웹 서비스 보안 성능 평가 테스트 방법론 연구*

이동휘** · 하옥현***

요 약

IT에서 보안은 위험 및 위협으로부터 시스템을 보호하고, 피해를 방지하며 Risk를 최소화해야 한다. 이와 같은 맥락으로 정보보안 제품의 정보가 처리, 저장, 전달되는 과정에서 정보와 시스템의 보안기준, 즉 기본적인 기밀성, 가용성, 무결성과 부차적인 명확성, 증명가능성, 감지, 경보 및 방어능력 등이 충분히 보장될 수 있도록 하여야 한다. 웹 서비스에서 보안은 가장 중요한 요소이며, 웹 특성상 서비스를 위해 80번 포트 같은 통로를 열어놔야 하는 구조로서, 웹 어플리케이션, 웹 소스 및 서버, 네트워크 모든 요소가 근본적인 취약점을 안고 있다. 이에 따라 이런 요소를 통해 웹 프로그램의 설정오류나 개발 오류 및 웹 애플리케이션 자체의 취약점을 이용한 홈페이지 와 웹 서버 해킹을 방지하며, 효율성을 높이는 웹서비스 보안 BMT 수행 방법론을 제시하고자 한다.

A Study on Web Service Security Testing Methodology for Performance Evaluation

Dong Hwi Lee** · Ok Hyun Ha***

ABSTRACT

The risks and threats in IT security systems to protect, prevent damage and Risk should be minimized. Context of information security products such as information processing, storage, delivery, and in the process of information system security standards, That is the basic confidentiality, availability, integrity and secondary clarity, potential evidence, detection, warning and defense capabilities, to ensure sufficient and should be. Web services are the most important elements in the security, the web nature of port 80 for the service to keep the door open as a structure, Web applications, web sources and servers, networks, and to hold all the elements are fundamental weaknesses. Accordingly, these elements through a set of Web application development errors and set-up errors and vulnerabilities in Web applications using their own home pages and web servers to prevent hacking and to improve the efficiency of Web services is proposed methodology performs security BMT.

Key words : BMT, Web Service

접수일 : 2010년 10월 1일; 채택일 : 2010년 12월 5일

* 본 연구는 지식경제부 지역혁신센터사업인 산업기술보호특화센터 지원으로 수행되었음.

** 경기대학교 산업보안학과

*** 교신저자, 호남대학교 경찰법행정학부

1. 서 론

IT의 가용성과 보안에 대한 요구가 계속 증가하고 있으며, 대부분 사용하고 있는 정보보호 시스템 보안의 요구도 급속히 상승하고 있다. 이에 따라 정보 보호 시스템 성능 시험과 관련된 표준 ISO/IEC 9126/12119/14598이 있다[1]. 또한 유지보수성과 소스코드 Metrics를 정의한 ISO/IEC 9126을 기반으로, 표준화된 측정 모델을 사용한 연구[4]에서 절차 표준화는 평가 작업 결과의 비교가능성을 더욱 향상시킨다[11].

보안제품 BMT와 관련된 정보 보호 표준은 수많은 전문가들이 만든 '모범사례'(Best Practices)이다. 표준은 단일한 척도로 다양한 기업이나 조직체의 프로세스를 비교하는 Benchmark이며 그들의 고객에게 투명한 프로세스를 제공한다[7].

ISO/IEC 27001 표준과 일치하는 웹 기반 ISM-Benchmark는 회사의 프로필과 25개 보안 조치 항목에 대한 질문에 답변으로부터 사용자 회사의 보안 조치 수준이 어느 정도인지를 평가하는 도구로 정보 보안을 개선하기 위해 보안 조치 개발 및 운영 단계에 사용될 수 있다[5].

하지만 본 논문에서는 웹 특성상 서비스를 위해 80포트 같은 통로를 열어놔야 하는 구조 및 웹 어플리케이션, 소스, 서버, 네트워크가 근본적인 취약점 요소를 안고 있다. 이에 따라 이요소를 통해 웹 프로그램의 설정오류나 개발 오류 및 웹 애플리케이션 자체의 취약점을 이용한 홈페이지와 웹 서버 해킹을 방지하며, 효율성을 높이는 BMT수행 방법론을 제시하고자 한다. 제 2장의 Benchmark 개요에서는 BMT의 종류와 방법에 대한 변천과정을 살펴보고 제 3장에서는 웹 보안에 관련된 내용을 알아본 후 제 4장에서는 웹 서비스 보안 성능 평가 방법론이 어떻게 구현되는 지를 살펴본다.

2. Benchmark Test

컴퓨팅 분야에서 Benchmark는 프로그램 또는 다른 여러 작업을 실행하여 객체의 상대적인 성능을 평가하며 이를 위해 수많은 표준 평가와 개별적인 시험들이 행해진다. 즉, Benchmark는 어떤 컴퓨팅 시스템의 성능에 대하여 정량적 결과를 얻을 수 있도록 하는 메커니즘이다[2]. BMT는 규격화된 평가방법으로 컴퓨팅 시스템들의 성능을 조사하고 특정 기준에 따라 이들을 서로 비교할 수 있도록 해준다. BMT는 주로 컴퓨터 시스템의 하드웨어 성능을 위한 것으로 알려져 있지만 동일한 하드웨어상에서 프로그램 언어의 컴파일러나 인터프리터의 성능을 비교하는 소프트웨어 Benchmark도 있다[9].

2.1 하드웨어 BMT

SPEC는 자사의 테스트 프로그램도 고정된 비교 지점을 갖기 위해 기준 컴퓨터(reference machine)에서 수행한다. VAX-11/780와 SPARCstation 10/40은 각각 테스트 프로그램 SPEC95와 SPEC92의 기준 컴퓨터이다.

이러한 방식으로 시스템 구성요소 및 시스템의 성능은 테스트 매개변수와 연관 지어 비교될 수 있다. 하지만 많은 업체들이 자사 시스템을 특별히 테스트 프로그램에 맞추어 최적화 시키므로 인해, 전체 성능을 테스트 매개변수와 완벽하게 연관시킬 수 없는 것이 문제점으로 지적된다[10].

2.2 소프트웨어 Benchmark

소프트웨어 Benchmark는 실행속도를 통해 서로 다른 프로그래밍시스템의 성능을 비교하는데 사용된다. 소프트웨어 Benchmark가 만들어질 때는 동일한 알고리즘이 여러 가지 프로그램 언어로 구현되어 프로그램의 실행시간이 서로 비교된다. 이 때 알고리즘이 각각의 프로그램 언어에서 최적

화 될 수 있도록 다양한 프로그램 언어의 전문가의 도움이 필요하다. 이러한 Benchmark의 고전적인 예는 Ackermann함수이다[6]. 대부분의 BMT가 CPU 시간 또는 알고리즘의 반복 횟수 등을 나타내는 테이블을 만드는데 중점을 두지만, 테스트 집합이 클 경우 이 테이블이 부정확한 경향이 있다. 테이블의 결과로부터 나온 해석은 종종 불일치하는 경우가 많다. 이러한 문제를 해결하기 위해 BMT 도구로 성능 프로파일(Performance Profile)이 제안 되었다. 성능 프로파일은 성능 매트릭스를 위한 분포함수이며 최적화 소프트웨어의 성능을 평가 및 비교하기 위한 도구로 뛰어난 결과를 보여주었다[3].

2.3 Benchmark Test 방법

모든 BMT 방법에 적용되는 4개의 원칙으로 비교가능성(comparability), 대표성 및 지표성(representative and indicative), 입도(granularity), 그리고 정확한 명세법(precise specification method)이 있다[2].

2.3.1 비교가능성

정량적 Benchmark 결과는 이질적인 인터페이스와 함께 구현된 시스템을 초월하여 비교될 수 있어야 한다.

2.3.2 대표성 및 지표

일련의 Benchmark 세트는 응용 프로그램의 도메인을 대표하고 지표가 될 수 있어야 한다.

2.3.3 입도

Benchmark의 크기의 정도는 응용 프로그램과 도메인의 구조적 병목현상(bottlenecks)을 정확히 드러낼 수 있어야 한다.

2.3.4 정확한 명세법

각 Benchmark는 영어로 설명된 기능, 환경, 측정 명세서를 포함하고 있어야 한다.

3. 웹 보안 점검 목록

보안 코딩 시 많이 사용되는 핵심 보안 프로세스를 국가정보원 및 한국인터넷진흥원(KISA)에서 권고하는 보안코딩 권고안을 준수하여 Compact-odule화 하여 실제 웹 해킹이 발생되는 point에 삽입되어, 웹 해킹을 방어하며, 실제 웹 해킹이 발생하는 Point에 대한 불법 침입 시도에 대하여 실시간 모니터링 기능, 홈페이지 위·변조 탐지 등의 기능제공여부 검증 및 판단.

- 최소한의 프로그램 수정만으로 웹 해킹 방어 가능여부
- 웹 프로그래밍 언어(asp, dotnet, php, jsp)의 특징에 최적화되게 모듈화가능여부.
- 불법침입 시도에 대한 해킹 발생 포인트 레벨에서 실시간 모니터링 기능 제공여부

3.1 웹 해킹

기존의 SQL-Injection 기법보다 확장된 개념이다. 크게 2가지 방식으로 공격이 되며 공격 쿼리의 일부분을 HEX 인코딩하거나 전체 쿼리를 HEX 인코딩하여 보안장비와 필터링 설정을 우회하는 기법이다.

Mass라는 단어의 사전적인 의미는 대량의, 집단이라는 뜻을 가지고 있다. 즉, 한 번의 공격으로 대량의 DB 값이 변조가 되어 해당 웹 사이트에 치명적인 악영향을 준다. DB 값 변조 시 악성 스크립트를 삽입하여 이용자들이 감염되거나 봇이 설치되어 DDoS 공격에 좀비 컴퓨터로 이용이 가능해진다.

Mass SQL-Injection은 IIS 환경의 MS-SQL을 사용 중인 ASP 기반 웹 애플리케이션에만 발생.

〈표 1〉 웹 서비스 취약점

취약점	현상	대응책
Path Manipulation	시스템 중요파일(password, 소스코드 등)에 접근하여 시스템 침탈 가능	인가된 경로 외의 파일에 접근하지 못하도록 소스 변경
SQL Injection	DB 공격하여 비인가 정보 획득, 데이터 변조 가능	사용자 입력값이 DB Query 문에 직접 조합되지 않도록 소스 변경
Cross-Site Scripting	페이지 가로채기, 바이러스 설치, 백도어 등의 스크립트 공격 가능	사용자 입력값에 대해 필터링 하도록 소스 변경
HTTP Response Splitting	페이지 가로채기, 바이러스 설치, 백도어 등의 스크립트 공격 가능	사용자 입력값에 대해 필터링 하도록 소스 변경
Trust Boundary Violation	프로그램 내부 데이터 조작 공격	중요하게 다루어지는 내부 데이터의 외부 사용자 입력값 이용하지 않도록 소스 변경
Uncheked Return Value : Missing Check against Null	시스템 간접정보 획득 및 프로그램 오동작 가능	사용자 입력값에 대해 검사하도록 소스 변경
J2EE Misconfiguration : Missing Error Handling	시스템 간접정보 획득 가능성	Error 발생시 시스템 정보가 노출되지 않도록 설정파일 변경

3.2 XSS(Cross Site Scripting)

게시판에 공격자가 글과 함께 악성 스크립트를 함께 올려 관리자나 일반 사용자가 공격자의 글을 읽었을 때 cookie, session 값을 공격자가 획득, 홈페이지 변조까지 할 수 있음.

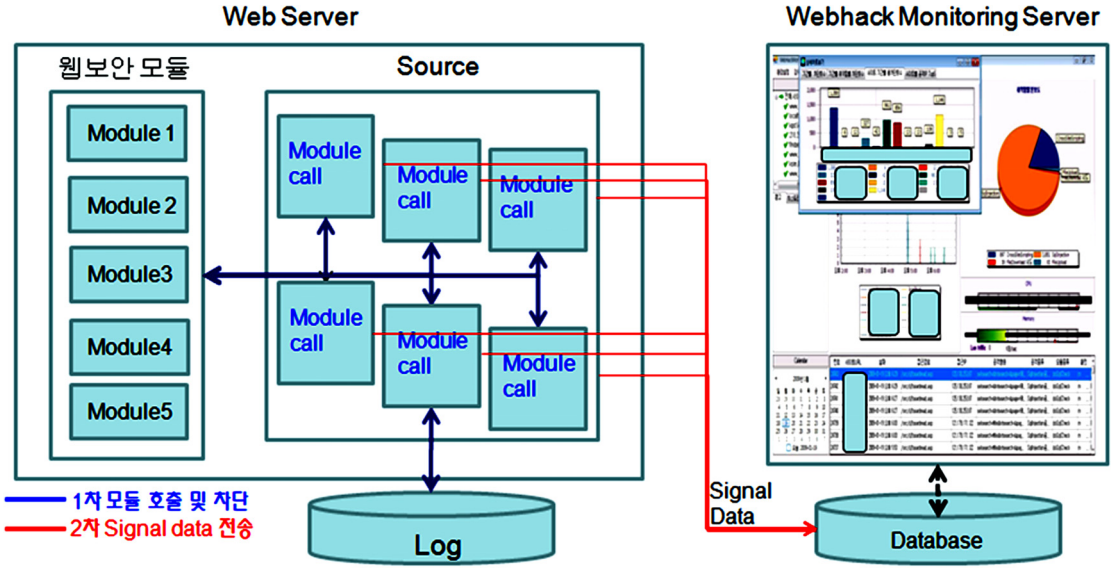
- 세부점검항목

- 인증권한우회 : 세션ID추측, SQL구문삽입, GET/POST 인자변조, 쿠키조작, 크로스사이트, 자바스크립트 인증 우회, 관리자 인증부재
- 정보노출 : 쿠키/Hidden 필드, 과다한 에러 노출, 암호화 취약, 소스 노출, 주석정보노출, 불필요한 파일
- 변조 : SQL 구문삽입, 쿠키/Hidden 필드, 자바스크립트 인증 우회, 크로스사이트
- 계정획득 : 사전기반 Brute-force 공격방어부재
- 명령실행 : SQL 구문삽입, 업로드 가능한 파일 확장자 점검부재

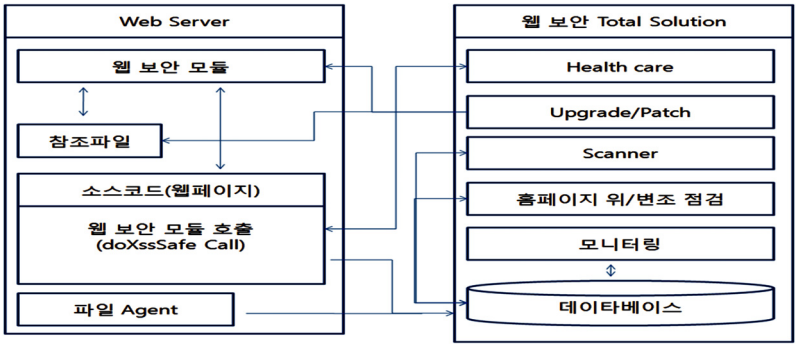
<표 1>의 목록상의 취약점 외에 백여 종 이상의 애플리케이션 취약점이 존재하며 취약점의 대부분이 애플리케이션 소스 변경에 의해 조치되어야 하며, 애플리케이션 취약점들은 개발단계 부터 디버깅 관점으로 관리되어야 한다(개발자 업무의 70%가 디버깅 업무).

4. 웹 보안 BMT 방법론

(그림 1)은 모니터링 서버를 사용하여 웹 서비스의 보안을 성능 평가하는 구조도 이다. 웹 보안을 위한 Benchmark Test는 웹 서비스 성능을 측정하는 것이 아니라 (그림 1)과 같이 보안 코딩된 모듈들이 모니터링 에이전트가 되어 정밀한 모니터링과 로깅이 가능여부를 측정하고 가용성을 해치지 않는 범위 내에서 소스 수정의 (그림 2)는 웹 서버와 웹 보안 솔루션(BMT) 간의 점검을 통해서 서



(그림 1) 웹 서비스 보안 성능평가 구조도



(그림 2) 웹 서비스 보안성능평가 수행 구조도

버의 각 모듈과 통신을 통한 점검으로 웹 서비스 상의 개인 정보 보호, 기밀 유지 및 IT 보안 또는 기술적 문제 외에도 포괄적인 내용을 담고 있어 서비스의 형태나 크기와 무관하게 적용될 수 있다. 또한 웹 보안의 동적 특성을 감안한 ISMS의 기본 개념을 적용하여 지속적인 피드백과 개선 작업을 통합하여 위협, 취약점 또는 보안사고 후 발생한 문제를 점검하는데 역점을 두고 있다.

최소화로 취약점 보완 가능 여부 및 소스레벨 차원에서 방어를 하기에 서비스와 무 저하 성능 보장여부를 측정한다. 그리고 모듈 콜과 연동하여 통합보안관제시스템과 연동 모니터링 기능 제공 여부를 확인하고 Filtering 모듈의 경우 White List 모듈과 Black List 모듈로 나누어서 적합한 룰 적용 가능여부 소스레벨에서 심층방어가 가능여부를 확인 한다.

위의 방법론을 이용하여 평상시 웹 서비스 사용 상황, 웹 방화벽을 작동한 후의 테스트, 마지막으로 웹 서비스 보안 성능평가 모듈을 적용한 후를 <표 2>와 같은 결과로 도출 하였다.

〈표 2〉 웹서비스 보안 성능평가테스트 결과

a : 웹서버, 데이터베이스, 기본구조

User1			User2		
Count	Last	Total	Count	Last	Total
1	0.203	0.203	1	0.203	0.203
2	0.141	0.344	2	0.172	0.375
48	0.266	9.891	48	0.188	10.438
49	0.203	10.094	49	0.14	10.578
50	0.281	10.375	50	0.281	**10.406**

b : 웹서버, 데이터베이스, 웹방화벽

User1			User2		
Count	Last	Total	Count	Last	Total
1	0.797	0.797	1	0.797	0.813
2	0.219	1.016	2	0.234	1.047
48	0.234	12.609	48	0.234	12.656
49	0.235	12.844	49	0.235	12.891
50	0.265	13.109	50	0.265	**13.125**

c : 기본구조에서 BMT 성능평가 후

User1			User2		
Count	Last	Total	Count	Last	Total
1	0.172	0.172	1	0.172	0.172
2	0.187	0.359	2	0.203	0.375
48	0.235	10.25	48	0.234	10.64
49	0.203	10.453	49	0.141	10.781
50	0.204	10.672	50	0.204	**10.687**

* Last : 테스트 스크립트가 수행되는 시간.
* Total : 50회 테스트 스크립트가 수행된 누적 시간.

<표 2>와 같이 웹 서비스 성능평가 모듈을 작동 한 후 테스트 하였을 때 웹 서비스 안전성과 효율성이 높다는 것을 알 수 있다.

5. 결 론

웹 서비스 구조상 근본적인 침해대응에 대한 해결책을 제시하기 위해 WEB-SOURCE에 대해 수행해야할 성능평가 항목에 대한 수행방법론을 제시한 바, 웹소스 취약점 및 각 서비스 요소들 간의 계속적인 취약점에 대해 OWASP-10에 대한 자동 모듈화 구조로 injection 테스트 하는 것이 웹 서비스 보안 성능평가 방법론이라고 본 논문에서는 제시 하였다.

웹 서비스 보안 성능평가에서 효율과 성능, 보안성을 명확히 구분하는 것은 쉽지 않은 문제이다. 기존의 사례에서처럼 성능만을 점검하는 기타 BMT 방법론 보다는 논 본문에서 제시한 방법론이 정보보안의 고유목적으로 가장 중요한 요소로 꼽히는 가용성과 연관시켜 이해할 수도 있다. 정확한 웹서비스 보안 성능평가 방법론을 결정하기 위해서 웹서비스의 보안항목과 성능항목에 대한 요소의 표준 작업이 선행 되어야 하겠다.

참 고 문 헌

[1] 한국정보보호진흥원, “정보보호시스템 성능시험 방법론 연구”, 2007.

[2] M. Tsai, C. Kulkarni, C. Sauer, N. Shah, and K. Keutzer, “A Benchmarking Methodology for Network Processors”, 1st Network Processor Workshop, 8th Int. Symposium on High Performance Computer Architectures(HPCA), Boston, MA, 2002.

[3] E. D. Dolan and J. J. More, “Benchmarking optimization software with performance profiles”, Mathematical Programming, Vol. 91, No. 2, pp. 201-213, 2002.

[4] R. Baggen, K. Schill, and J. Visser, “Standardized Code Quality Benchmarking for

Improving Software Maintainability", in 4th International Workshop on Software Quality and Maintainability(SQM 2010), 2010.
[5] IT Security Center, Information-technology Promotion Agency(IPA), "Information Security Measures Benchmark (ISM-Benchmark)", 2008.
[6] Uwe Schöning, Theoretische Informatik kurzgefasst, Spektrum Akademischer Verlag, Heidelberg, 2001.
[7] Ted Humphreys, "ISO ISMS Standards", ETSI Security Workshop, 2006.
[8] Pass Mark Software, "Anti-Virus and Internet Security Products Performance Benchmarking(2010)", 2010.
[9] Wikipedia, Benchmarking : http://de.wikipedia.org/wiki/Benchmark_(EDV).
[10] Performancetest : http://www.performance-test.de.
[11] Wikipedia, ISO/IEC 27000-series : http://en.wikipedia.org/wiki/ISO/IEC_27000-series.

이 동 휘

2000년 경기대학교 컴퓨터과학과 (이학사)
2003년 경기대학교 정보보호기술 공학과(공학석사)
2006년 경기대학교 정보보호학과 (정보보호학박사)
2008년~현재 경기대학교 산업보안학과 연구교수

하 옥 현

1978년 성균관대학교 정치 외교학과(정치학사)
1980년 서울대학교 행정 대학원 (행정학석사)
1998년 프랑스 사회과학 대학원 (EHESS) 박사과정(DEA 취득)
2005년 고려대학교 정보보호대학원(공학박사)
2008년~현재 호남대학교 경찰법행정학부 교수