웹 애플리케이션 취약성 정보를 제공하는 웹 서버 모니터링 시스템

한은섭° 김명호
숭실대학교 컴퓨터 학과
xeon007@ss.ssu.ac.kr , kmh@computing.ssu.ac.kr

Web server monitoring system to provide Security Vulnerabilities information in Web Application

Eun-Sop Han° Myung-Ho Kim
School of Computing, SoongSil University

요 약

인터넷의 급속한 보급으로 인하여 많은 웹 서버들이 생겨나게 되었다. 이러한 웹 서버의 관리와 운영에 모니터링 시스템이 많이 쓰이고 있다. 대부분의 모니터링 시스템들은 각 서버의 자원 활용 상태나 서비스의 가용성에 대한 정보를 제공하고 있지만 웹 서버의 또 다른 문제인 해킹에 대한 정보는 제공되지 않고 있다. 이것은 해킹에 대한 예방이 기존의 보안 솔루션으로 가능했기 때문인데, 요즘 화두가 되고 있는 웹 해킹에 대해서는 기존의 보안 솔루션으로 예방이 어려워지고 있다. 이러한 문제점을 해결하기 위해 본 논문에서는 웹 애플리케이션의 취약성에 대해 알려주고 웹 해킹 시도로 의심되는 행위에 대한 로그를 제공해 주는 모니터링 시스템을 구현한다. 구체적으로 웹 애플리케이션 언어들 중에서 많이 쓰이는 PHP에 대한 웹 해킹 취약성에 대하여 조사하여 웹 애플리케이션의 취약성 정보를 제공해 주고 웹 해킹 의심로그를 생성해 주는 시스템을 구현한다. 구현된 시스템은 관리되는 서버의 웹 애플리케이션의 취약성을 알 수 있고 웹 해킹 시도로 의심되는 로그를 남겨줌으로써 웹 서버 관리자에게 웹 해킹에 대한 보안성 향상에 필요한 정보를 제공해주는 이점이 있다.

1. 서 론

빠른 인터넷의 확산으로 컨텐츠를 제공하는 웹 서버가 많이 증가하였다. 웹 서버들에 관리와 운영을 위해 모니터링 시스템이 많이 사용된다. 일반적으로 모니터링 시스템이란 서버의 관리와 운영을 위해 필요한 정보를 제공하는 시스템 이다. 기존의 모니터링 시스템들은 웹 서버나 클러스터를 구성하는 각각의 노드들의 자원 정보와 서비스 정보, 성능 정보 등을 제공해지만 웹 서버의 관리와 운영상에 커다란 문제점인 해킹에 대한 정보는 제공되고 있지 않다[1,2].

웹 해킹에 대한 정보를 알아내기 위해서는 시스템 자체적으로 저장되는 로그를 수작업으로 분석하거나 웹 애플리케이션으로 입력되는 입력 값들을 검증해야한다. 시스템 자체적으로 저장되는 로그는 그 양의 방대함으로 인하여 관리자들이 수작업을 통해 분석하기에는 어려움을 지니며 웹 애플리케이션으로 들어오는 입력 값들을 검증하는 형태는 애플리케이션 코드 자체에서 제공되지 않으면 매우 어렵다. 이러한 관리자의 어려움을 줄이기 위해 해킹에 대한 정보를 제공해주는 모니터링 시스템이 필요하다. 따라서 본 논문에서 웹 해킹 의심 정보와 웹 해킹 취약성의 정보를 제공해주는 모니터링 시스템을 제안한다.

웹 애플리케이션들 중에서 일반적으로 쓰이는 웹 애플리케이션 언어인 PHP에 대하여 웹 해킹의 취약성을 조사하고 PHP에 관하여 웹 해킹의 취약성 정보를 제공해주는 모니터링 시스템을 설계하고 구현한다.

본 논문은 총 5장으로 구성되어 있다. 2장에서는 일반적인 모니터링 시스템에 대하여 기술하며 3장에서는 웹 해킹에 대하여 기술한다. 4장에서는 제안된 모니터링 시스템을 설계하고 구현한 과정에 대하여 설명한다. 5장에서는 본 연구의 결론과 향후 과제에 대하여 기술한다.

2.기존의 모니터링 시스템

2.1 Ganglia

Ganglia 모니터링 시스템은 대용량 클러스터에 맞게 확장성이 뛰어난 모니터링 시스템으로 여러 계층구조를 가질 수 있다. 그러나 기본적인 클러스터 노드들의 정보(물리적 자원, 서비스 가용성)만을 제공해 줄 수 있다[1].

2.2 NetLogger

분산된 애플리케이션을 위한 성능분석, 성능측정을 수행하는 모니터링 시스템이다. 시스템에서 수행되는 애플리케이션에 대한 사용성 등을 모니터링 하는 시스템이다. 다양한 애플리케이션을 모니터링 하지만 웹 애플리케이션의 취약성은 알 수 없다[2].

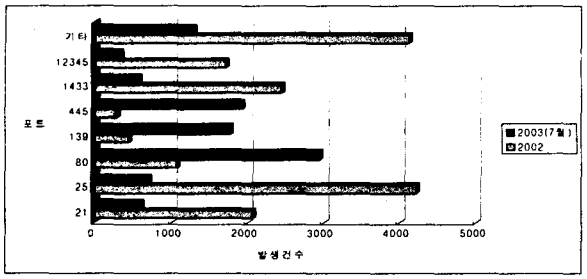
2.3 비교

앞에서 살펴본 바와 같이 기존 모니터링 시스템은 서버의 효율성, 가용성 등의 모니터링 정보를 제공하고 있다. 그러나 기존 모니터링 시스템들은 웹 해킹에 관한 어떤 정보도 제공하지 않는다.

3. 웹 해킹

3.1 해킹의 동향

웹 애플리케이션 해킹은 모든 사용자들에게 웹 서비스를 위해 열어 놓은 포트로 들어와서 웹 애플리케이션의 취약점을 통해 해킹을 수행한다. 이와 같은 웹 애플리케이션 해킹의 문제점은 라우터나 방화벽 같은 기존의 보안 솔루션으로는 막아낼 수 없다는 점이다.



<그림1> 최근해킹 동향

최근의 해킹 동향 <그림1> 에서 보이듯이 운영체제나 프로토콜 설계상의 버그 등으로 해킹을 시도하는 횟수는 줄어들고 웹 애플리케이션 취약점을 찾아서 해킹을 시도하는 웹 애플리케이션 해킹이 증가하고 있다.

3.2 웹 해킹의 취약성

웹 해킹을 방지하기 위하여 지금까지 발표되어 악용되었던 웹 애플리케이션 취약성 유형을 크게 10가지로 구분지어서 OWASP (Open Web Application Security Project)에서 발표하였다.[3]

1. 검증되지 않은 파라미터의 허용
2. 부적절한 접근 통제
3. 부적절한 계정과 세션관리
4. 크로스 사이트 스크립팅 허점
5. 버퍼 오버플로우
6. 시스템 명령어 삽입 허용
7. 잘못된 오류 처리
8. 안전하지 않은 암호화 메커니즘 사용
9. 원격 관리 허점
10. 웹 서버와 애플리케이션 서버의 구성 설정 오류

3.3 PHP의 취약성

앞에서 말한 대표적인 애플리케이션의 10가지 취약성을 PHP언어에서 살펴보면 다음과 같다.

"부적절한 계정과 세션관리"와 "안전하지 않은 암호화 메커니즘 사용"은 세션의 사용이나 사용자 인증을 하는 과정에서 생기는 취약성이다. 세션정보를 권한 없는 사용자가 접근할 수 없는 디렉터리 또는 데이터베이스에 저장하고 세션아이디나 세션의 통신에 SSL(Secure Socket Layer)을 사용하여 취약성을 제거한다.

"버퍼 오버플로우" 취약성은 PHP에선 C언어의 포인터와 같이 실행 중에 메모리를 할당 받는 방법이 없으므로 취약성이 없다.

"잘못된 오류처리" 취약성은 PHP의 설정파일인 php.ini에서 오류처리에 관한 설정을 오류가 발생하여도 외부 사용자에게 보여주지 않게 설정함으로써 제거된다.

"웹 서버와 애플리케이션 서버의 구성 설정" 취약성은 사용하고 있는 웹 서버와 애플리케이션 서버의 보안 권고문을 보고 설정을 수정하거나 패치를 하면서 제거한다.

"원격 관리 허점"은 웹 서버를 원격에서 관리하는 프로그램에 취약성이다. 기본적으로 원격관리 프로그램과 웹 서버의 통신은 SSL과 같은 보안 프로토콜을 사용 하며 관리 프로그램의 인증을 철저히 해서 제거 해야 한다.

PHP에서 웹 서버의 패치나 설정수정, 웹 서버의 정책변화만으로 취약성을 제거할 수 없는 것 들은 "검증되지 않은 파라미터의 허용", "부적절한 접근통제", "크로스 사이트 스크립팅 허점", "시스템 명령어 삽입 허용"이다. 위에 언급한 4가지의 취약성은 웹 서버에 등록되어 있는 웹 애플리케이션들의 코드를 검사함으로써 취약성들을 검출할 수 있다. 4가지의 취약성은 PHP함수가 입력된 값의 평가 없이 웹 서버의 내부 명령을 실행시킬 수 있는 함수들과 파일을 업로드 시키는 함수, SQL 쿼리를 실행시키는 함수들을 이용한 것이다.[4]

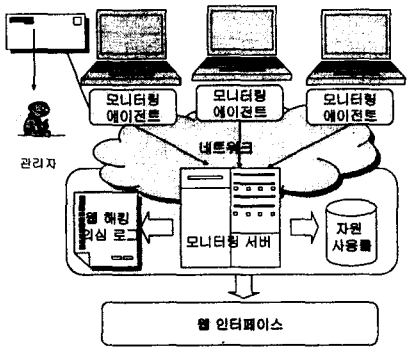
또한 위의 4가지 취약성을 이용하여 웹 해킹을 시도하는 방법도 웹 서버로 들어오는 패킷의 데이터를 검사하여 의심 가는 패킷을 알아낼 수 있다. <표1>은 PHP의 취약성을 이용하여 공격하는 대표적인 입력 값들의 예를 나타낸 것이다.

<표1> 대표적인 취약성을 공격하는 유형의 입력 값들

공격 시도	공격 내용
부적절한 접근 시도	http://server/vul.php?table=../../../../../../../../etc/passwd
실행명령삽입	test;rm -rf *
실행코드업로드	upload.php.txt
SQL코드취약성	a'|'1<2'

4.제안하는 모니터링 시스템

본 논문에서 구현한 모니터링 시스템은 자원 정보나 시스템의 성능정보 제공과 같은 일반적인 기능에 추가적으로 PHP로 만들어진 웹 애플리케이션 취약성 정보와 웹 해킹 의심되는 행위에 대한 로그를 제공하는 시스템이다.



<그림2> 전체 모니터링 시스템 구성도

이 모니터링 시스템은 <그림 2> 와 같이 크게 3부분으로 구성된다. 각각의 웹 서버에 설치되는 모니터링 에이전트와 모니터링 에이전트에서 보내온 정보를 수집하고 관리하는 모니터링 서버, 수집된 정보를 웹상에서 보여주는 모니터링 웹 인터페이스 부분으로 구성된다.

모니터링 에이전트는 각 웹 서버에 로드를 줄여주기 위하여 C 언어로 구현하였으며, 모니터링 서버와 모니터링 웹 인터페이스는 이식성과 웹 인터페이스를 구현하기 위해 자바로 구현하였다.

<표2> 대표적인 취약성을 가지는 함수

취약성	취약 함수
PHP 코드 실행	require()
	include()
	eval()
	preg_replace()
명령실행	exec()
	passthru()
	system()
	popen()
	mysql_query()
파일 노출	fopen()
	readfile()
	file()
기타 취약성	구버전 upload()
	is_uploaded_file()

4.1 모니터링 에이전트

모니터링 에이전트는 모니터링 모듈, 웹 애플리케이션 검색 모듈, 패킷 검색 모듈로 구성되어있다. 웹 애플리케이션 검색 모듈은 새로 등록되었거나 수정된 PHP파일을 찾아내어 웹 애플리케이션 취약성을 가지고 있는 함수를 검색 한다. PHP에서 여러 유형별 취약성을 가지는 대표적인 함수는 <표2> 에 나타나 있다. 검색된 파일에서 취약성을 내포하는 함수들의 사용상의 주의 점과 취약성을 최소화 할 수 있는 예제 코드를 <그림3>과 같이 이메일로 관리자에게 같이 전송한다.

```
xeon.dss.or.kr
index.php
15 line system()
        system()함수의 사용 시 주의사항
        system()함수의 입력 값에 명령어 삽입 취약성
        취약성 방지 예문
        if(!(eregi(";",$input))&&!(eregi("|",$input)))
                system($input);
```

<그림 3> 이메일 내용

패킷 검색 모듈은 웹 서버(80포트)로 들어오는 모든 패킷을 검사하여 "검증되지 않은 파라미터의 허용" 그리고 "시스템 명령어 삽입 허용", "부적절한 접근 통제", "크로스 사이트 스크립팅 허점" 취약성 등에 대한 해킹 공격으로 의심 되는 패킷들을 실시간으로 로그를 남겨 모니터링 서버에 보내준다.

대표적으로 시스템 명령을 삽입 취약성을 일으키는 형태로는 ';', '|',등 이고 검증되지 않은 파라미터의 취약성과 부적절한 접근통제를 이용하는 방법으로는 '../','₩.','PHP',''등이 있다. 크로스 사이트 스크립팅을 이용하는 입력 값은 "<",">"등이 있다.

```
Source          destination     info                                    data
203.xxx.xxx.xx 203.xxx.xxx.xx GET /index.html HTTP/1.1
203.xxx.xxx.xx 203.xxx.xxx.xx POST /textarea.php HTTP/1.1 Website
=httpd%3a%2f%2f..%2f..%2f..%2fetc%2fpasswd%0d%0a
```

<그림 4> 웹 해킹 의심 패킷 로그 내용

남겨지는 로그의 저장 형태는 <그림4>와 같이 해킹시도를 하는 사용자의 IP와 해킹의심이 되는 입력 값의 내용과 입력 값들이 전해지는 애플리케이션 이름을 남긴다.

4.2 모니터링 서버

모니터링 서버는 여러 개의 모니터링 에이전트에서 전송한 서버의 자원 정보들을 DB에 저장하는 모니터링 DB 모듈과 웹 해킹 의심 로그를 하루단위로 저장해주는 로그 저장 모듈로 구성된다.

모니터링 서버는 모니터링 에이전트에서 전송한 정보들을 실시간으로 모니터링 웹 인터페이스로 전송하고 DB를 활용하여 웹 서버의 자원상황을 저장한다. 또한 모니터링 에이전트에서 전송한 웹 해킹 의심로그를 모니터링 웹 인터페이스로 전송하고 하루단위로 의심 되는 로그를 저장한다.

4.3 모니터링 웹 인터페이스

모니터링 서버에서 전송한 실시간 정보로 현재 웹 서버들의 상황을 보여주고 한 시간, 하루, 일주일, 한달, 일년 등으로 누적된 그래프 형태로 제공하고 DB를 활용하여 사용자가 원하는 통계 그래프도 제공한다. 또한 웹 해킹 의심 로그를 각 서버별로 보여준다.

5.결론 및 향후 과제

본 논문에서 웹 서버의 관리와 운영상 중요한 웹 해킹 취약성 정보를 제공해주는 웹 서버 모니터링 시스템을 구현하였다. 모니터링의 실시간 자원정보표시 기능과 더불어 웹 애플리케이션의 취약성에 대한 정보와 웹 해킹 의심 로그를 제공함으로써 아래와 같은 효과가 기대된다.

첫째, 관리대상 웹 서버에서 동작되는 웹 애플리케이션들의 취약성을 알 수 있으므로 웹 애플리케이션 설계자들에게 취약성과 예방법을 통보, 취약성을 예방할 수 있다.

둘째, 관리대상 웹 서버에 웹 해킹의심 로그를 저장함으로써 공격자의 IP와 해킹이 시도되는 목표 애플리케이션과 해킹 시도 방법을 알려줌으로서 새로운 취약성의 예방에 관한 정보를 제공해준다. 현재 구현된 시스템은 웹 애플리케이션 중 PHP에 대하여만 검색이 되는 상태이기 때문에 다양한 웹 애플리케이션들의 취약성을 알려줄 수는 없다.

향후 연구 계획은 다양한 웹 애플리케이션들의 취약성을 검증해줄 수 있는 애플리케이션 취약성의 추출방법에 대한 연구를 수행할 것이다.

6.참고문헌

[1].Matthew.L.Massie, Brent.N.Chun, David.E.Culler, "The Ganglia Distributed Monitoring System: design ,Implementation, and Experience", February2003

[2].Brian Tierney, Dan Gunter, NetLogger: A Tookit Distributed System Performance Analysis ,Journal of parallel and Distributed Computing, 1997

[3].OWASP project, http://www.owasp.org

[4].PHP and the OWASP Top Ten Security Vulnerabilities, http://www.sklar.com/page/artcle/owasp-top-ten