한 요구 사항을 표 5와 같이 만족시킨다. 또한 제안 시스템은 표 6과 같은 부가적인 기능이 있다.

IV. 제안 탐지 기법의 검증 분석

1. 제안 탐지 시스템 구성

제안 탐지 시스템은 그림 4와 같이 구성되며 PC에 악성코드가 발현해도 시스템 전체의 파손을 막기 위해 MS의 Virtual PC를 사용하여 "가상 서버 시스템"을 윈도우 2000으로 설치한다. 그 시스템 안에 "탐지 대상 시스템"과 "제안 탐지 시스템"을 구성한다. "탐지 대상 시스템"은 실제 해킹된 사이트를 찾아 그와 동일한 환경과 내용으로 DB와 웹서버, 그리고 웹프로그램 소스를 설치했다. 그리고 "제안 탐지 시스템"은 공개 소스 IDS인 Snort를 사용하여 유출되는 패킷을 분석한 후 본 연구의 시그너쳐에 따른 유포패턴을 확인했을 경우 경고를 일으키게 한다. 또한 패킷 캡쳐 기능이 있는 winpcap 라이브러리를 설치하여 캡쳐된 패킷을 Snort가 분석할 수 있도록 구성한다. 클라이언트 PC는 가상서버시스템의 외부에 설치한다. 표 7은 각 시스템이 구성하는 자세한 구성 요소들이다.

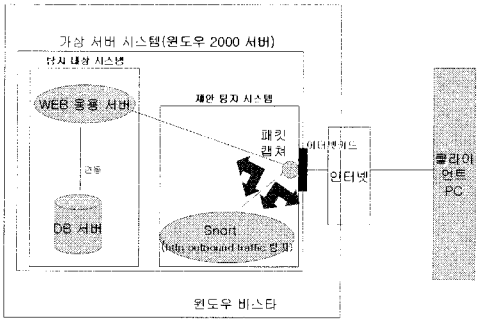

그림 4. 전체 시스템 구성도
Fig. 4. System Architecture

표 7 전체 시스템의 구성 요소
Table 7. Components of System

서버	항목	내용
가상 서버 시스템	운영체제	윈도우 2000 서버(MS Virtual PC 기반)
	domain	test.com
	ip	192.168.0.6
탐지 대상 시스템	DB	MS - SQL 2000
	웹서버	IIS 6.0
	웹프로그램 언어	ASP
	해킹 상태	SQL Injection 침해
제안 탐지 시스템	IDS	Snort(version 2.8.3.2)
	Packet Capture Library	winpcap (version 4.0.2)

2. 탐지율 분석

제안 탐지 시스템의 탐지율을 분석하기 위한 탐지 대상 시스템의 실제 해킹 현황은 HDSI라는 툴로 SQL Injection으로 침해되어 DB의 "Board"라는 테이블에 8176개 행(ROWs)들이 악성코드를 유포하는 "〈script src=http://s.lbs66.cn/kr.js〉〈/script〉" 자바스크립트로 해킹된 상태다. 50개의 웹페이지의 해킹 당한 HTTP Outbound Traffic에 대해 탐지율을 분석한 결과 표 8과 같이 100%의 탐지율을 보였다. 탐지율이 100%인 원인은 8176개의 동일한 형태, 즉 모두 〈SCRIPT SRC="외부도메인"〉이라는 시그너쳐의 형태로 침해되었기 때문이다.

표 8. 악성코드 탐지율
Table 8 Detection rate

해킹당한 웹페이지 수	false positive 갯수	false negative 갯수	탐지율
50	0	0	100%

또한 더 다양한 형태의 탐지율을 분석하기 위해 인터넷에 악성코드를 유포하는 것으로 알려져 있는 웹페이지의 HTML 소스를 복사하여 탐지 대상 시스템에 동일하게 생성시킨 후 이를 탐지하는 기법으로 추가적인 테스트를 하였다. 표 9는 악성코드를 탐지하는 백신과의 탐지율을 비교한 결과이다. 비교대상은 네이버 PC그린의 카스피스키 엔진으로 33%를 탐지하였고 제안 탐지 시스템은 91.6%의 우수한 탐지율을 보인다. 백신 탐지율이 낮은 원인의 일부는 신종 악성코드를 탐지하지 못하기 때문이다. 탐지 못한 8건 중 3건은 신종 악성코드로 각각 일정시간 이후 백신에 탐지 되였다. 나머지 5건은 악성코드 숙주서버가 곧바로 치료되었으므로 신종 악성코드인지 확인하지 못하였다. 반면에 제안 기법은 악성코드 자체가 아닌 유포를 탐지하므로 신종 악성코드도 탐지되었다. 제안 탐지 시스템이 탐지 못한 1건은 IFRAME 태그의 SRC 속성값이 외부가 아닌 자기자신의 도메인을 가르키며 악성코드가 유포가 되는 것으로, 그 서버 자신이 악성코드 숙주서버인 것으로 추정되며, 현재 가리키는 웹페이지 주소가 치료되어 확실한 원인은 분석할 수는 없다.

표 9. 탐지율 비교
Table 9. Comparison of detection rate

악성코드 유포페이지갯수	제안 탐지 시스템 탐지 갯수(율)	백신 탐지 갯수(율)
12	11(91.6%)	4(33%)

표 10은 표 9에서 제안 팀지 시스템이 탐지한 악성코드의 시그너쳐를 분석한 결과이다. 또한 트래픽 양에 대한 탐지율의 변화는 없는 것으로 분석되었다.

표 10. 악성코드를 탐지한 시그너쳐
Table 10. Signatures of Malicious codes

항목	갯수
String.fromCharCode	4 건
IFRAME	2 건
document.write("〈OBJ...width="0"	1 건
〈script src="외부도메인"	3 건
unescape	1 건

V. 결 론

본 연구에서는 홈페이지를 변조해서 유포하는 악성코드에 대한 분석 결과를 근거로 대응 방안에 대한 요구 사항을 제시하고, 이를 만족시키는 효율적인 악성코드 탐지 기법을 제안하였다.

제안 기법은 악성코드 자체의 탐지가 아닌 Outbound Traffic을 통한 악성코드 유포의 탐지로서 기존 기법들과는 여러 차이점이 존재한다. 웹서버에서 해킹되어 유포되고 있는 악성코드를 실시간으로 쉽게 발견하고 컴퓨터에 감염되기 전에 예방조치로도 활용가능하며 악성코드 숙주서버도 신속히 발견할 수 있다. 또한 해커들이 새로운 악성코드 해킹 기법을 만들거나 신종 악성코드를 제작하더라도 악성코드를 유포하는 태그와 코드는 대부분 동일하므로 새로운 기법에 대한 대응도 될 수 있다.

제안 기법은 특정한 웹서버를 통한 악성코드의 유포를 탐지하는 기법이므로 광범위한 영역에서보다는 한정된 인트라넷에 더 효율적이며, 향후 악성코드의 새로운 유포 기술의 등장에 대한 대응이 요구된다. 또한 본 연구에서는 현재 구현할 수 있는 제한된 환경에서의 분석만을 제시했는데 연구의 체계적인 검증을 위해서는 더욱 다양한 유형의 악성코드 유포

환경을 구축하여 검사할 필요가 있다.

제안 기법은 XSS의 시그너쳐와 일부 동일하므로 XSS 탐지 기법으로 전환하는 것도 용이하다. 또한 개인정보유출도 추가적인 시그너쳐를 만들면 HTTP Outbound Traffic으로 탐지 가능하므로 향후 연구로 제시한다.

참고문헌

[1] 김미선, 김진보, 양형초, 김용민, 서재현, "웹 2.0과 Ajax 보안 취약점," 정보과학회지, 제 25권, 제 10호, 43-48쪽, 2007년 10월.

[2] 한국정보보호 진흥원, "*2009년 1월 인터넷침해사고 동향 및 분석 월보*," 2009년 2월.

[3]안철수 연구소, *http://blog.ahnlab.com/ahnlab/576*.

[4] Joanna Rutkowska, "*Introducing Stealth Malware Taxonomy*," COSEINC Advanced Malware Labs, November, 2006.

[5] 우종우, 하경휘, "시그너처 패턴기반의 악성코드 탐색도구의 개발," 한국컴퓨터 정보학회논문지, 제 10권 제 6호, 127-135쪽, 2005년 12월.

[6] 한국정보보호진흥원, "*웹 해킹을 통한 악성 코드 유포 사이트 사고 사례*," 2005년 6월.

[7] 김대유, 김정태, "홈페이지에 삽입된 악성코드 및 피싱과 파밍탐지를 위한 웹 로봇의 설계 및 구현," 한국해양정보통신학회논문지, 제 12권, 제 11호, 1993-1998 쪽, 2008년 11월.

[8] 최상명, "*중국발 해킹 위협과 대응방안*," 순천향대학원 석사학위논문, 2008년 8월.

[9] 한국정보보호진흥원, "*Muma, Hantian Trojan 분석*," 2005년 8월.

[10] 한국정보보호진흥원, "*2006년 9월 인터넷 침해사고 동향 및 분석월보*," 2006년 10월.

[11] 한국정보보호진흥원, "*자동화된 SQL Injection 공격을 통한 악성코드 대량 삽입 수법 분석 (Mass SQL Injection)*," 2008년 12월.

[12] 한국정보보호진흥원, "*웹쉘의 현황 및 분석*," 2007년 2월.

[13] 손우용, 송정길, "통합보안 관리시스템의 침입탐지 및 대응을 위한 보안 정책 모델," 한국컴퓨터정보학회논문지, 제 9권, 제 2호, 81-87쪽, 2004년 6월.

[14] 우종우, 하경휘, "시그너처 패턴기반의 악성코드 탐색도구의 개발," 한국컴퓨터정보학회논문지, 제 10권, 제 6호, 127-135쪽, 2005년 12월.

[15] 서정택, "가상환경을 이용한 악성코드 탐지기술," 정보보호학회지, 제 17권, 제 4호, 74-82쪽, 2007년 8월.

[16] 양경철, 이수연, 박원형, 박광철, 임종인, "전자우편을 이용한 악성코드 유포방법 분석 및 탐지에 관한 연구," 정보보호학회논문지, 제19권, 제1호, 93-101쪽, 2009년 2월.

[17] 박준홍, 최병호, 고대식, "행동패턴을 이용한 실시간 악성프로그램 탐지", 한국정보기술학회논문지, 제6권, 제6호, 124-130쪽, 2008년 12월.

저 자 소 개

최 병 하

2009: 단국대학교 정보통신대학원 정보통신학과(석사)

관심분야 : 네트워크 보안, 데이터베이스 최적화, 소프트웨어 개발방법론

조 경 산

1979: 서울대학교 전자공학과(학사)

1981: 한국과학원 전기전자공학과(공학석사)

1988: 텍사스 대학원(오스틴) 전기전산공학과(Ph.D.)

1988~1990: 삼성전자 컴퓨터부문 책임연구원, 실장

1990~현재: 단국대학교 컴퓨터학부 교수

관심분야 : 네트워크시스템 및 이동통신보안, 컴퓨터시스템