WMS(Webpage Monitoring System)을 이용한 웹 페이지 변조 감시 시스템에 관한 연구

이상규, 유혁
고려대학교 컴퓨터정보통신대학원 컴퓨터공학과
redzebra@korea.ac.kr, hyx@os.korea.ac.kr

A study of Web page monitoring system to prevent file modification attack

Sangkyu Lee, Chuck Yoo
Department of Computer Engineering, Korea University

요 약

현대 사회의 정보전달 매체는 대부분 웹을 통해 이뤄진다. 특히 기업의 경우에는 자신들의 홈페이지를 통해 회사소개, 제품소개 등의 정보를 전달하고 있다. 홈페이지는 단순 정보전달의 기능뿐만 아니라 기업의 이미지를 대표하는 심볼로 사용 되어진다. 이러한 웹 페이지는 보안솔루션을 이용하여 외부로부터 위·변조를 예방하고 있으나 한계를 가진다. 이 논문에서는 보안솔루션을 우회 또는 내부 사용자로부터 위·변조된 웹 페이지를 모니터링 하여 피해를 최소화하는 방안을 제시한다.

1. 서 론

월드 와이드 웹(World Wide Web)이 사람들 사이의 정보 전달(Communication)에 있어 혁명을 일으키게 된 후, 매일 만들어 지는 새로운 웹 페이지의 수는 수천만 개, 온라인 문서들은 수십억 개에 이르고 있다.[1] 기업에서 사용하는 웹 페이지는 단순 정보전달의 기능뿐만 아니라 기업의 이미지를 대표하는 중요한 의미로 해석된다. 이런 홈 페이지가 외부의 해킹이나 내부 사용자의 실수로 위·변조가 된다면 기업의 이미지는 하락하게 된다.

방화벽, IDS, IPS, 웹 방화벽등 많은 보안 솔루션을 도입하여 정보 자산을 보호하고 있지만, 보안 솔루션의 한계를 극복하지 못하고 XSS[2], SQL Injection[3], 업로드/다운로드 취약점[4]을 이용한 웹 페이지의 위·변조 공격 기술은 나날이 발전하고 있다.

완벽한 보안이 불가능하기 때문에 웹 페이지가 변경된 사실을 관리자에게 통보하여, 정상적으로 변경된 웹 페이지인지 해킹에 의한 변경된 웹 페이지인지를 확인하고, 해킹에 의해 변경되었다면 해당 페이지를 구성하는 파일을 복원하여 피해를 최소화 한다. 2006년 국가정보백서에 의하면 웹 페이지 변조 해킹이 16,692건으로 스팸릴레이, 피싱경유지, 일반 해킹에 비해 많이 발생한다.[5]

구분	2004년 총계	2005년 1	2	3	4	5	6	7	8	9	10	11	12	2005년 총계
스팸릴레이	3,297	287	490	776	524	885	830	874	324	277	213	261	593	6,334
피싱경유지	220	61	64	64	66	97	116	112	125	90	94	107	91	1,087
일반해킹	20,780	1,126	1,089	902	801	649	470	1,375	904	633	546	625	400	9,520
홈페이지변조	(4,812) 6,478	1,005	1,366	1,445	1,424	801	696	1,912	554	492	385	134	16,692	
합계	24,297 (29,109)	7,952	2,648	3,106	2,836	3,055	2,217	3,057	3,265	1,554	1,345	1,378	1,218	33,633

※ 괄호안의 수치는 2004년 해킹사고처리 현황을 2005년 기준으로 표시
(홈페이지 변조건수 추가한 것임)

[그림 1] - 해킹사고처리 현황

본 연구에서는 웹 페이지를 구성하는 파일레벨에서

의 변경여부를 체크하고 해킹에 의해 변조된 웹 페이지를 사용자에게 알려주는 방안을 제시한다. 논문의 구성은 2장에서 웹 서버의 구성 및 웹 방화벽의 특성에 대해 설명하고, 3장에서는 WMS(Webpage Monitoring System)의 아키텍처를 제안한다. 4장에서는 WMS를 통한 웹 페이지 모니터링에 대한 결과를 제시하고, 마지막으로 5장에서는 결론 및 향후 연구 방향에 대하여 기술한다.

2. 관련연구

2.1 웹 서버의 구성

웹 서버는 그림-2과 같이 사용자가 요청한 내용을 인터넷 상에서 웹 문서를 보여줄 수 있도록 해주는 프로그램이 웹 서버이다.

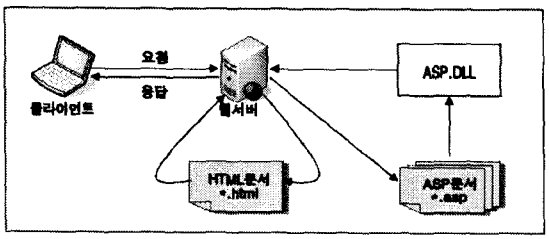

그림 2 - 웹 서버 동작원리

사용자가 요청한 웹 페이지를 처리해주는 웹 서버는 웹을 구동시키는 프로세스와 웹을 구성하는 폴더와 파일로 나누어 진다. 웹 서버의 config 파일의 설정값에 따라 서비스를 제공하는 웹의 root 디렉토리를 다르게 설정할 수 있으며, 웹 페이지를 만든 언어에 따라 html, asp, php, jsp등 다양한 확장자를 가지고 있다.

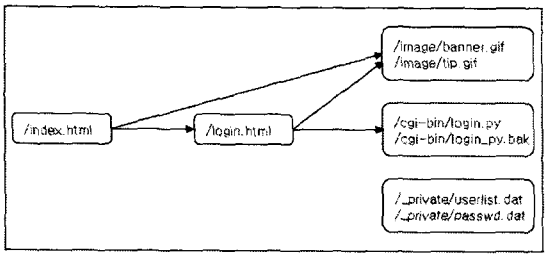

그림 3 - 웹 어플리케이션 구조

그러나 어떤 웹 서버를 사용하든지, 어떤 언어로 작성하던지 공통적인 것은 그림-3[7]과 같이 웹 서버의 config 파일에 설정된 폴더 아래 파일로 구성되어 진다는 것이다.

2.2 웹 어플리케이션 방화벽

웹 어플리케이션 방화벽은 Application 계층 분석 기술과 정규화 기술을 바탕으로 보다 특화된 검사 엔진을 탑재해 침입탐지시스템(IDS)이나 침입방지시스템(IPS)이 탐지할 수 없는 웹 관련 공격들을 방어할 수 있도록 만들어 졌다.

웹 방화벽은 네트워크를 지나는 HTTP/HTTPS 트래픽을 분석하여 웹 서버의 종류와 상관없이 보호가 가능한 네트워크 기반의 방화벽과 웹 서버가 제공하는 API를 기반으로 구현되어 IIS나 Apache 웹 서버의 플러그인 형식으로 탑재되는 웹 서버기반의 방화벽으로 구분된다.

웹 보안기능을 수행하기 위해 그림-4와 같이 웹 서버와 웹 브라우저 사이에 인라인 방식 혹은 역 프록시 방식으로 구성된다. 인라인 방식 구성은 일반적인 방화벽과 같이 라인 중간에 웹 어플리케이션 방화벽이 들어가는 형식으로 구축되고, 사용자가 웹 어플리케이션 방화벽의 존재를 인식하지 않게 되는 네트워크의 투명성을 제공한다. 역 프록시 방식은 DNS에서 웹 서버의 IP주소를 웹 어플리케이션 방화벽의 주소로 변경하여 사용자가 프록시로 설정된 웹 어플리케이션 방화벽에 웹 서버로의 접속을 요청하고, 웹 어플리케이션 방화벽은 그 내용을 다시 웹 서버에게 보내주게 된다. 이러한 구성은 웹 사이트의 구성과 보안정책에 따라 적용한다.

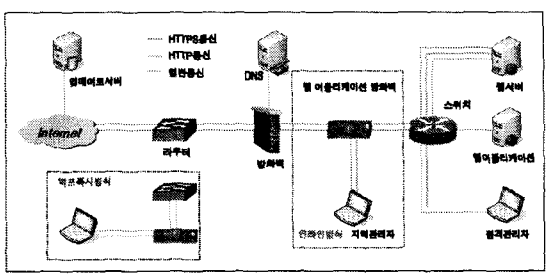

그림 4 - 웹 어플리케이션 방화벽 구성

2.3 웹 어플리케이션 방화벽 문제점

웹 방화벽을 사용하면 알려진 패턴 매칭 및 입력값에 대한 검증, 비정상 침입탐지 기능으로 HTTP 요청에 대해 차단하거나 다른 URL을 보낼 수는 있으나 웹 방화벽을 우회하거나 알려지지 않는 공격에 대해서는 완벽한 차단이 어렵다. 웹 서버를 구성하

는 파일에 대한 변경여부를 체크하는 부분이 필요하다.

3. WMS 아키텍처

웹은 그림-5와 같이 3단계로 구성되어 있다.[6] WMS는 middle layer 부분의 웹 서버를 구성하는 파일의 변경유무를 체크하여 사용자에게 통보한다.

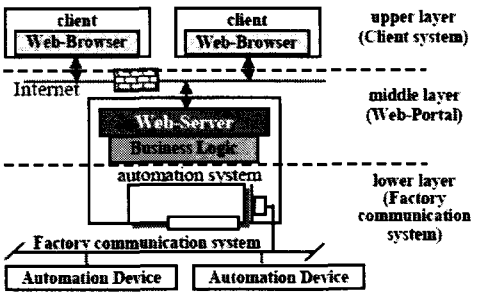

그림 5 - Web architecture

WMS는 크게 3가지로 구분된다. 그림-6의 프로세스흐름과 같이 웹 서버를 구동하고 있는 파일 리스트의 변경여부를 체크하는 부분, 변경된 파일 리스트를 사용자에게 알려주는 부분, 사용자에 의해 변경된 파일을 이전 파일로 복원하는 부분으로 나누어진다.

3.1 파일체크

처음 WMS를 구동할 경우에 웹 서버의 root 디렉토리 전체를 검색하여 현재의 파일 리스트와 파일 크기, 마지막 변경시간의 정보를 파일로 남긴다.

이번 연구에서는 특정 웹 서버의 root 하위 디렉토리의 파일의 변경여부 체크를 하였다.

웹 서버의 디렉토리 구성은 기본적으로 Apache 웹 서버의 경우 DocumentRoot 설정 값에 의해 root 디렉토리가 정해지고, Windows의 IIS 웹 서버의 경우 c:₩inetpub₩wwwroot 폴더를 root 디렉토리로 사용한다.

WMS는 root 디렉토리 아래의 폴더 및 파일의 변경유무를 스케줄 단위로 검색한다. 파일의 크기, 변경된 시간을 원본과 비교 하여 변경된 파일은 파일 위치 및 파일명을 보관한다. 또한 신규로 생성된 파일과 삭제된 파일리스트를 체크한다.

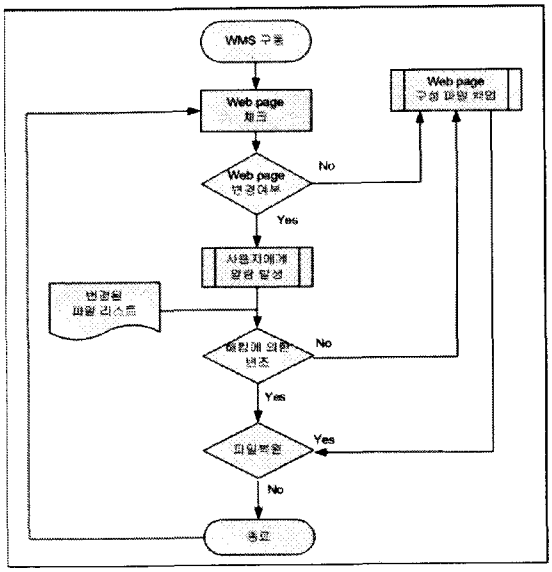

그림 6 - WMS 프로세스 흐름도

3.2 알람기능

웹 페이지를 통해 변경된 파일의 리스트, 신규로 생성된 파일리스트, 삭제된 파일리스트를 구분하여 볼 수 있도록 구성한다.

3.3 파일 복원 기능

사용자는 변경된 파일이 정상적인지 비정상적인지를 판단하여, 해킹에 의한 파일의 변경이 발생한 경우에는 기존에 백업된 파일로 교체한다.

4. 구현

본 연구에서는 Linux 서버에 Apache 웹 서버를 설치하고, asp을 이용하여 간단한 웹 페이지를 만들어, WMS의 기능을 구현해 보았다.

4.1 대상시스템

. Fedora Linux core 6

. Apache Web Server 2.2.3

4.2 WMS 테스트

WMS가 최초 구동 시 웹 서버의 root 디렉토리 하위 폴더 및 파일을 복원이 필요한 경우를 대비하여 /back 디렉토리에 저장하고, root 하위 디렉토리의 파일을 스캔하여 원본의 파일리스트를 만든다.

WMS를 구동하여 파일을 체크한 결과를 사용자는 Web 페이지를 통해 확인한다.

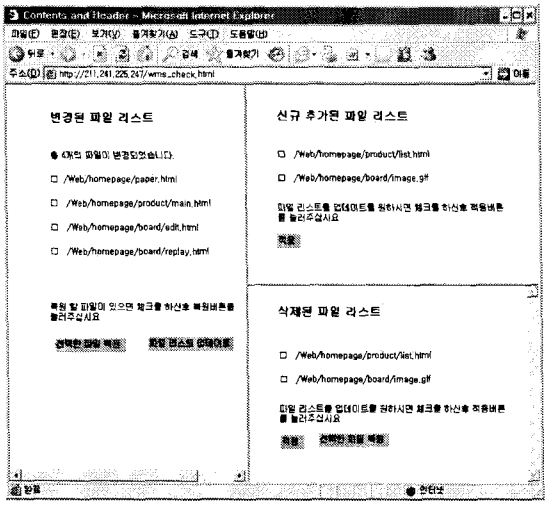

그림 7 - WMS 점검 결과

위의 그림-7에서와 같이 WMS가 구동되면 어떤 파일들이 변경이 되었는지 확인이 가능하며, 정상적으로 변경된 파일이 아닌 경우에는 해당 파일을 선택하여 백업된 파일로 복원한다. 또한 변경된 파일을 비교할 파일 리스트에 업데이트를 수행한다.

신규로 추가된 파일에 대해서는 사용자가 확인 후 정상적으로 업로드 된 파일의 경우 원본의 파일리스트에 추가한다. 또한 삭제된 파일리스트에 대해서 원본 파일리스트를 업데이트 하거나 실수로 지워진 파일에 대해서 복원한다.

4.3 성능비교

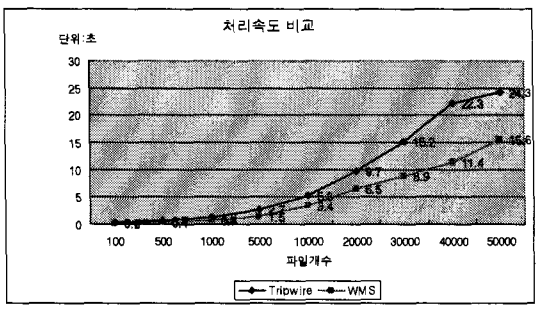

그림 8 - tripwire와 WMS 성능비교

파일의 무결성 체크 프로그램인 Tripwire는 해쉬를 이용하여 파일의 무결성을 체크한다. 그러나 파일마다 해쉬값을 비교하기 때문에 속도가 느리고,한글지원이 되지 않는 문제점이 있다. 웹 서버의 파일 개수가 1000개 미만은 Tripwire와 WMS의 성능이 비슷하였으나, 파일개수가 증가하면 WMS의 성능이 좋아졌다.

5. 결론 및 향후 과제

기존의 웹 보안 솔루션을 이용하여 기업의 정보를 보호하고 있으나, 웹 페이지에 대한 보안은 완벽하지 않았다. WMS를 이용하여 기업에서 운영하고 있는 웹 페이지에 대한 모니터링을 수행하고, 해킹이나 내부의 악의적인 행위에 의해 변경된 웹 페이지를 체크하여 사용자에게 알람을 발생하고, 해킹에 의한 변경인 경우 기존에 백업된 파일을 이용하여 복원함으로써 피해를 최소화 하였으며, 파일의 무결성 체크 프로그램인 Tripwire와 비교 시 최대 50%의 성능이 좋아졌다.

웹 페이지가 많고 파일의 수가 많은 웹 서버에 대해서는 속도가 떨어진다. 앞으로 파일의 검색 방법의 연구를 통해 보다 성능이 좋은 검색방안을 연구가 되어야 할 것이다.

6. 참고문헌

[1] 현우석,사이트간 웹 사용 마이닝을 위한 데이터 전처리의 성능향상, 한국정보과학회, P1.2-17,2006

[2] G. A. Di Lucca, A. R. Fasolino, M. Mastoianni, P. Tramontana. Identifying Cross Site Scripting Vulnerabilities in Web Applications. Proceedings of the Web Site Evolution, Sixth IEEE International Workshop on (WSE'04), September 2004

[3] William G. J. Halfond, Alessandro Orso. AMNESIA: analysis and monitoring for Neutralizing SQL-injection attacks. Proceedings of the 20th IEEE/ACM international Confrerence on Automated software engineering ASE '05, November 2005.

[5] 2006 국가정보보호백서 P.14, 2006

[6] M. Wollchlaeger, P. Neumann and Th. Bangemann, "Web service for remote maintenance of fieldbus based automation systems", IEEE-AFRICON, Oct. 2002, p.247-252

[7] 이종일, 웹 어플리케이션 공격의 이해, 2006