홈페이지(Web)서버

구축 보안가이드

정보전산처 전산운영부

2008.4

목 차

1. 개요

이 지침은 고려대학교의 IT 보안 통제가 적절하게 수행됨을 보장하기 위함과, 교내 구축되는 홈페이지 서버의 보안사고 예방을 그 목적으로 한다.

보안사고 예방을 위해서는 최초 시스템을 구축하는 단계에서부터 보안을 고려하는 것이 중요하며, 보다 상세설정은 다음을 참고하기 바란다.

① 한국정보보호진흥원(KISA) '홈페이지 개발 보안 가이드'
- http://www.kisa.or.kr/

② 국가사이버안전센터(NCSC) '홈페이지 보안관리 매뉴얼'
- http://www.ncsc.go.kr

본 가이드에서는 운영 시스템과 네트워크 장비를 이용해 서비스 환경을 구축할 때 적용해야 하는 최소한의 보안 설정과 취약점 점검 항목을 다루고자 하였으며 시스템의 최초 설치 시 필요한 기본적인 절차와 항목들을 체크리스트 형태로 제시하였고, 웹 어플리케이션 취약점 점검 등 최근의 공격기법에 대한 점검과 대응책과, 시스템 또는 네트워크 장비에서 적용되어야 하는 기본사항들을 언급 하였으며, 세부설정에 대해서는 직접적으로 언급하지는 않았다. 보다 상세한 세부설정정보는 해당시스템/어플리케이션의 매뉴얼 또는 개발/유지보수업체를 통해 확인하여 작업을 하여야 한다.

2. 웹 서버 취약점 점검

2.1 OS/운영체제 보안

모든 시스템은 어플리케이션 보안에 앞서 호스트 OS 의 보안 작업이 선행되어야 한다.
아무리 웹 서버를 안전하게 설정 및 운영한다 해도, 웹 서버가 설치될 OS 가 안전하지 않다면 결코 웹 서버의 안전을 보장할 수 없다.

구 분	설 명
OS 최신패치를 적용	- 시스템에 설치된 OS 의 보안패치를 모두 적용하며, 항상성을 유지해야 한다
웹 서버 전용	- 가급적 웹(홈페이지) 서버 전용 호스트로 구성한다. 즉, 여타

호스트로 구성	다른 어플리케이션(메일, FTP 등)을 함께 구동하지 않는다. - 서비스운영에 최소한의 프로그램만 남겨두고, 불필요한 서비스는 반드시 제거한다. - 관리자만 로그인이 가능하도록 한다. - 개발완료 후에는 개발도구(예:컴파일러) 및 백업파일들을 반드시 제거한다
서버에 대한 접근제어	- 웹 서버 접근은 콘솔 접근만을 허용하도록 하며, 불가능 할 경우 관리자 IP 만 접근이 가능하도록 접근제어를 수행한다
강력한 관리자 계정 패스워드 사용	- 관리자 계정의 패스워드는 모든 보안의 가장 기본으로 유추가 불가능한 강력한 패스워드를 사용한다(예, "영문대소문자 + 숫자 + 특수문자"의 조합으로 8 글자 이상)

2.2 웹 서버 설치 보안

구 분	설 명
소스코드 형태의 배포본 설치	- 웹 서버 소프트웨어가 소스코드와 바이너리 형태로 배포되는 경우, 소스코드를 다운로드 받아 필요한 기능만 설치하고, - 소스의 다운로드는 해당 프로그램의 공식사이트를 이용하고, - 다운로드 후 MD5 해쉬값을 반드시 비교한다.
설치 시 네트워크 접속 차단	- 웹 서버를 설치하기 전부터 보안설정을 안전하게 끝낼 때까지 호스트의 네트워크 접속을 완전 차단한다.
웹 서비스 영역의 분리	- 웹 서버의 루트 디렉토리와 OS 의 루트 디렉토리를 다르게 지정한다.
링크 사용금지	- 공개 웹 컨텐츠 디렉토리 안에서 서버의 다른 디렉토리나 파일들에 접근할 수 있는 심볼릭 링크, aliases, 바로가기 등을 사용하지 않는다.
자동 디렉토리 리스팅 사용중지	- 디렉토리 요청 시 디렉토리 내에 존재하는 파일 목록을 보여주지 않도록 설정한다 이를 위해 자동 디렉토리 리스팅 기능의 사용을 중지시킨다.
기본 문서 순서 주의	- 웹 서버에서는 디렉토리 요청 시 기본적으로 보여지는 파일들을 지정할 수 있도록 되어 있는데 이 파일 목록을 올바로 지정하여 기본 문서가 악의적인 목적의 다른 파일로 변경되지 않도록 한다.
샘플파일, 매뉴얼파일, 임시 파일 제거	- 웹 서버 설치 후에는 웹 서버를 설치하면 기본적으로 설치되는 샘플 파일이나 매뉴얼 파일은 반드시 삭제하며, - 관리 등의 이유로 일부 필요한 경우 접근제어를 설정하고 - 웹 서버를 정기적으로 검사하여 임시 파일들을 삭제해야 한다.

웹 서버에 대한 불필요한 정보노출 방지	- 웹 서버 종류, 사용 OS, 사용자 계정 이름 등 웹 서버와 관련된 불필요한 정보가 노출되지 않도록 한다.
업로드 제어	- 불필요한 파일 업로드는 허용하지 않는다. 파일 업로드를 허용해야 한다면, 대량의 업로드로 인한 서비스 불능상태가 발생하지 않도록 한다. - 또한 업로드를 허용해야 하는 파일의 종류를 지정하여 그 외 종류의 파일들은 업로드가 불가능하도록 한다. - 업로드된 파일은 웹 서버에 의해 바로 처리되지 못하도록 해야 한다. 처리되기 전에 반드시 수동이나 자동으로 파일의 보안성 검토를 수행하도록 한다. 또한, 업로드 되는 폴더의 실행권한을 제거하여 악성 파일이 업로드 되었을 시 실행되지 못하도록 한다. 업로드 폴더를 웹 서비스 폴더와 별도로 사용하도록 한다.
인증과 접근제어의 사용	- 웹 서버에서 제공하는 인증 기능과 접근제어 기능을 필요한 곳에 적절하게 활용한다. - 대부분 웹 서버에서는 사용자 아이디/패스워드 기반의 인증 기능과 특정 IP 나 도메인에 대한 접근제어 기능을 제공한다
패스워드 설정 정책 수립	- 웹 서버의 인증 기능을 이용하는 경우에, 유추가 불가능한 강력한 패스워드를 사용한다. (예, "영문대소문자 + 숫자 + 특수문자"의 조합으로 8 글자 이상)
설치 후 패치 수행	- 웹 서버 기본 설치 후 웹 서버와 관련된 취약점 정보를 얻고, 패치나 업그레이드를 수행한다.
설정 파일 백업	- 웹 서버를 인터넷에 연결하기 전에 초기 설정 파일을 백업 받아서 보관해 둔다. - 또한 변경이 있을 때마다 설정 파일을 백업하여, 문제발생 상황을 대비하도록 한다.
SSL/TLS 사용	- 보안과 기밀성이 요구되는 경우 SSL 이나 TLS 를 사용한다.

3. DB 취약점 점검

3.1 My-SQL

구 분	설 명
DB 시스템 보안패치 적용	- My-SQL 이 동작하는 시스템에 대한 보안패치를 적용한다.

DBMS 계정 확인	- My-SQL 디폴트 설치 시 설정되지 않은 채 비어있는 데이터베이스 관리자(root) 패스워드를 변경 한다. - 또한 디폴트로 설정되는 관리자 계정(root)을 추측해 내기 어려운 이름으로 변경(Rename)한다. - My-SQL 설치 시 기본적으로 생성되어 있는'test'계정은 삭제 한다.
원격에서 My-SQL 서버로의 접속가능 여부	- 원격에서 My-SQL 통신이 불필요한 경우 디폴트로 리스닝 하는 3306/tcp 포트를 차단해 데이터베이스가 로컬로 설치된 PHD 어플리케이션에 의해서만 사용되게 한다. - 3306/tcp 포트를 리스닝 하지 못하게 하면 다른 호스트로부터 직접 TCP/IP 접속을 통한 My-SQL 데이터베이스를 공격할 가능성이 줄어들게 되며, mysql. socket 을 통한 로컬 커뮤니케이션은 여전히 가능하다.
DB 내의 사용자 별 접속/권한 설정 확인	- 데이터베이스에 대한 적절한 권한 설정이 되어 있지 않은 경우 DBA 가 아닌 사용자가 중요 테이블에 대한 조작을 할 수 있으므로 각 User 별 데이터베이스권한 설정이 적절하게 이 루어져야 한다. - 또, DB 생성 후 사용자 접근 권한 설정 시, 관리상의 편의성을 이유로 모든 권한을 부여하는 경우가 있는데 일반 사용자에게는 최소한의 권한만을 부여하도록 한다. - 특히, 일반 사용자에게 process 권한을 부여하게 되면, 해당 사용자가 'show processlist' 실행을 통해 실행 중인 쿼리를 모니터링 할 수 있게 되어 비밀번호 등이 노출되는 위험이 있다.
My-SQL 최신버전 및 보안패치 적용	- 최신 버전과 보안패치를 적용한다.

3.2 MS-SQL

구 분	설 명
최신 서비스 팩 설치 및 보안패치 설치	- 최신의 서비스 팩과 보안패치를 적용해야 한다. - MS-SQL 2000 http://www.microsoft.com/korea/sql/downloads/2000/sp4.asp - MS-SQL 2005 http://www.microsoft.com/downloads/details.aspx?familyid=b6c71ea-d649-47ff-9176-e7cac58fd4bc&displaylang=en
외부로부터의 SQL	- SQL Server 포트는 TCP/1433, TCP/1434 를 Default 로

Server 포트 접속차단 여부 확인	사용하고 있는데, 이 포트를 통해 많은 공격이 일어나고 있어, 임의의 다른 포트로 변경하여 운영해야 한다..
확장 프로시저 제거	- 서버의 유지 관리를 위해 MS-SQL 에서 제공하고 있는 확장 프로시저 중, 자주 해킹에 이용되고 있는 특정 프로시저를 제거한다. 특히 xp_cmdshell 은 반드시 제거하도록 한다.
인증	- 인증은 윈도우 인증으로 통합하여 사용한다. 강화된 보안기능을 제공하는 윈도우 인증의 암호만료, 암호속성, 감사, 계정잠금 등의 기능을 사용할 수 있다.
sa 계정 패스워드를 변경	- MS-SQL Server 설치 시 생성되는 기본 계정으로 강력한 패스워드로 변경 사용한다..

3.3 오라클

<table>
<tr><th>구 분</th><th>설 명</th></tr>
<tr><td>최소 설치를 진행한다.</td><td>- 꼭 필요한 요소만 설치하여야 한다. 무엇이 꼭필요한 요소인지 확실치 않다면, 일반적인 구성으로 설치한다</td></tr>
<tr><td>디폴트 사용자 아이디를 잠그고(Lock), 기간 만료설정 및 패스워드를 변경한다.</td><td>- 오라클 데이터베이스를 설치하면 다수의 디폴트 사용자 아이디가 생긴다. 이때 오라클의 사용자 관리도구(DBCA : Database Client Administration Tool)가 자동으로 잠그고 기간만료 시키지 못하는 아이디들 SYS, SYSTEM, SCOTT, DBSNMP, OUTLN, 그리고 3 개의 JSERV 사용자 아이디들을 잠그고 기간만료 한다.

▶ 아래 Default 계정은 계정 잠금 또는 패스워드를 변경해야 한다.
<table>
<tr><th colspan="4">오라클 Default user/password</th></tr>
<tr><td>user</td><td>password</td><td>User</td><td>password</td></tr>
<tr><td>scott</td><td>tiger or tigger</td><td>system</td><td>manager</td></tr>
<tr><td>dbsnmp</td><td>dbsnmp</td><td>sys</td><td>change_on_install</td></tr>
<tr><td>tracesvr</td><td>trace</td><td>outln</td><td>outln</td></tr>
<tr><td>ordplugins</td><td>ordplugins</td><td>ordsys</td><td>ordsys</td></tr>
<tr><td>ctxsys</td><td>ctxsys</td><td>mdsys</td><td>mdsys</td></tr>
<tr><td>adams</td><td>wood</td><td>blake</td><td>paper</td></tr>
<tr><td>clark</td><td>cloth</td><td>jones</td><td>steel</td></tr>
<tr><td>lbacsys</td><td>lbacsys</td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>디폴트 사용자 아이디 중에서 잠그고 기간 만료하지 않은 계정의 패스워드를 변경한다.</td><td>- 잠그고 기간 만료하지 않은 디폴트 사용자 계정이 있을 경우 (SYS, SYSTEM, SCOTT, DBSNMP, OUTLN, 그리고 3 개의 JSERV 사용자 계정) 패스워드를 변경한다.</td></tr>
</table>

<table>
<tr><td>데이터 목록
(Data Dictionary)
보호해야 한다.</td><td>- "데이터 목록(Data Dictionary)"를 보호하기 위해서는"파라미터 파일(Parameter File)"인 init<sid>.ora 의 내용을 OS 가 제공하는 에디터를 이용하여 아래와 같이 수정하면 된다.
O7_DICTIONARY_ACCESSIBILITY = FALSE
- 오라클 9i 의 디폴트로 위의 값을 갖지만, 오라클 8i 에서는 해당 값이 TRUE 로 설정되어 있어 반드시 수정해야 한다.</td></tr>
<tr><td>권한(Privilege)의
부여(Grant) 확인</td><td>- 사용자들에게 꼭 필요한 최소권한(least privilege)만을 부여(GRANT)하여야 한다.
- PUBLIC 사용자 그룹에서 불필요한 권한을 회수(REVOKE) 하여야 한다.
- PL/SQL 보다 강력한, 아래와 같은 패키지들도 오용될 소지가 있으므로 주의해야 한다.
<table>
<tr><th>패키지명</th><th>패키지의 역할</th><th>발생할 수 있는 문제점</th></tr>
<tr><td>UTL_SMTP</td><td>임의의 메일 메시지를 임의의 사용자간에 전송할 수 있도록 하는 패키지.</td><td>이 패키지를 PUBLIC 그룹에서 사용할 수 있도록 권한부여(GRANT)하면 허가받지 않은 메일전송이 발생할 수 있음.</td></tr>
<tr><td>UTL_TCP</td><td>외부의 네트워크 서비스로 TCP 컨넥션을 열 수 있도록 하는 패키지.</td><td>임의의 데이터가 데이터베이스 서버와 외부의 네트워크 서비스 사이에서 오갈 수 있음.</td></tr>
<tr><td>UTL_HTTP</td><td>HTTP를 통한 데이터 검색 등을 가능케 하는 패키지.</td><td>HTML 형식의 임의의 데이터가 전송될 수 있음</td></tr>
<tr><td>UTL_FILE</td><td>파일처리와 관련된 패키지</td><td>설정이 잘못되는 경우, 정보시스템상의 모든 파일에 TXT LEVEL의 접근이 가능할 수 있음.</td></tr>
<tr><td>DBMS_RANDOM</td><td>저장된 데이터를 암호화하는데 사용되는 패키지</td><td>일반적으로 대부분의 사용자들은 데이터를 암호화하는 권한을 가져서는 안됨.</td></tr>
</table></td></tr>
<tr><td>강력한 인증정책을
수립하여 운영</td><td>- 클라이언트에 대한 철저한 인증이 필요하다.
1. 원격인증기능을 제공하는 오라클 9 를 사용할 경우 이 기능의 비활성화 (FALSE)를 설정하여 보안을 강화할 수 있다.
2. 오라클 "파라미터 파일"인 init<sid>.ora 의 내용을 OS 가 제공하는 에디터를 이용하여 아래와 같이 수정한다.
" REMOTE_OS_AUTHENTICATION = FALSE "
- 데이터베이스 서버가 있는 시스템의 사용자 수를 제한 한다.
1. 오라클 데이터베이스가 운영되고 있는 시스템의 사용자 수를 OS 차원에서 제한하고,
2. 불필요한 계정은 삭제하여야 한다.</td></tr>
<tr><td>네트워크를 통한
접근제한</td><td>- 방화벽 뒤에 설치하여 DB 서버를 보호한다.
- 원격에서의 오라클 리스너의 설정을 임으로 변경할 수 없도록 변경설정을 제한한다.
1. 방법 : listener.ora (오라클 리스너 설정화일) 내의</td></tr>
</table>

	파라미터를 다음과 같이 설정한다. 2. ADMIN_RESTRICTIONS_listener_name=ON - 접속을 허용할 네트워크 IP 주소 대역을 제한 한다. 1. .클라이언트 접속을 제어하려면 "Oracle Net valid node checking"기능을 이용한다. 2. protocol.ora(Oracle Net configuration file)내의 파라미터를 아래와 같이 설정한다. tcp.validnode_checking = YES tcp.excluded_nodes = { list of IP addresses } tcp.invited_nodes = { list of IP addresses } ● tcp.invited_nodes 에 허용할 IP 를 기입한다. - 네트워크 트래픽을 암호화해라 1. 가능하다면 Oracle Advanced Security'를 사용하여, 네트워크 트래픽을 암호화 할 수 있다. 2. Oracle Advanced Security 는 오라클 DB 엔터프라이즈 에디션에서만 제공되는 기능이다. - 데이터베이스 서버의 OS 보안을 강화해라 1. 불필요한 서비스를 제거하고, 사용하지 않는 포트(TCP, UDP)를 차단한다. 2. 운영시스템의 OS 와 DB 에 대한 모든 중요 패치를 정기적으로 실시 한다. 3. 참고사이트 - http://otn.oracle.com - http://technet.oracle.com

● **주요취약점 "홈페이지 개발보안가이드"(한국정보보호진흥원)"참고**

1. Oracle tnslsnr(TNS listener) 패스워드 미설정
2. Oracle Net Services Link 버퍼 오버플로우 취약점
3. Oracle TNS Listener 버퍼 오버플로우 취약점

4. 어플리케이션 점검 방법(취약점 및 보호대책)

4.1 SQL Injection(악의적인 명령어 주입 공격)

개요	대부분의 웹 사이트들은 사용자로부터 입력받은 값을 이용해 데이터베이스 접근을 위한 SQL Query 를 만들고 있다. 사용자 로그인 과정을 예로 들면,

	사용자가 유효한 계정과 패스워드를 입력했는지 확인하기 위해 사용자 계정과 패스워드에 관한 SQL Query 문을 만든다. 이때 SQL injection 기법을 통해서 정상적인 SQL query 를 변조할 수 있도록 조작된 사용자 이름과 패스워드를 보내 정상적인 동작을 방해할 수 있으며 이러한 비정상적인 SQL Query 를 이용해 공격한다.
보호대책	1. 데이터베이스와 연동을 하는 스크립트의 모든 파라미터들을 점검하여 사용자의 입력 값이 SQL injection 을 발생시키지 않도록 수정한다. 2. 사용자 입력이 SQL injection 을 발생시키지 않도록 사용자 입력 시 특수문자(' " / ₩ ; : Space -- +등)가 포함되어 있는지 검사하여 허용되지 않은 문자열이나 문자가 포함된 경우에는 에러로 처리한다. 3. SQL 서버의 에러 메시지를 사용자에게 보여주지 않도록 설정한다. 공격자는 리턴 되는 에러 메시지에 대한 분석을 통하여 공격에 성공할 수 있는 SQL Injection 스트링을 알아낼 수 있다. 따라서 SQL 서버의 에러 메시지를 외부에 제공하지 않도록 한다. 4. 웹 애플리케이션이 사용하는 데이터베이스 사용자의 권한을 제한한다. 5. 가능하면 일반 사용자 권한으로는 모든 system stored procedures 에 접근하지 못하도록 하여 웹 애플리케이션의 SQL Injection 취약점을 이용하여 데이터베이스 전체에 대한 제어권을 얻거나 데이터베이스를 운용중인 서버에 대한 접근이 불가능하도록 한다.

4.2 업로드 취약점

개요	홈페이지는 사용자들을 위하여 여러 가지 종류의 게시판을 사용하고 게시판들은 파일을 첨부하는 기능 등 다양한 기능을 가지고 있는데 이런 게시판의 첨부파일 업로드를 기능을 악용하여 웹 서버의 권한이 노출될 수 있다. 만일 게시판에 업로드 되는 파일의 확장자에 대한 적합성 여부를 검증하는 루틴이 존재하지 않으면 공격자가 조작한 Server Side Script 파일을 업로드하고 업로드 된 파일이 서버 상에 저장된 경로를 유추한 후 이 경로를 통해 Server Side Script 파일을 실행하여 쉘을 획득할 수 있고 이러한 과정을 통해 웹 서버의 권한이 노출될 수 있다. 쉘 권한 획득 후, 시스템의 명령어를 홈페이지를 통해 실행하고 웹 브라우저를 통해 그 결과 값을 보며 시스템 관리자 권한이나 인근 서버의 침입을 시도 할 수 있다. 특히, 웹 서버 데몬의 구동이 Unix 의 root 와 같은 시스템 관리자의 권한으로 구동 될 경우, 시스템의 관리자 권한이 공격자에게 그대로 노출될 수 있는 상당히 위험한 보안 취약점이다.

보호대책	첨부 파일 업로드 기능을 통한 스크립트 업로드 및 실행을 금지시킨다. 1. Upload 파일을 위한 디렉토리에는 실행설정을 제거(웹 서버) Upload 파일을 위한 전용 디렉토리를 별도 생성하여 httpd.conf 와 같은 웹 서버 데몬 설정파일에서 실행설정을 제거함으로써, Server Side Script 가 Upload 되더라도 웹 엔진이 실행하지 않게 환경을 설정한다. 2. 첨부파일의 확장자 필터링 처리 사용자가 첨부파일의 Upload 시도 시, Upload 되는 파일의 확장자를 검토하여 적합한 파일인지를 검사하는 루틴을 삽입하여, 적합한 파일의 확장자 이외의 파일에 대해서는 업로드 되지 않도록 하며, 이런 필터링 규칙은 서버에서 구현한다.

4.3 취약한 세션 관리 (Cookie Injection)

개요	클라이언트에서 동작되는 쿠키는 암호화 등의 문제를 비롯하여 그 구조상 클라이언트 측에서의 조작으로 인한 다양한 문제점을 가지고 있어, 많은 웹 프로그래밍 언어들에는 서버에 클라이언트의 정보를 저장하는 세션(Session)을 지원하고 있다. 적절히 보호되지 않은 쿠키를 사용하면 Cookie Injection 등과 같은 쿠키값 변조를 통하여 다른 사용자로의 위장 및 권한상승 등의 문제가 생길 수 있다. 또한 쿠키 및 세션은 Cookie Sniffing 및 악성스크립트실행(XSS)를 통한 Cookie Hijacking 등과 같은 쿠키 값 복사를 통한 현재 활성화된 사용자의 권한복제 위험성이 존재한다.
보호대책	1. 전송 중의 자격 증명 보호 가장 효과적인 방법은 SSL 과 같은 기술을 사용하여 로그인 트랜잭션 전체를 암호화하는 방법이다. 서버로 전송하기 이전에 클라이언트 단에서 패스워드를 해쉬하는 형태로 변경하는 단순한 방법으로는 다른 공격자가 실제 패스워드를 모르는 상태에서 해쉬된 정보를 가로채어 서버로 그대로 전송하는 일이 발생하는 경우 별다른 보안을 제공하지 못함 2. Cookie 대신 보안성이 강한 Server Side Session 을 사용 Client Side Session 방식인 Cookie 는 그 구조상 다양한 취약점에 노출될 수 있으므로 가능한 웹 서버에서 제공되는 Server Side Session 을 사용하는 것이 바람직하다

4.4 악의적인 명령 실행(XSS)

<table>
<tr><td>개요</td><td>Cross-site scripting(이하 XSS) 취약점은 웹 페이지가 사용자에게 입력 받은 데이터를 필터링하지 않고 그대로 동적으로 생성된 웹 페이지에 포함하여 사용자에게 재전송할 때 발생한다.
자바스크립트처럼 클라이언트 측에서 실행되는 언어로 작성된 악성 스크립트 코드를 웹 페이지, 웹 게시판 또는 이메일에 포함시켜 사용자에게 전달하면, 해당 웹 페이지나 이메일을 사용자가 클릭하거나 읽을 경우 악성 스크립트 코드가 웹 브라우저에서 실행이 된다.
공격자는 XSS 취약점이 존재하는 웹 사이트를 이용하여 자신이 만든 악의적인 스크립트를 일반 사용자의 컴퓨터에 전달하여 실행시킬 수 있는데, 이러한 공격방법을 통해 사용자 쿠키를 훔쳐서 해당 사용자권한으로 로그인하거나 브라우저 제어가 가능하다</td></tr>
<tr><td>보호대책</td><td>사용자 입력으로 사용 가능한 문자들을 정해놓고, 그 문자들을 제외한 나머지 모든 문자들을 필터링 한다. 필터링 해야하는 대상은 GET 질의 무자열, POST 데이터, 쿠키, URL, 일반적으로 브라우저와 웹 서버가 주고 받는 모든 데이터를 포함한다.
< 특수문자 변경 예 >
<table>
<tr><td>변경 전</td><td><</td><td>></td><td>(</td><td>)</td><td>#</td><td>&</td></tr>
<tr><td>변경 후</td><td><</td><td>></td><td>&#40</td><td>&#41</td><td>&#35</td><td>&#38</td></tr>
</table>
게시판에서 HTML 포멧을 사용할 수 없도록 설정하고 필요한 경우 모든 HTML 을 사용하지 못하게 설정 후 필요한 HTML tag 만 쓸수 있도록 설정한다.</td></tr>
</table>

4.5 버퍼 오버플로우

<table>
<tr><td>개요</td><td>가장 일반적인 취약성의 하나로, 지정된 버퍼의 크기보다 큰 데이터를 저장함으로써 실행 시 오류를 발생시키는 취약성을 말한다. 프로그램이 버퍼 오버플로우를 내재하고 있을 경우, 해커는 이 공격을 이용하여 자신이 원하는 실행코드를 수행할 수 있고, 이 취약점을 이용해 쉘 코드를 실행함으로써 원격에서 Shell 을 얻을 수 있게 된다.
버퍼 오버플로우 취약점은 사이트의 컨텐츠를 제공하는 웹 서버, 웹 애플리케이션 서버 혹은 웹 애플리케이션 자체에 존재할 수 있다. 널리 사용되는 서버 제품군에 존재하는 버퍼 오버플로우는 일반에 널리 알려지게 되고 이로 인해 해당 제품의 사용자는 상당한 위험에 노출되게 된다. 이미지를 생성하는데 사용되는 그래픽 라이브러리와 같이 웹 애플리케이션이</td></tr>
</table>

	사용하는 라이브러리도 버퍼 오버플로우 공격에 노출될 가능성이 있다. 버퍼 오버플로우는 자체 제작한 웹 애플리케이션 코드에도 존재할 수 있으며, 자체 제작한 웹 애플리케이션의 경우 웹 애플리케이션이 전형적으로 갖는 검증 부재 문제로 인해 버퍼 오버플로우가 발생할 확률이 상대적으로 높지만, 해당 애플리케이션의 소스 코드나 상세한 에러 메시지를 일반적으로 해커가 입수하기 어려우므로, 취약점을 성공적으로 공격할 수 있는 능력이 상당히 제한된다.
보호대책	웹 서버와 웹 애플리케이션 서버 제품군 혹은 인터넷 환경 상에서 사용되는 제품들에 대해 최신의 버그 리포트를 지속적으로 참고하여, 해당 제품군의 최신 패치를 적용한다. 버퍼 오버플로우를 방지하기 위한 기초적이며 확실한 방법은 개발자가 프로그램 개발 시부터 입력 값에 대한 검증을 하는 것이다. 사용자입력, 환경변수, 파일, 타 시스템으로부터의 입력 등 다양한 프로그램의 입력에 대한 검증이 필요하다 자체 제작한 애플리케이션 코드의 경우 HTTP 요청을 통해 사용자의 입력을 받아들이는 모든 코드를 검토하여 입력 값에 대해 적절한 크기를 점검 버퍼 오버플로우를 예방하기 위하여 일반적으로 다음 사항들을 고려한다. ◦ 범위를 점검하는 안전한 함수를 사용하여 입력에 대한 점검 실시 ◦ ITS4 와 같은 코드검사 툴(source code scanner)을 사용하여 버퍼 오버플로우가 발생 가능한 함수에 대한 점검 실시 (http://www.cigital.com/its4/) ◦ 버퍼 오버플로우의 발생에 대한 검출이 가능한 컴파일러 사용 ◦ 다양한 입력을 사용한 프로그램 테스트

4.6 부적절한 파라미터

개요	일반적으로 웹 애플리케이션은 HTTP 요청(또는 파일) 값을 통해 다음 동작을 결정하게 되는데 공격자는 URL, 쿼리 문자열, HTTP 헤더, 쿠키, HTML 폼 인자, HTML hidden 필드 등 모든 HTTP 요청을 변조할 수 있으며, 이를 통해 사이트의 보안 메커니즘을 우회하고자 시도한다.
보호대책	인수 변조를 방지할 수 있는 가장 좋은 방법은 모든 인자에 대해 사용 전에 입력 값 검증을 수행 < "허용(Positive) 방식"을 사용하여 인자를 검증 예 > ◦ 데이터 유형 (문자열, 정수형, 실수형 등) ◦ 허용된 문자셋 (character set) ◦ 최대 / 최소 길이 ◦ Null 값의 허용 여부

	◦ 반드시 필요한 인자와 그렇지 않은 인자 ◦ 중복 허용 여부 ◦ 숫자의 범위 ◦ 타당한 것으로 지정된 값 (열거형 - enumeration 사용) ◦ 타당한 것으로 지정된 패턴 (정규식 사용)

4.7 접근통제 취약점

개요	접근통제는 특정 사용자들에게만 웹 콘텐츠나 기능들에 접근할 수 있도록 허가해 주는 것으로 일반적으로 관리자 페이지에 대한 접근통제가 필요 그러나 일반적으로 추측하기 쉬운 URL(ex: /admin, /manager)을 사용하고 있고있어, ID/패스워드에 대한 크랙 또는 접근 허가 정책에 대해 요청하는 부분의 정보를 변경함으로써 접근이 가능한 경우가 많다. 웹 관리자의 권한이 노출될 경우 홈페이지의 변조뿐만 아니라 취약성 정도에 따라서 웹 서버의 권한까지도 노출될 위험성이 존재한다.
보호대책	일반사용자의 접근이 불필요한 관리자 로그인 페이지 주소를 유추하기 어려운 이름으로 변경한다. 중요한 정보를 가진 웹 서버의 특정 페이지들은 관리자 또는 특정 사용자만 접근할 필요가 있는 주요 페이지들은 웹 서버에서 적절한 설정을 통하여 특정 사용자만 접근이 가능하도록 사용자 접근을 제한한다.

4.8 기 타

부적절한 환경설정 (서버 설정관련) 취약점	웹 애플리케이션을 운영하는데 있어서 개발자와 시스템 관리자들이 자주 실수로 놓치게 되어 발생하는 문제들로 테스트 파일이나 sample 파일, 웹 서버 설치 시 기본적으로 설치되는 파일 또는 관리자나 운영자가 임시로 만들어 놓은 파일들이 아무런 접근 권한 없이 웹 홈 디렉토리에 놓여 있을 경우 일반 사용자가 이 파일명을 직접 입력하여 디렉토리 정보, 시스템정보 및 중요한 파일 정보를 획득한다. 보호 대책은 웹 서버의 종류를 파악하고 이 웹 서버에서 기본적으로 설치되어 있는 파일명을 입력하여 중요한 시스템 정보, 디렉토리 정보 등이 있는지 조사하고 웹 취약점 점검도구를 이용하여 기본적으로 설치되어 있는 테스트 파일이 존재하는지 조사한다.
WebDAV 취약점	윈도우 서버에서 기본으로 설치되는 원격관리기능인 WebDEV 가 계속 사용가능하도록 설정되어 있고, WebDEV 라이브러리 파일의 속성 및 홈페이지 디렉토리에 쓰기 권한이 모두 허용되어 있는 경우 해커가 WebDEV 도구를 사용하여 원격에서 홈페이지 디렉토리에 임의의 파일을

	삽입하여 변조 가능하다. 보호 대책은 원격 홈페이지 서버 관리를 하지 않는다면 WebDEV 기능을 중지시키고 만약 꼭 사용해야 한다면 WebDEV 관련 라이브러리 파일의 보안 권한을 확인하여 'Everyone' 그룹을 삭제하고 홈 디렉토리의 쓰기 권한을 제한한다.
디렉토리 리스팅 취약점	홈페이지의 속성을 설정하는 '웹사이트 등록정보'에 특정 디렉토리에 대하여 '디렉토리 검색' 항목이 체크되어 있거나(IIS 웹서버), 'httpd.conf 파일'에서 'Indexes' 옵션이 ON 되어 있는 경우(아파치 웹서버)에 인터넷 사용자에게 모든 디렉토리 및 파일 목록이 보여지게 되고, 파일의 열람 및 저장도 가능하게 되어 비공개 자료 유출이 가능하다. 보호 대책은 각 웹서버 제품별로 설정을 변경하여 디렉토리 리스트가 보이지 않도록 조치한다.
주요 애플리케이션 보안 대책	자체적으로 홈페이지를 개발할 때 뿐 아니라 기존의 상용/공개용 웹관련 프로그램을 사용시 기존의 상용/공개용 웹 프로그램들에 존재하는 취약점들을 방치했을 경우 이를 이용한 자동화된 스캐닝 및 공격으로 인해 대규모의 피해가 발생 가능하다. 보호 대책은 상용프로그램 제공업체에서 제공하여 패지 정보를 확인하여 보안 취약점을 확인하고 이를 패치 하는 것이 반드시 필요하다.
웹페이지 개발 보안 가이드	(1) 사용자에게 전달된 값(HIDDEN form 필드, parameter)를 재사용할 경우 신뢰하지 말것 주로 회원정보 변경 모듈에 사용자의 key 값(예, id)를 hidden form 필드로 전송한 후, 이를 다시 받아서 update 에 사용하는 경우가 있는데, 공격자가 이 값을 변경할 경우 다른 사용자의 정보를 변경할 수 있는 취약점이 존재 (2) 최종 통제 메커니즘은 반드시 서버에서 수행 JavaScript, VBScript 등을 사용한 사용자 입력 값 점검 루틴은 우회될 수 있기 때문에, 서버에서 최종 점검하는 것이 필요하다. 물론 서버의 부하를 줄이기 위해서 1 차적으로 클라이언트 레벨에서 점검할 수 있으나 보안 통제수단으로 사용할 수 없다. 첨부파일 업로드 기능에 스크립트 파일의 전송을 제한하기 위해서 파일 확장자 검사를 script 를 사용해서 웹 브라우저 레벨에서 수행할 경우, 공격자는 해당 script 를 우회해서 서버에 원하는 스크립트파일을 전송 (3) 클라이언트에게 중요 정보를 전달하지 말것 Java Applet, ActiveX 를 사용해서 C/S 애플리케이션을 작성하는 경우, 클라이언트에서 실행되는 컴포넌트에 중요 정보를 하드 코딩해서는 안된다. Cookie 에 중요 정보를 전달할 경우 암호화해서 사용 (4) 중요 정보 전송 시 POST Method 및 SSL 을 적용 사용자로부터 중요 정보를 받을 때는 POST Method 를 사용해야 하며,

	그중요도에 따라 SSL 을 사용한 암호화 통신을 적용해야 한다. SSL 은 자료 전송 시 암호화를 지원하므로, 민감한 정보는 애플리케이션 레벨의 암호화를 고려 (5) 중요한 트렌젝션이 일어나는 프로세스에 사용자의 비밀번호를 재확인 사용자의 개인정보변경 프로세스에 비밀번호 재확인하는 루틴을 추가할 경우 불법적인 위장으로 인한 추가 피해를 줄일 수 있음 (6) 중요 정보를 보여주는 페이지는 캐쉬를 사용하지 못하도록 설정 중요 정보를 보여주는 화면에 no-cache 설정을 하지 않을 경우, 로그아웃을 한 이후에도 [뒤로가기] 버튼을 사용해서 해당 내용을 볼 수 있는 위험이 존재 (7) 적절한 방법으로 암호화 자체 개발한 암호화 알고리즘 사용을 지양하며, 공인된 암호화 알고리즘 (3DES, SEES, AES 등)을 사용하는 것을 고려해야 한다. 암호화키를 사용하지 않는 알고리즘은 암호화 알고리즘이 아니라, 단순 인코딩 알고리즘으로 기밀성을 보장할 수 없다. 암호화키는 소스에 hard-coding 되어서는 안되며, 제한된 사람만이 접근이 가능하도록 운영

5. 웹 패키지 S/W 관리

5.1 사용중인 웹 패키지 S/W 파악

현재 홈페이지 내에서 사용 중인 웹 패키지 S/W 를 파악하고, 리스트를 구축하여 관리 한다. 특히 공개용 웹 게시판의 경우, 소스가 공개돼 있어 취약점이 자주 발표되고 있고 또한 취약점 패치가 바로 이루어지지 않는 경우가 있을 수 있어, 중요한 웹 서버에서는 가능한 사용을 하지 않도록 한다. 중요 웹 서버에서 부득이 하게 공개용 웹 게시판을 사용할 경우에는 필히 웹 방화벽을 적용, 운영하여야 한다.

5.2 주기적인 취약점 및 패치 확인

사용 중인 웹 패키지 S/W 의 새로운 취약점 발견 여부 및 해당 취약점에 대한 패치를 주기적으로 실시하여야 한다. 주요 보안 사이트와 보안 메일링 리스트를 이용해 주요 최신 취약점을 확인하여 취약점 발표 여부와 함께 패치를 진행하도록 한다.

가. 주요 웹 패키지 리스트

- 제로보드 (http://www.nzeo.com)

- 테크노트 (http://www.technote.com)
- phpBB (http://www.phpbb.com)

나. 주요 보안 취약점 발표 사이트

- Securityfocus (http://www.securityfocus.com)
- Xfocus (http://xforce.iss.net/xforce/alerts/advisories)
- Microsoft (http://www.microsoft.com/technet/security/advisory/default.mspx)
- FrSIRT (http://www.frsirt.com/)
- milw0rm (http://www.milw0rm.com/)

다. 주요 보안관련 메일링 리스트

- Securityfocus (http://www.securityfocus.com/archive)
- Securiteam (http://www.securiteam.com/)

5.3 주요 웹 패키지 보안대책

제로보드	1. 기존 제로보드 프로그램을 일부 수정하여 사용하고 있는 경우 - 새로운 패치를 모두 설치할 경우, 운영 중인 게시판의 동작에 문제가 있을 수 있으므로 패치를 설치하지 않고, - 현재 사용 중인 버전을 확인 후, 각 패치 버전 별 수정내용을 확인하여 변경이 필요한 개별 파일의 소스를 수정하거나 부분 패치 파일을 설치한다. - 제로보드 버전 확인 방법 ■ http://www.홈페이지 주소/게시판 디렉토리명/license.txt 예) http://www.nzeo.com/bbs/license.txt 2. 제로보드 프로그램을 수정 없이 그대로 사용 중인 경우, - 가장 최신버전의 패치를 설치한다 - 제로보드 패치 다운로드 http://www.nzeo.com/bbs/zboard.php?id=cgi_download2
테크노트	1. 공식 사이트 : http://www.technote.co.kr/ 2. 테크노트 프로그램을 수정 없이 그대로 사용 중인 경우, - 가장 최신버전의 패치를 설치한다 (2007. 2, TECHNOTE 6.9P 발표). 3. 테크노트 패치 다운로드 http://www.technote.co.kr/php/technote1/board.php?board=bug&command=skin_insert&exe=insert_down_69p

〈 print.cgi 수정 〉
print.cgi 소스에서 31번째 라인에 있는 &parse; 함수의 바로 아래 라인에 아래의 코드를 추가한다.

```
&error_message('파일명 확인') if($FORM 'img'=~/\;/);
&error_message('파일명 확인') if($FORM 'img'=~/\%/);
&error_message('파일명 확인') if($FORM 'img'=~/\|/);
```

〈 library/Lib-5.cgi 수정 〉
library/Lib-5.cgi 첫 번째 라인에 아래의 코드를 추가한다.

```
&error_message('파일명 확인') if($FORM 'filename' =~/\;/);
&error_message('파일명 확인') if($FORM 'filename' =~/\%/);
&error_message('파일명 확인') if($FORM 'filename' =~/\|/);
```

그누보드

1. 공식 사이트 : http://sir.co.kr/?doc=_gb.php
2. 취약점에 대한 보안대책

- **취약점 1 : 외부 PHP 소스 실행 취약점**

index.php에 아래 내용 추가

```
if (!$doc || ereg("://", $doc))
    $doc = "./main.php";
```

- **취약점 2 : 폼메일을 이용한 스팸 메일 발송**

formmail.php 내에 다음의 내용 추가

```
// 회원에게 메일을 보내는 경우 메일이 같은지를 검사
if ($mb[mb_email] != $email)
    echo "";
    exit;

(3.38 이하 버전)
// 이전 폼 전송이 같은 도메인에서 온것이 아니라면 차단
if (!preg_match("/^(http|https):\/\/$_SERVER[HTTP_HOST]/i",
    strtolower($_SERVER[HTTP_REFERER])))
    echo "";
    exit;
```

6. 취약점 점검 체크리스트

1. 호스트 OS 보안 점검항목

No.	점검항목	O	X	비고
1	OS 에 대한 최신 패치를 적용하였는가?			
2	OS 취약점 점검을 실시하였는가?			
3	웹 서버 전용 호스트로 구성하였는가?			
4	서버에 대한 접근 제어설정을 하였는가?			
5	서버가 DMZ 영역에 위치하는가?			
6	관리자 계정은 8 자 이상(특수문자 사용) 패스워드를 사용하는가?			
7	관리자 계정의 패스워드를 주기적으로 변경하는가?			
8	파일 접근권한을 설정하였는가?			

2. 웹 서버 설치보안 점검항목

No.	점검항목	O	X	비고
1	소스코드 형태의 배포본으로 설치하였는가?			
2	설치 시 네트워크 접속 차단을 하였는가?			
3	웹 프로세스의 권한 제한을 설정하였는가?			
4	로그 파일 보호설정을 하였는가?			
5	웹 서비스 영역을 분리하였는가?			
6	서버의 다른 디렉토리로의 심볼릭 링크를 제거하였는가?			
7	디렉토리 리스팅 기능을 사용중지 하였는가?			
8	기본 문서(index 파일) 설정을 하였는가?			
9	샘플 파일, 메뉴얼 파일, 임시 파일을 제거하였는가?			
10	웹 서버에 대한 불필요한 정보 노출을 방지하였는가?			
11	불필요한 파일의 업로드를 제한하였는가?			
12	웹 서버에서 인증과 접근제어 기능을 사용하였는가?			
13	패스워드 설정 정책은 수립하여 운영하였는가?			
14	동적 컨텐츠 실행에 대한 보안 대책은 수립하였는가?			
15	웹 서버 설치 후 패치를 수행하였는가?			
16	설정 파일을 백업하였는가?			
17	SSL/TLS 을 사용하였는가?			

3. 네트워크 취약점 점검항목

No.	점검항목	O	X	비고
1	네트워크 장비의 원격 접근 제한 설정을 하였는가?			
2	SNMP 기능을 사용하였는가?			
3	네트워크 장비의 디폴트 아이디/패스워드 사용금지			
3-1	community 문자열을 재설정 하였는가?			
3-2	SNMP 암호화 기능을 사용하였는가?			
4	네트워크 장비의 디폴트 아이디/패스워드를 변경하였는가?			
5	네트워크 장비의 불필요한 서비스를 중단하였는가?			
6	설정을 통해 장비의 로그인 시간을 제한하였는가?			
7	네트워크 장비의 로그를 관리하고 있는가?			

4. DB 취약점 점검항목

4-1. My-SQL				
No	**점검항목**	**O**	**X**	**비고**
1	사용 중인 My-SQL 의 최신 패치를 적용하였는가?			
2	Default 관리자 아이디를 변경하였는가?			
3	모든 DBMS 계정에 대해 패스워드를 설정하였는가?			
4	원격에서 My-SQL 서버로의 접속을 적절히 제한하였는가?			
5	My-SQL 계정에 대한 접속차단을 설정하였는가?			
6	시스템 사용자들의 DB 에 대한 권한설정을 하였는가?			
7	데이터 베이스내의 사용자별로 접속/권한 설정을 하였는가?			
8	My-SQL 의 데이터 디렉토리 보호설정을 하였는가?			
4-2. My-SQL				
No.	**점검항목**	**O**	**X**	**비고**
1	DB 서버에 대한 OS 취약점 점검을 실시하였는가?			
2	guest 계정을 삭제하였는가?			
3	public DB 의 부여권한을 해제하였는가?			
4	SYSADMIN 그룹의 사용자 제한을 설정하였는가?			
5	DB 서버로의 원격접속을 적절히 제한하였는가?			
6	최신 서비스 팩을 설치하였는가?			
7	DB 서버로의 연결 보안감사를 설정하였는가?			
4-3. Oracle				
No	**점검항목**	**O**	**X**	**비고**

.				
1	Oracle 설치 시 최소 설치를 하였는가?			
2	디폴트 아이디의 사용제한 또는 패스워드를 변경하였는가?			
3	Data Dictionary 보호를 위해 파라미터 내용을 수정하였는가?			
4	사용자에게 최소한의 권한만을 부여하였는가?			
5	클라이언트 인증을 통한 원격인증을 제한하였는가?			
6	DB 시스템의 사용자 수를 제한하였는가?			
7	원격에서의 오라클 리스너 설정변경을 제한하였는가?			
8	원격접속을 허용할 IP 대역을 설정하였는가?			
9	DB 서버에서 불필요한 서비스를 제거하였는가?			
10	Oracle 의 취신 보안패치를 설치하였는가?			
11	DB 서버의 최신 보안패치를 설치하였는가?			

5. 웹 어플리케이션 보안점검 점검항목

5-1. SQL Injection				
No.	**점검항목**	**O**	**X**	**비고**
1	로그인 폼에 대해 점검하였는가?			
2	게시판 글 조회 란에 대해 점검하였는가?			
3	게시판 URL 조작에 대해 점검하였는가?			
4	회원가입 페이지 ID 조회 란에 대해 점검하였는가?			
5	우편번호 조회 란에 대해 점검하였는가?			
6	확장 프로시져 기능에 대해 점검하였는가? (MS-SQL 경우)			
5-2. XSS				
No.	**점검항목**	**O**	**X**	**비고**
1	게시판의 입력란 (제목, 작성자, 메일주소, 글 입력란)에 대해 점검			
2	하였는가?			
3	홈페이지의 조회 란에 대해 점검하였는가?			
4	홈페이지 URL 조작에 대해 점검하였는가?			
5-3. 파일업로드				
No.	**점검항목**	**O**	**X**	**비고**
1	게시판에 업로드 되는 파일의 확장자를 체크하는가?			
2	변경된 첨부파일의 확장자를 인식할 수 있는가?			
5-4. 쿠키변조				
No.	**점검항목**	**O**	**X**	**비고**
1	쿠키 내의 id 값을 변경 시 인식할 수 있는가?			
2	쿠키 내의 코드 값 변경 시 인식할 수 있는가?			

5-5. 다운로드 취약점				
No.	**점검항목**	**O**	**X**	**비고**
1	다운로드 URL 에 대한 유효성 여부를 체크하였는가?			
2	웹 서버에서 다운로드 가능한 확장자를 등록하였는가?			
3	디렉토리 리스팅이 발생하지 않도록 설정하였는가?			

6. 웹 패키지 S/W 관리

No.	**점검항목**	**O**	**X**	**비고**
1	사용 중인 웹 패키지 S/W 를 파악하고 있는가?			
2	주기적인 취약점 및 패치를 실시하고 있는가?			

"끝"