ISBN : 000-00-000-0000-0

웹 어플리케이션 취약점 분석
('17년 3분기)

2017. 09

KiSTi www.kisti.re.kr
한국과학기술정보연구원
Korea Institute of Science and Technology Information

웹 어플리케이션 취약점 분석
('17년 3분기)

2017. 09

부 서 : 첨단연구망센터 첨단연구망정보보호실
제출자 : 이형주 (lhj275@kisti.re.kr)

KiSTi www.kisti.re.kr 한국과학기술정보연구원 Korea Institute of Science and Technology Information

[목 차]

제1장 서론

최근 웹을 이용한 침해사고를 미연에 방지하기 위한 하나의 방법으로써 웹 취약점 분석에 관한 연구가 활발히 진행되고 있다. 네트워크 환경이 실 생활에 필수 요소로 자리 잡은 지금 웹은 모든 응용 계층 중에 가장 많이 사용하는 프로토콜이 되었다. 이러한 환경의 변화로 많은 양의 웹 응용 프로그램들이 등장하게 되었고, 이들의 취약점을 이용한 공격사례들이 증가하게 되었다.

웹 서비스는 개방된 환경에서 보안장비를 거치치 않고 사용자와 서버 간 통신이 연결되는 구조적으로 취약한 특성을 가지고 있어 이로 인해 악의적인 공격자에 의한 공격타겟이 되기 쉽다. 이러한 공격을 보호하기 위한 대책으로 보안장비 도입을 통한 실시간 모니터링, 보안정책 관리 등의 보안조치를 수행하고 있다. 하지만 이러한 보안시스템들은 웹 어플리케이션 취약점의 근원적인 문제 해소를 보장하지 못한다. 따라서 날로 지능화되는 공격기법에 대응하기 위해서는 웹 어플리케이션에 대한 지속적인 취약점 점검 및 개선조치가 반드시 필요한 실정이다.

이에, 과학기술사이버안전센터에서는 선제적인 웹 취약점 제거를 통해 보안사고를 미연에 방지할 수 있도록 자동화기반의 취약점 진단 시스템을 구축·보급하여 대상기관에서 운영중인 웹사이트의 균형적인 보안수준 향상을 도모하고 있다.

본 기술보고서에서는 17년도 3/4분기에 취약점 블랙박스 테스트를 통해 주로 탐지된『**시스템관리 취약점, 취약한 파일존재 취약점, 권한인증 취약점**』을 중심으로 취약점에 대한 상세한 설명과 취약점 개선에 필요한 조치방안을 기술하고자 한다.

제2장 관련 연구

제1절 웹 어플리케이션 취약점 유형

본 절에서는 웹 어플리케이션에서 발생하는 취약점의 정의와 주요 탐지 유형에 대하여 살펴본다. 과학기술사이버안전센터에서 정의한 17개의 취약점 유형은 아래와 같다.

1 관리자 페이지 노출 취약점

일반적으로 추측이 가능한 관리자 페이지 경로(/admin, /manager 등)를 사용하거나, 프로그램 설계상의 오류, 인증 미흡으로 인해 관 리자 메뉴에 직접 접근이 가능하며 권한인증이 가능한 취약점

2 디렉터리 나열 취약점

서버내의 모든 디렉터리 혹은 중요한 정보가 포함된 디렉터리에 대해 인덱싱이 가능하게 설정되어 중요파일 정보가 노출될 수 있는 취약점

3 시스템 관리 취약점

응용 프로그램 설치 중에 생성되는 설치·임시 파일이 존재하거나 웹상에서 윈도우 로그인 창이 노출되는 등 시스템 상 설정 미비로 인해 발생하는 취약점

4 WEBDAV 취약점

IIS 일부 버전의 취약점으로 악의적인 HTTP 요청을 이용하여 FTP나 시스템에 직접 접근하지 않고 원격에서 파일을 수정 및 처리가 가능한 취약점

5 불필요한 Method 허용 취약점

웹 서비스 제공 시 불필요한 Method(PUT, DELETE, OPTIONS 등) 허용으로 외부 공격자에 의해 악성파일을 업로드 하거나 중요파일에 대한 조작이 가능해지는 취약점

6 취약한 파일 존재 취약점

웹 루트 하위에 내부 문서나 백업파일, 로그파일, 압축파일과 같은 파일이 존재할 경우 파일명을 유추하여 파일명을 알아내고, 직접 요청하여 해킹에 필요한 서비스 정보를 획득할 수 있는 취약점

7 계정 관리 취약점

회원가입 시에 안전한 패스워드 규칙이 적용되지 않아서 취약한 패스워드로 회원 가입이 가능할 경우 무차별 대입공격을 통해 패스워드가 누출될 수 있는 취약점

8 실명인증 취약점

본인 확인 과정상에서 취약한 프로그램을 악용하여 사용자정보를 변조하는 공격으로 관리자 위장을 통해 개인정보를 수집하거나 기타 공격에 악용할 수 있는 취약점

9 전송 시 개인정보 노출 취약점

프로그램이 보안과 관련된 민감한 데이터를 평문으로 통신 채널을 통해서 송·수신 할 경우, 통신채널 스니핑을 통해 인가되지 않은 사용자에게 민감한 데이터가 노출될 수 있는 취약점

10 파일 다운로드 취약점

외부 입력값에 대해 경로 조작에 사용될 수 있는 문자를 필터링하지 않는 취약점을 악용하여 예상 밖의 접근 제한 영역에 대한 경로 문자열 구성이 가능해져 시스템 정보누출, 서비스 장애 등을 유발 시킬 수 있는 취약점

11 파일 업로드 취약점

공격자가 웹 사이트에 있는 게시판이나 자료실의 파일 업로드 기능을 이용하여 공격자가 만든 특정 공격 프로그램을 업로드하여 웹 서버의 권한 획득이 가능한 취약점

12 소스코드 내 중요정보 노출 취약점

소스코드 주석문에 민감한 정보(개인 정보, 시스템 정보 등)이 포함되어 있는 경우, 외부 공격자에 의해 패스워드 등 보안 관련정보가 노출될 수 있는 취약점

13 공개용 웹 게시판 취약점

공개용 게시판을 사용할 경우 인터넷에 공개된 각종 취약점 정보로 인해 홈페이지 변조 및 해킹 경유지로 사용될 수 있는 취약점

14 크로스사이트스크립트(XSS) 취약점

공격자가 클라이언트 스크립트를 악용하여 웹사이트에 접속하려는 일반 사용자로 하여금 공격자가 의도한 명령이나 작업을 수행하는 공격으로, 세션탈취, 웹사이트 위변조, 악성 스크립트 삽입 및 실행, 접근경로 리다이렉트 등의 다양한 공격을 유발할 수 있는 취약점

15 구문삽입(SQL-Injection) 취약점

URL 요청 또는 웹 요청에 포함되는 웹 어플리케이션에서 입력 폼 및 URL입력란에 SQL 문을 삽입하는 형태의 공격으로 시스템 내부정보를 열람 또는 조작할 수 있는 취약점

16 권한인증 취약점

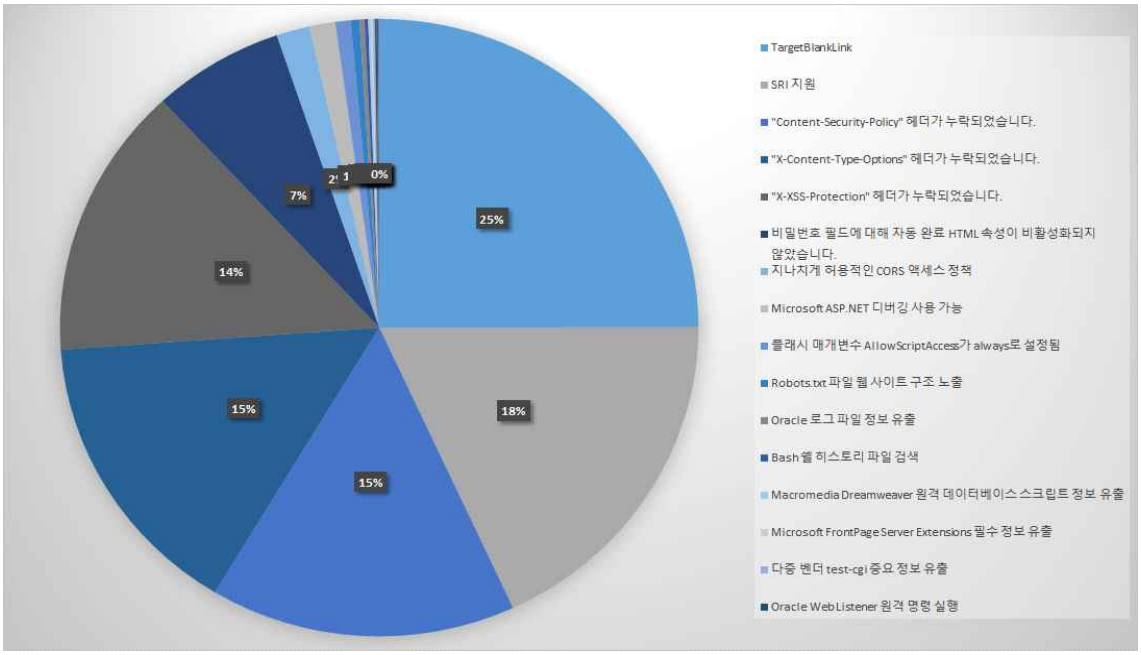
웹 어플리케이션 상에서 모든 실행 경로에 대해서 접근제어를 검사하지 않거나 불완전하게 검사하는 취약점을 이용하여 임의의 명령 실행이 가능한 악의적인 파일을 서버로 업로드하여 권한을 탈취할 수 있는 취약점

17 에러처리 취약점

웹 서버에 별도의 에러페이지를 설정하지 않은 경우, 에러 메시지를 통해 서버 데이터 정보 등 공격에 필요한 정보가 노출되는 취약점

제2절 웹 취약점 유형 별 주요 탐지현황

과학기술사이버안전센터에서는 웹 어플리케이션 분야의 취약점을 탐지하기 위하여 다수의 패턴을 보유하고 있으며, 앞서 분류된 웹 취약점 유형들이 포함하고 있는 주요 탐지패턴 현황은 아래와 같다. 이번 보고서에는 17년도 3/4분기에 주로 탐지된 21개의 취약점 패턴 분석내용을 중점적으로 다루도록 한다.



순번	취약점 유형	주요 탐지패턴
1	시스템 관리 취약점	[1-1] 비밀번호 필드에 대한 자동 완료 HTML 속성 비활성화
		[1-2] Lotus Domino 관리 데이터베이스로의 올바르지 않은 액세스
		[1-3] TargetBlankLink
		[1-4] SRI 지원
		[1-5] "Content-Security-Policy" 헤더 누락
		[1-6] "X-Content-Type-Options" 헤더 누락
		[1-7] "X-XSS-Protection" 헤더 누락
		[1-8] 지나치게 허용적인 CORS 액세스 정책
		[1-9] 플래시 매개변수 AllowScriptAccess가 always로 설정됨
2	취약한 파일존재 취약점	[2-1] Bash 쉘 히스토리 파일 검색
		[2-2] Oracle 로그 파일 정보 유출
		[2-3] Macromedia Dreamweaver 원격 데이터베이스 스크립트 정보 유출
		[2-4] Microsoft FrontPage Server Extensions 필수 정보 유출
		[2-5] 다중 벤더 test-cgi 중요 정보 유출
		[2-6] Oracle Web Listener 원격 명령 실행
		[2-7] Microsoft ASP.NET 디버깅 사용 가능
		[2-8] Robots.txt 파일 웹 사이트 구조 노출
		[2-9] 웹 서버 액세스 제어 파일 올바르지 않은 사용 권한 설정
		[2-10] PCCS MySQL Database Admin Tool 관리자 비밀번호 유출
		[2-11] Apache server-status 정보 유출
3	권한인증 취약점	[3-1] Banner Rotating 01 권한 상승

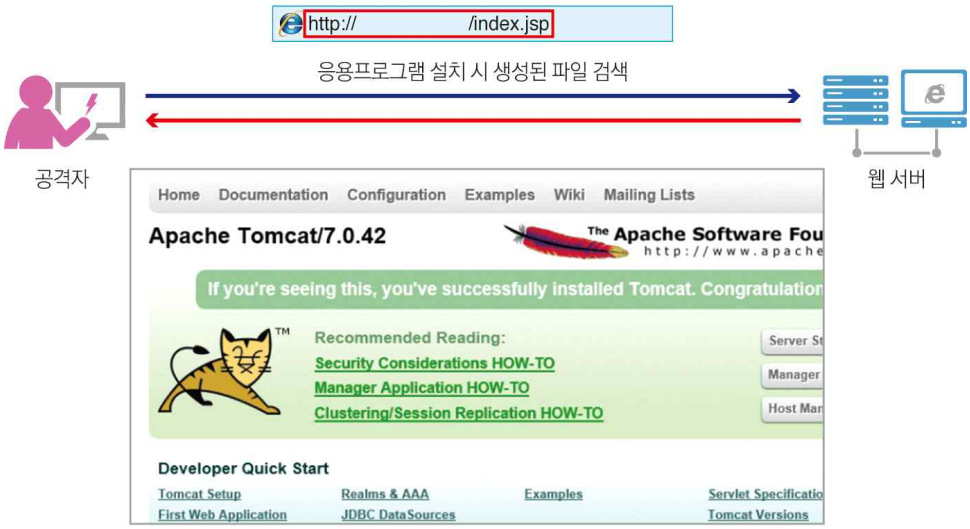
<2017년도 3/4분기 웹 취약점 유형 탐지현황>

제3장 웹 취약점 유형 별 상세 조치방안

제1절 시스템 관리 취약점

취약점 설명

웹서버 관련 응용 프로그램 설치 시 기본적으로 생성되는 임시파일이나 샘플 파일들은 시스템 혹은 DB 구성에 관한(환경 변수, 설치 모듈 정보 등) 많은 정보를 포함하고 있다. 이러한 파일이 외부로 노출되면 공격자는 시스템 내부정보 획득을 통해 2차 공격에 악용할 수 있는 위험성이 존재한다.



<Tomcat 기본 설치페이지 노출>

사전예방조치 방안

1 불필요한 파일관리

* 홈페이지 서버에 기본 설치 파일, 테스트 파일 등과 같은 불필요한 파일을 삭제
* 홈페이지 서비스와 관련 없는 디렉터리는 일반 사용자가 접근하지 못하도록 설정

```
<Files ~" ₩.bak$">
 Order allow, deny
 Deny from all
</Files>
```

2 심볼릭 링크 사용 설정 제거

* httpd.conf 파일에서 DocumentRoot 항목을 아래와 같이 수정

```
...
<Directory "/usr/local/www">
 Options FollowSymLinks ← 제거
</Directory>
...
```

3 특정파일의 내용 보기 방지

* httpd.conf 파일에서 임의의 사용자가 파일을 열람할 수 없도록 아래와 같이 설정

Apache

```
AddType application/x-httpd-php .php .php3 .inc .html .phtml .bak
AddType application/x-httpd-php-source .phps
```

CGI

```
Directory "CGI를 허용하고자 하는 디렉터리"
.....
Options ExecCGI
.....
</Directory>
AddHandler cgi-script .cgi .pl
```

* php.ini 내의 설정 중에서 display_errors 값을 Off로 설정하여 PHP의 Error나 Warning 메시지 노출 제거

```
display_errors = Off
```

* PHP에서는 각 코딩 라인에 @을 사용하여 해당 라인의 에러 메시지를 출력하지 않는 방법을 제공

```
$abc = @mysql_connect($connect, $id, $pw);
@$abc = mysql_connect($connect, $id, $pw);
```

4 주요 파일시스템 설정 변경

* /etc/fstab의 내용을 수정하여 /tmp, /dev/shm 등 임의의 디렉터리 실행 권한 제거 또한, Apache 웹서버의 설정 파일이 위치하는 디렉터리의 권한을 700으로 변경

```
/dev/hda5 on / type ext3 (rw)
none on /proc type proc (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/hda1 on /boot type ext3 (rw)
none on /dev/shm type tmpfs (rw,noexec)
/dev/hda9 on /tmp type ext3 (rw,noexec,nosuid,nodev)
/dev/hda6 on /var/lib type ext3 (rw,nosuid)
/dev/hda7 on /var/log type ext3 (rw,noexec,nosuid,nodev)
/dev/hda8 on /var/spool type ext3 (rw,noexec,nosuid,nodev)
```

[1-1] 비밀번호 필드에 대한 자동 완료 HTML 속성 비활성화

◎ 개요

최근 웹 언어 규격으로 많이 쓰이고 있는 HTML5에서는 사용자가 브라우저를 활용하여 특정 사이트에 로그인 할 때 계정정보를 전부 입력하지 않아도 편리하게 로그인할 수 있도록 계정정보 자동완성기능을 제공한다. 이러한 기능을 통해 사용자 이름과 비밀번호 필드가 자동입력 될 경우 비인가자에 의한 로그인 권한 탈취가 가능하게 된다.

◎ 조치방안

* 로그인 폼의 자동입력기능인 "autocomplete" 옵션 비활성화
  · "autocomplete"의 속성은 "On", "Off"의 두 가지 옵션이 있는데, 이를 모두 생략 할 경우에는 자동입력기능이 활성화 되므로 반드시 비활성화 옵션 설정 필요

일반적인 취약한 로그인 폼 예시

```
<form action="login.html" method="get">
    Username: <input type="text" name="firstname" /><br />
    Password: <input type="password" name="lastname" />
    <input type="submit" value="Submit" />
<form>
```

"autocomplete" 비활성화 옵션 추가 예시

```
<form action="login.html" method="get">
    Username: <input type="text" name="firstname" /><br />
    Password: <input type="password" name="lastname" autocomplete="off"/>
    <input type="submit" value="Submit" />
<form>
```

[1-2] Lotus Domino 관리 데이터베이스로의 올바르지 않은 액세스

◎ 개요

그룹웨어 웹 호스팅 어플리케이션인 Lotus Domino 서버는 웹 서버 구성관련 다양한 관리 데이터베이스로 구성되는데 이에 잘못된 접근제어 정책이 적용될 경우 비인가자에 의해 사용자 이름, 관리자 비밀번호, 머신이름 및 민감한 파일 위치 등과 같이 웹 어플리케이션에 대한 민감한 정보가 유출될 위험성이 존재한다.

◎ 조치방안

* Domino 서버에 있는 모든 ".NSF" 파일 중 데이터베이스 관련파일(NAMES, LOG, ADMIN4, DOMCFG 등) 시스템 데이터베이스에 대한 접근제어(ACL) 설정필요

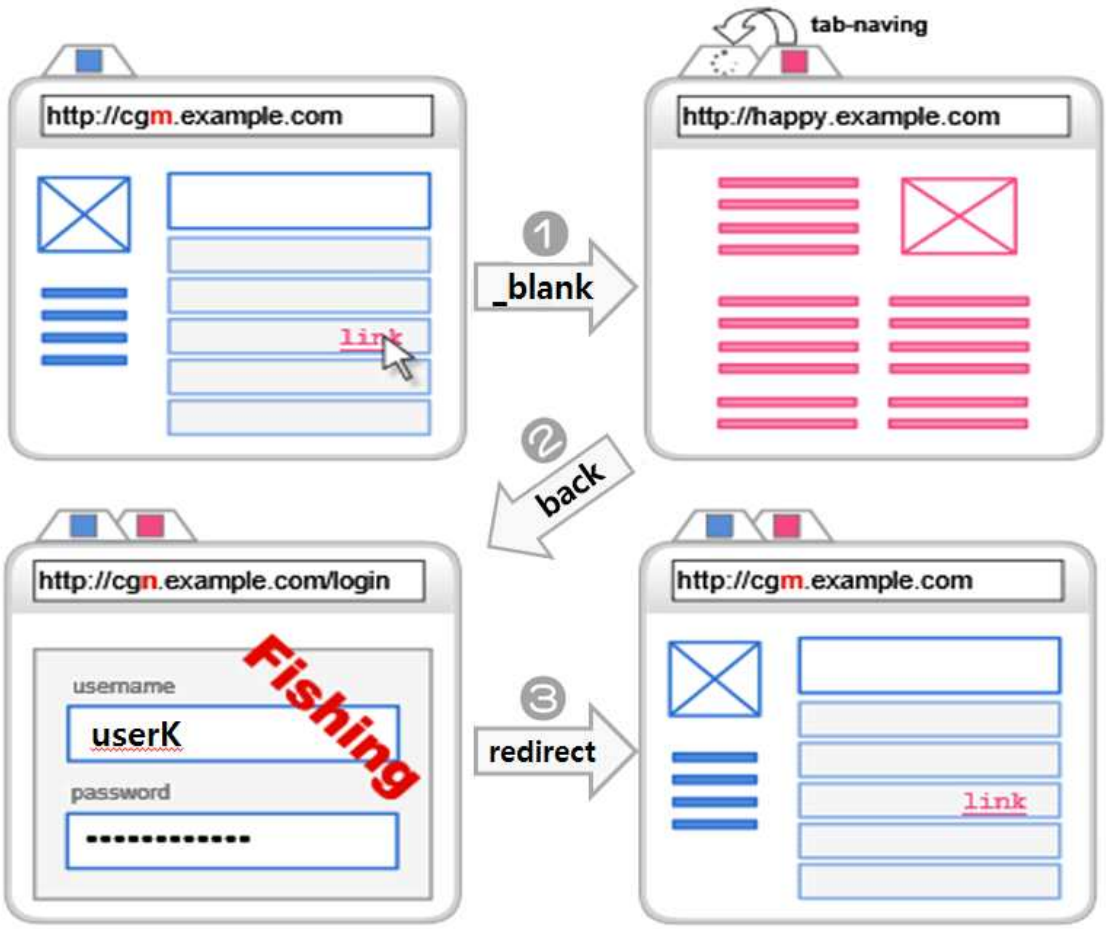
(참고) 웹 서버 관리주체의 역할에 따른 권한 목록

역할구분	기본권한	선택적 권한
Manager	· 문서생성 · 개인 에이전트 생성 · 개인폴더 / 보기 생성 · 공유폴더 / 보기 생성 · LotusScript / Java 에이전트 생성 · 공개문서 읽기 · 공개문서 작성	· 문서삭제 · 문서복제 또는 복사
Designer	· 문서생성 · 개인 에이전트 생성 · 개인폴더 / 보기 생성 · 공유폴더 / 보기 생성 · 공개문서 읽기 · 공개문서 작성	· 문서삭제 · LotusScript / Java 에이전트 생성 · 문서복제 또는 복사
Editor	· 문서생성 · 공개문서 읽기 · 공개문서 작성	· 문서삭제 · 개인 에이전트 생성 · 개인폴더 / 보기 생성 · 공유폴더 / 보기 생성 · LotusScript / Java 에이전트 생성 · 문서복제 또는 복사
Auther	· 공개문서 읽기	· 문서생성 · 문서삭제 · 개인 에이전트 생성 · 개인폴더 / 보기 생성 · LotusScript / Java 에이전트 생성 · 공개문서 작성 · 문서복제 또는 복사
Reader	· 공개문서 읽기	· 개인 에이전트 생성 · 개인폴더 / 보기 생성 · LotusScript / Java 에이전트 생성 · 공개문서 작성 · 문서복제 또는 복사
Depositor	· 문서 생성	· 공개문서 읽기 · 공개문서 작성 · 문서복제 또는 복사(공개문서읽기가 허용된 경우)
No access	· 없음	· 공개문서 읽기 · 공개문서 작성 · 문서복제 또는 복사(공개문서 읽기가 허용된 경우)

[1-3] TargetBlankLink

◎ 개요

최근 흔하게 발생하는 공격유형 중의 하나로, HTML 문서 내에서 링크(target이 blank인 Anchor 태그)를 클릭했을 때 새롭게 열린 탭 또는 페이지에서 기존 문서의 location을 피싱 사이트로 변경해 해당링크로 접근하는 사용자를 대상으로 정보를 탈취하는 취약점이 발생하며, 이 공격방식은 메일이나 오픈 커뮤니티에서 쉽게 악용될 수 있다.



<피싱 사이트를 통한 정보 유출 예시>

◎ 조치방안

* 기존 문서의 location을 피싱사이트로 변조하지 못하도록 링크에 "rel=noopener noreferrer" 옵션 추가

Secure Coding

```
<a href="http://example.com" target="_blank"
rel="noopener noreferrer">Link</a>
```

*noopener : location 변경과 같은 비정상적인 자바스크립트의 요청을 거부
*noreferrer : 브라우저에서 링크 접근 시 HTTP referrer 헤더를 전달하지 않는 설정옵션

[1-4] SRI 지원

◎ 개요

대부분의 브라우저에서는 Javascript로 제작된 HTTP(S) 요청으로 다양한 기능을 구현할 수 있도록 기본적으로 자바스크립트 실행기능이 활성화 되어있어 동적 스크립트가 실행될 수 있는 환경에 노출되어 있다. 이러한 취약점을 악용하여 공격자가 웹페이지에 악성 Javascript를 삽입할 경우, 해당 웹페이지에 접근하는 사용자들이 악성스크립트에 감염될 수 있는 위험성이 존재한다.

◎ 조치방안

* 다른 도메인의 SRC 태그를 사용하는 스크립트 및 링크에 대한 무결성 검사를 지원할 수 있도록 해쉬 기반 검증기능 구현

SRI를 지원하지 않는 취약한 스크립트 예시

```
<script src="https://example.com/example-framework.js"
crossorigin="anonymous"></script>
```

SRI를 지원하는 샘플 스크립트 예시

```
<script src="https://example.com/example-framework.js“
integrity="sha384-oqVuAfXRKap7fdgcCY5uykM6+R9GqQ8K/uxy9rx7HNQlGYl1kPzQho1
wx4JwY8wC"crossorigin="anonymous"></script>
```

[1-5] "Content-Security-Policy" 헤더 누락

◎ 개요

대부분의 웹 브라우저에서 발생하는 공격은 브라우저가 정상적인 스크립트와 공격자가 악의적으로 주입한 스크립트를 구별할 수 없는 특성으로 인해 발생한다. 이렇게 안전하지 않은 웹 어플리케이션 환경으로 인해 브라우저가 페이지를 렌더링하는 방식을 수정하여 XSS를 포함한 여러유형의 공격에 노출될 수 있다.

◎ 조치방안

* XSS와 같은 공격의 피해를 최소화 하기 위한 옵션으로, 스크립트를 허용할 URL, 즉 신뢰할 수 있는 URL을 헤더에 설정하고 브라우저가 해당 소스의 리소스만 실행하거나 렌더링하도록 설정

Apache

.htaccess 파일 설정

```
Header always set Content-Security-Policy "default-src https: data: 'unsafe-inline' 'unsafe-eval'"
```

nginx

nginx.conf 파일 설정

```
add_header Content-Security-Policy "default-src https: data: 'unsafe-inline' 'unsafe-eval'" always;
```

lighttpd

lighttpd.conf 파일 설정

```
setenv.add-response-header = ("Content-Security-Policy" => "script-src 'self'; object-src 'self'")
```

HTML

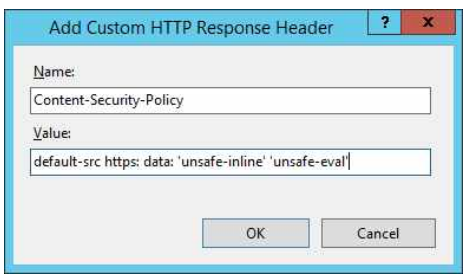
헤더 파일 설정

```
<meta http-equiv="Content-Security-Policy" content="default-src https:">
```

IIS

IIS manager → HTTP Response Headers → Add → Name : Content-Security-Policy 기입 → Value : default-src https: data: 'unsafe-inline' 'unsafe-eval' 기입 설정

<HTTP Response Headers 메뉴>



<HTTP Content-Security-Policy>

[1-6] "X-Content-Type-Options" 헤더 누락

◎ 개요

안전하지 않은 웹 어플리케이션 환경으로 인해 브라우저에서 컨텐츠 렌더링 시 확장자 검증을 하지 않는 취약점을 악용하여 공격자가 신뢰되지 않은 컨텐츠를 임의로 업로드 할 경우, HTML 태그 변조를 통해 악성코드 배포, 시스템 장악, 민감정보 유출 등 다양한 공격에 악용될 수 있다.

◎ 조치방안

* 신뢰되지 않은 컨텐츠가 사용자 브라우저에서 실행되는 것을 방지하기 위해 모든 발신요청에서 업로드 확장자와 일치할 경우에만 실행하도록 HTTP 헤더 옵션 설정

Apache .htaccess 파일 설정

```
Header always set X-Content-Type-Options "nosniff"
```

nginx nginx.conf 파일 설정

```
add_header X-Content-Type-Options "nosniff" always;
```

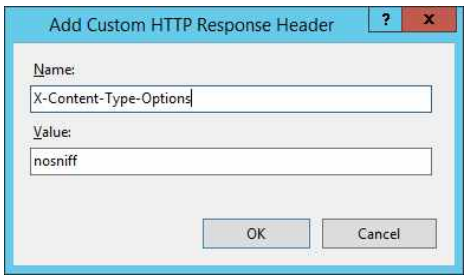
lighttpd lighttpd.conf 파일 설정

```
setenv.add-response-header = ("X-Content-Type-Options" => "nosniff",)
```

IIS

IIS manager → HTTP Response Headers → Add → Name : X-content-Type-Options 기입 → Value : nosniff 기입 설정

<HTTP Response Headers 메뉴>

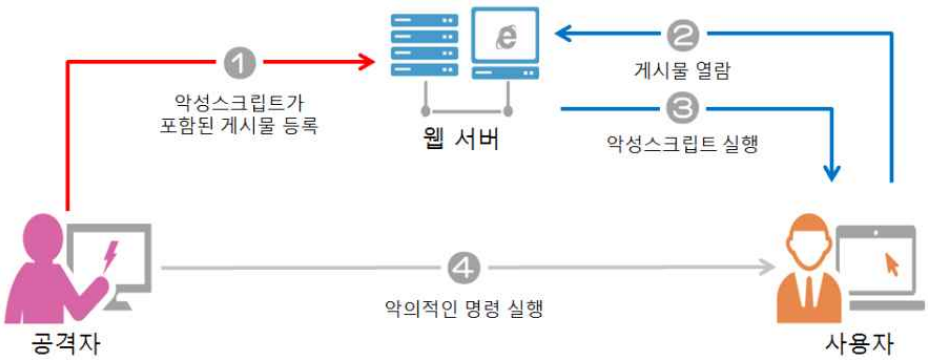


<HTTP X-Content-Type-Options>

[1-7] "X-XSS-Protection" 헤더 누락

◎ 개요

"X-XSS-Protection" 응답 헤더는 크로스 사이트 스크립팅 공격을 감지하고 중지시키는 브라우저(internet Explorer, Chrome, Safari)의 기능으로 사용되는데, 헤더에 보호조치가 되어있지 않을 경우, 공격자는 클라이언트 스크립트를 악용하여 웹사이트에 접속 하려는 일반 사용자로 하여금 공격자가 의도한 명령이나 작업을 수행할 수 있는 취약점이 존재



<저장형 XSS 공격으로 악의적인 명령 실행>

◎ 조치방안

* XSS와 같은 공격의 피해를 최소화 하기 위한 옵션으로, 스크립트를 허용할 URL, 즉 신뢰할 수 있는 URL을 헤더에 설정하고 브라우저가 해당 소스의 리소스만 실행하거나 렌더링하도록 설정

Apache

.htaccess 파일 설정

```
Header set X-XSS-Protection: 1; mode=block
```

nginx

nginx.conf 파일 설정

```
add_header X-XSS-Protection "1;mode=block";
```

lighttpd

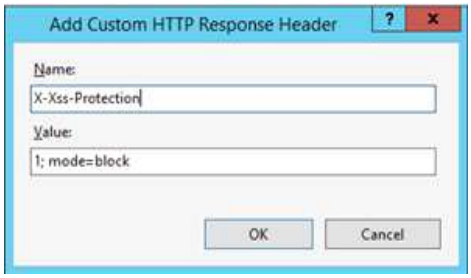
lighttpd.conf 파일 설정

```
setenv.add-response-header = ("X-XSS-Protection" => "1; mode=block",)
```

IIS

IIS manager → HTTP Response Headers → Add → Name : X-Xss-protection 기입 → Value : 1; mode=block 기입 설정

<HTTP Response Headers 메뉴>

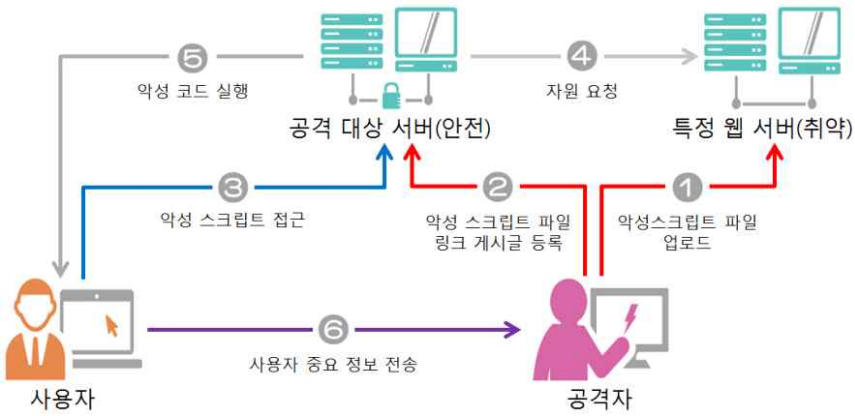


<HTTP X-Xss-Protection>

[1-8] 지나치게 허용적인 CORS 액세스 정책

◎ 개요

CORS(Cross-Origin Resource Sharing)는 오픈된 웹 환경에서 외부 웹 사이트의 리소스를 호출할 수 있도록 지원하는 표준 규약으로, 웹 사이트가 자원을 복제할 필요없이 외부 사이트의 자원을 요청할 수 있도록 허용하게 되는데 사용자가 속해 있는 내부 네트워크에 존재하는 웹 사이트가 Access-Control-Allow-Orign 응답 헤더를 잘못 정의 했을 경우 공격자는 외부에서 직접 접근할 수 없는 내부 네트워크의 웹사이트에 접속할 수 있다. 이를 통해 접근제어 우회 및 원격 쉘 업로드, 세션탈취 등 다양한 공격에 악용될 수 있다.



<악성 스크립트를 통한 세션 정보 탈취>

◎ 조치방안

* 외부 사이트에 대한 액세스 부여
* 외부 액세스가 필요하지 않은 경우 헤더 제거
  · 신뢰할 수 있는 사이트 목록을 "Access-Control-Oringin" 헤더에 추가

```
Access-Control-Allow-Origin: http://example.com
```

[1-9] 플래시 매개변수 AllowScriptAccess가 always로 설정됨

◎ 개요

FLASH SWF 파일의 경우 애니메이션을 보여주거나 역동적인 메뉴 바 등을 구현할 때 주로 활용되나 임의의 스크립트를 사용해 자바스크립트처럼 프로그래밍적인 기능을 추가하는 것이 가능한 구조적인 취약점을 악용하여 공격자는 악의적인 스크립트가 삽입된 플래시 파일을 특정 태그에 사용하여 악성스크립트가 자동으로 실행되도록 할 수 있다. 이러한 FLASH 타입의 SWF 컨텐츠를 등록할 때, AllowScriptAccess 매개변수 값을 'always'로 설정할 경우 외부 도메인의 페이지와도 통신할 수 있게 되어 내부서버 정보 유출이나 XSS 공격에 악용될 수 있다.

```
<html>
<body>
    <h1>DICA Exploit Analysis</h1>
    <object width="1024" height="768">
     <param name="movie" value="cve_2014_0556.swf"> </
param>
     <param name="allowscriptaccess" value="always"></
param>
     <embed src=" cve_2014_0556.swf "
type="application/x-shockwave-flash"
allowscriptaccess="always" width="1024" height="768">
     </embed>
     </object>
  </body>
</html>
```

<취약점을 유발하는 cve_2014_0556.swf 파일을 로딩하는 html 스크립트>

◎ 조치방안

* SWF 파일의 액세스 권한을 동일한 도메인만 허용할 수 있도록 AllowScriptAccess 매개변수를 'sameDomain'으로 설정 권고

```
〈object …〉
…
〈param name="allowscriptaccess" value="samedomain" /〉
…
〈/object〉
```

제2절 취약한 파일존재 취약점

취약점 설명

웹서버 구성 시 웹 루트 하위에 내부 문서나 로그파일, 압축파일, 백업파일과 같은 불필요한 파일이 존재할 경우 시스템 구성에 관한 많은 정보가 노출될 수 있다. 이러한 파일이 외부로 노출되면 공격자는 시스템 내부정보 획득을 통해 2차 공격에 악용할 수 있는 위험성이 존재한다.

<백업파일 노출>

사전예방조치 방안

* 웹서버 운영상에 필요한 파일인 경우, 웹서버에서 직접 접근이 불가능하도록 외부 디렉터리로 이동

주요 백업/로그 파일 확장자				
.bak	.backup	.old	.log	.!
.back	.sql	.txt	.new	.tmp

* 백업파일 및 로그파일 삭제 조치

IIS

```
dir C:₩(웹 디렉터리) * .bak(파일 확장자)
```

Apache

```
find /(경로) -name * .bak(파일 확장자)
```

[2-1] Bash 쉘 히스토리 파일 검색

◎ 개요

Bash 쉘 히스토리 파일은 사용자가 입력했던 명령어를 자동으로 저장하는 기능을 제공하는데, 디렉터리 구조, 실행 명령어, 계정정보 등 다양한 정보를 가지고 있다. 해당파일에 접근권한을 부적절하게 설정할 경우 일반적인 사용자에게도 중요정보를 노출 시킬 수 있으며, 공격자가 이를 악용하여 서버 기밀정보 유출 등과 같은 2차 공격에 활용 할 수 있다.

<부적절한 접근정책에 의한 정보유출>

◎ 조치방안

1 .bash_history 파일 제거

```
[root@Example] rm -rf ~/.bash_history
```

2 .bash_history 액세스 제한

설정 1 모든 사용자 제한

```
[root@Example] echo "unset HISTFILE" >> /etc/profile
[root@Example] source /etc/profile
```

설정 2 특정 사용자 제한

```
[root@Example] echo "unset HISTFILE" >> /home/ruo91/.bash_profile
```

3 메모리에 남아있는 history 명령어 삭제

```
[root@Example] history -c (또는 reboot)
```

* 중요정보가 노출될 수 있는 .bash_history 파일을 제거를 권장하나, 부득이하게 관리자가 사용해야할 경우 접근권한을 올바르게 설정

[2-2] Oracle 로그 파일 정보 유출

◎ 개요

Oracle에는 운영관리를 목적으로 통신을 주고받은 서버와 클라이언트 간의 통신 정보, 에러정보 등을 "sqlnet.log"파일과 "sqlnet.trc" 파일에 각각 저장한다. 여기에는 서버 버전 및 클라이언트와의 통신 정보를 기록한 추적결과 정보와 오류메시지가 포함되어 있는데, 해당 파일이 외부로 유출될 경우 공격자는 서버의 통신 정보를 얻어 향후 기밀정보 유출과 같은 2차 공격에 대한 위험이 존재한다.

◎ 조치방안

* sqlnet.log 파일과 sqlnet.trc 파일이 유출될 경우 Trace Assistant를 사용하여 파일정보가 해석될 위험이 존재하므로 추적기능 해제 및 접근권한 설정 권장
* sqlnet.ora 구성파일에서 clinet와 server의 추적기능 해제
  · 설정파일 경로: $ORACLE_HOME/network/admin/sqlnet.ora
  · 추적파일 경로: $ORACLE_HOME/network/trace/sqlnet.trc

```
TRACE_LEVEL_CLIENT = OFF
TRACE_LEVEL_SERVER = OFF
```

[2-3] Macromedia Dreamweaver 원격 데이터베이스 스크립트 정보 유출

◎ 개요

Macromedia의 Dreamweaver는 웹 개발 도구로 일반적으로 데이터베이스에 연결 테스트를 지원하기 위해 특정 테스트 스크립트가 업로드 된다. 이 스크립트가 서버에 남아있을 경우 공격자는 해당 스크립트를 악용하여 사용자 이름과 비밀번호를 입력하지 않고도 백엔드 데이터베이스 서버에 접근할 수 있게 된다.

◎ 조치방안

* DB 접근정보를 가지고 있는 MMHTTPDB 스크립트 파일 및 연결 스크립트 제거
  · Dreamweaver 파일 브라우저는 MMHTTPDB 스크립트 파일이 포함된 _mmServerScripts 디렉터리를 숨기기 때문에, 타사 FTP 클라이언트 또는 파일 브라우저 사용을 통해 _mmServerScripts 디렉터리를 볼 수 있다.

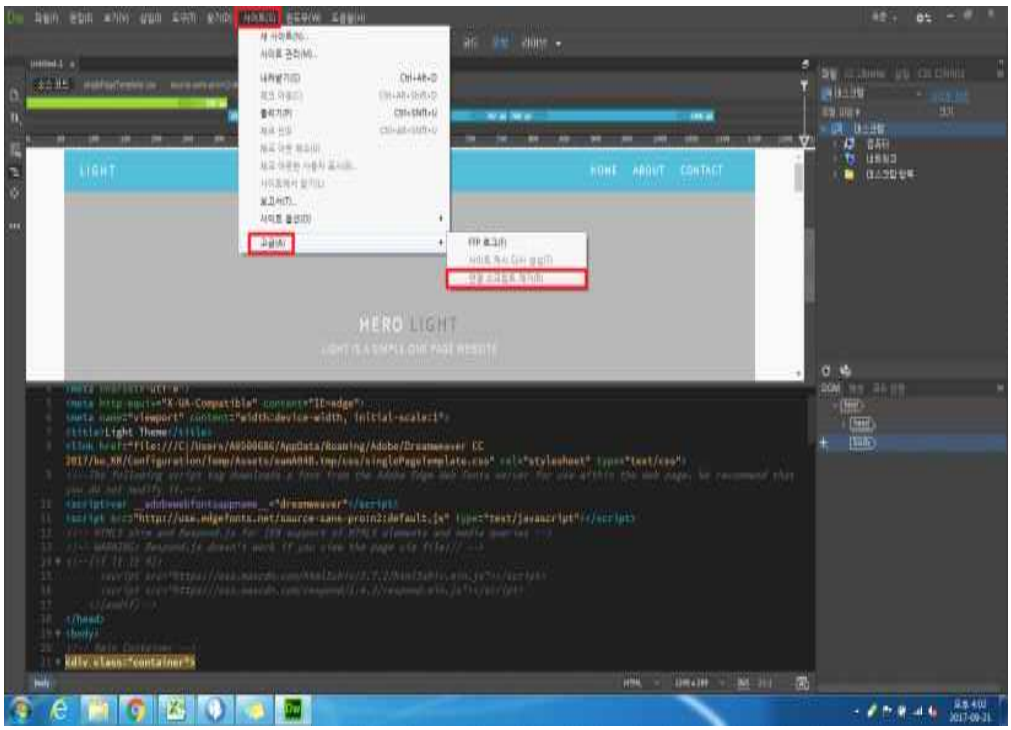
```php
<?php

if(extension_loaded("mbstring"))
{
        $acceptCharsetHeader = "Accept-Charset: " . mb_internal_encoding();
        header( $acceptCharsetHeader );
        $head = "<html><head><meta http-equiv='Content-Type' content='text/html; charset=" . mb_http_output() . "'></head>";
        echo( $head );
}

// Build connection object
//if ($connType == "MYSQL")
if ($_POST['Type'] == "MYSQL")
{
        require("./mysql.php");
        $oConn = new MySqlConnection($_POST['ConnectionString'], $_POST['Timeout'], $_POST['Host'], $_POST['Database'], $_POST['UserName'], $_POST['Password']);
}

// Process opCode
if ($oConn)
{
        $oConn->Open();

        if ($_POST['opCode'] == "IsOpen")
                echo($oConn->TestOpen());
        elseif ($oConn->connectionId && $oConn->isOpen)
```

※ DB 연동시 최초 입력되는 DB 계정 및 비밀정보 정보가
MMHTTPDB(php or asp) 스크립트 내에 자동으로 삽입되어 정보를 저장

<MMHTTPDB.php 스크립트 소스>

· 구성상의 불필요한 MMHTTPDB 스크립트 파일을 프로덕션 서버에 업로드 한 경우 MMHTTPDB 스크립트 파일을 삭제해야 하며, "Remove Connection Scripts" 명령을 통해 자동으로 스크립트 파일을 제거할 수 있다.

※ 조치경로 : menu → site → Advanced (고급) → Remove Connection scripts (연결 스크립트 제거)

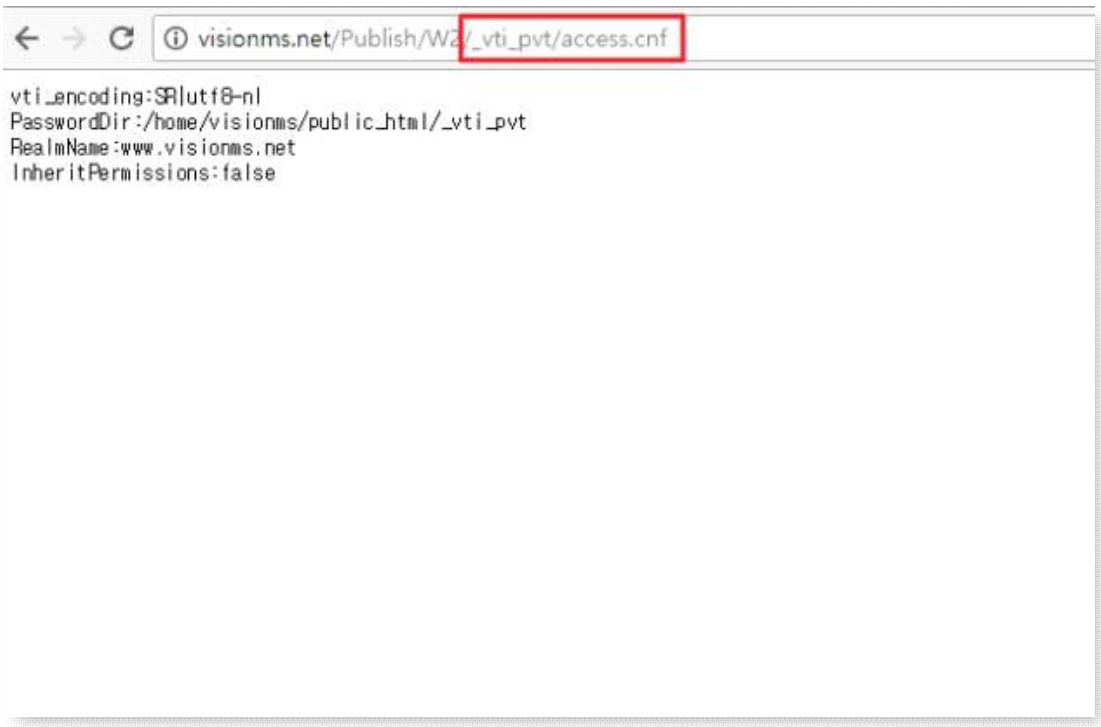


<연결 스트립트 자동삭제>

[2-4] Microsoft FrontPage Server Extensions 필수 정보 유출

◎ 개요

Frontpage는 Microsoft Office에서 기본으로 제공하는 웹 에디터로, Frontpage 서버 확장기능을 위한 다양한 파일이 설치되는데, 해당파일 중 일부는 사용자 이름과 비밀번호와 같은 민감한 정보를 포함하고 있다. 이러한 파일에 잘못된 접근제어 정책이 적용될 경우 비인가자에 의해 시스템 내부정보 탈취를 통해 2차 공격에 악용할 수 있는 위험성이 존재할 수 있다.



<직접 노출되는 access.cnf 파일 정보>

◎ 조치방안

* 기본설치 시 제공되는 중요파일에 대한 올바른 접근권한 설정

· 필요하지 않다면 Frontpage extentions 즉시 제거

· Frontpage 확장자 파일에 대한 접근권한 수정 권장

· FrontPage Server extentions의 최신 버전으로 업그레이드

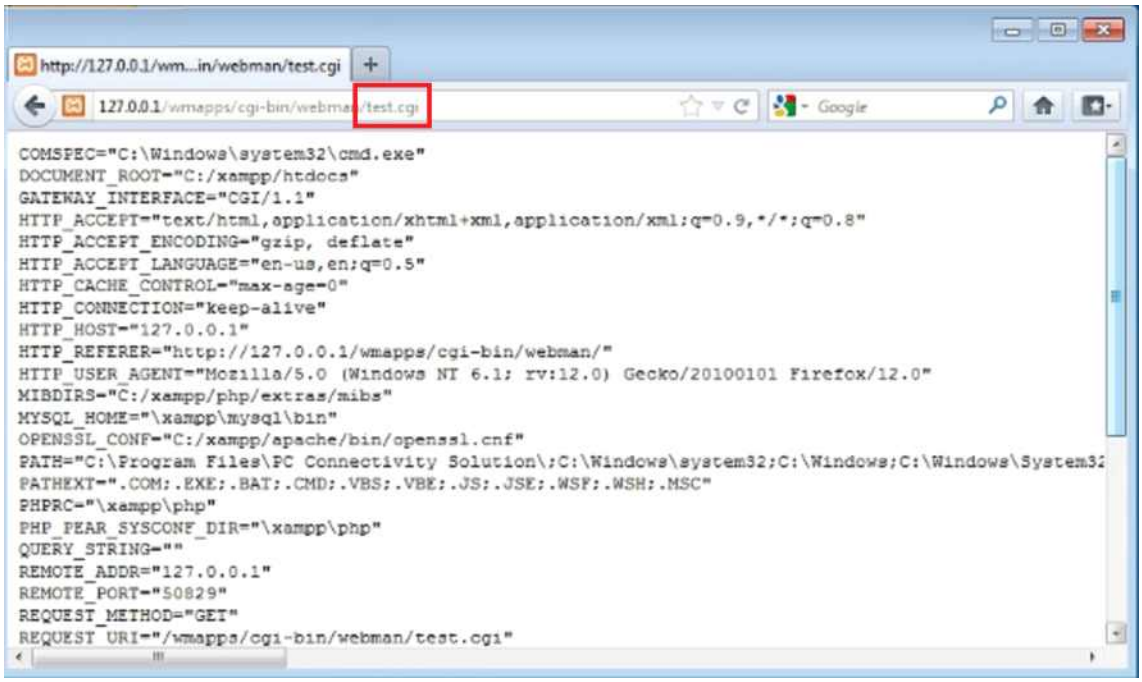
Frontpage 기본설치 파일용도 설명

확장자 파일	파일 용도
Access.cnf	HTTP 서버 특성적인 액세스 제어 정보가 저장
Administrators.pwd	암호화된 관리자 이름과 암호가 저장되며, Netscape 서버에만 사용
Authors.pwd	암호화된 제작자 이름과 암호가 저장되며, Netscape 서버에만 사용
Bots.cnf	FrontPage SDK를 사용하여 만들어진 사용자 정의 FrontPage기반 구성 요소에 사용
Botinfs.cnf	사용자 정의 FrontPage 기반 구성 요소(WebBots) 정보가 저장
Deptodoc.btr	웹 종속 데이터베이스
Doctodep.btr	웹 종속 데이터베이스
Frontpg.lck	웹 서버 리소스가 동시에 액세스되지 않도록하는 잠금 파일이며, 정지 시간에는 파일 크기가 0바이트가 되어야 함. C:₩ProgramFiles₩Microsoft FrontPage₩40₩Temp에 존재함
Linkinfo.cnf	현재 웹에 있지 않은 URL의 역 링크
Service.cnf	웹에 대한 메타 정보가 저장
Service.grp	Administrator 및 Author 그룹의 구성원을 나열하며, CERN 및 NCSA 서버에만 사용
Service.pwd	암호화된 암호 파일이 포함됩니다. IIS 및 WebSite 서버에는 사용되지 않음
Service.stp	FrontPage 확장웹의 Service.pwd 및 Service.grp 파일의 절대경로 저장 CERN 및 NCSA서버에만사용
Services.cnf	하위 웹 목록이 저장
Services.org	서버 익스텐션을 다시 설치할 때 하위 웹이 루트 웹의 하위 디렉터리로 생성되지 않도록 Services.cnf 파일의 기초로 사용
Service.lck	웹 서버 리소스가 동시에 액세스되지 않도록하는 잠금 파일. 유휴 시간에는 0 바이트가 되어야 함
Structure.cnf	FrontPage 확장 웹 탐색 정보 영역이 포함
Svcacl.cnf	하위 웹이 고유한 사용 권한 설정을 갖는지 여부와 IP 주소 제한에 대한 정보를 저장하여, 서버 익스텐션을 다시 설치할 때 하위 웹에 올바른 사용 권한 설정이 다시 적용되도록 함
Users.pwd	암호화된 최종 사용자 이름과 암호를 Netscape 서버에만 사용
Uniqueperm.cnf	이 파일의 존재는 서브웹이 고유한 퍼미션 설정을 가짐을 말해주며 퍼미션에 반대될 때는 루트 웹으로부터 인계 받음
Writeto.cnf	결과 저장 양식 처리기 파일처럼 웹 사용자에 의해 작성되는 파일의 역 링크. 웹 사용자에 의해 작성될 수 있는 파일은 일반 웹 컨텐트보다 보안 설정이 낮음

[2-5] 다중 벤더 test-cgi 중요 정보 유출

◎ 개요

Test-cgi는 주로 Apache 서버에서 활용하는 일반적인 샘플 및 디버깅 스크립트이다. 이 스크립트는 입력 변수에 대한 취약한 설정으로 로컬 파일 이름 또는 환경 변수를 노출시켜 보안상 위험성을 나타낼 수 있으며, 이를 악용한 공격자는 원격을 통한 내부 디렉터리 목록 및 내부정보를 탈취할 수 있다.



<웹 브라우저를 통해 노출되는 Test-cgi 환경변수>

◎ 조치방안

* 다중 벤더에 포함되어 있는 test-cgi 스크립트 실행권한을 제거 및 수정

· test-cgi 스크립트를 서버에서 제거하거나 사용 불가능하도록 실행 권한 제거

· 반영되는 변수 주위에 따옴표(")를 배치하여 스크립트를 수정

스크립트 수정 전

```
echo QUERY_STRING = $QUERY_STRING
```

스크립트 수정 후

```
echo QUERY_STRING = “$QUERY_STRING”
```

[2-6] Oracle Web Listener 원격 명령 실행

◎ 개요

Oracle의 Web Listener(Oracle Application Server의 구성요소)를 설치한 후 디폴트 셋팅을 그대로 사용할 경우 공격자가 원격에서 Oracle 소유의 UNIX 계정으로 시스템에 접근 후 임의의 명령을 수행할 수 있는 취약점이 존재한다. Oracle Web Listener에 있는 "ows-bin" 가상 디렉터리에 포함된 상당수의 batch 파일, DLL, 그리고 실행파일들은 임의의 명령을 실행하는 취약점을 가지고 있는데, 공격자는 "?, &"와 같은 접두문자를 활용하여 임의의 쉘 명령을 실행할 수 있다.

공격 예시

```
GET /ows-bin/perlidlc.bat?&dir HTTP/1.0
```

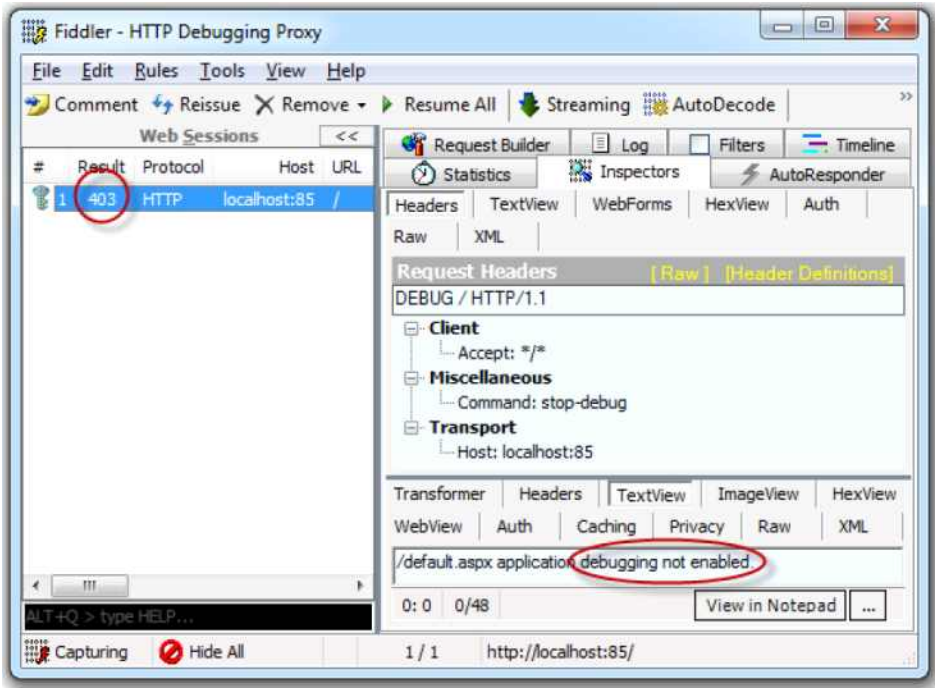
◎ 조치방안

* 원격실행 방지를 위해 ows-bin 가상(virtual) 디렉터리 제거
  · OSA(Oracle Application Server)의 기본설치 경로 : C:₩orant₩ows₩4.0₩bin

[2-7] Microsoft ASP.NET 디버깅 사용 가능

◎ 개요

Microsoft ASP.NET은 동적인 웹 사이트, 웹 애플리케이션, 웹 서비스를 제작하기 위한 웹 애플리케이션 프레임워크이다. 이 중 디버그 모드는 프로그램 실행을 제어하는 것이 가능한 개발자를 위한 모드로 사용되는데, 디버그 모드가 활성화 되어 있을 경우 공격자는 디버그모드 사용 유무 확인이 가능하다. 이를 악용하여 ASP로 제작된 악성 스크립트를 통해 서버에 악성명령을 보내고 디버깅 작업으로 서버 리소스를 소비시켜 스택오버플로우와 같은 공격을 발생시킬 수 있다.



<공격 도구를 사용하여 디버그모드 사용유무 확인>

◎ 조치방안

* ASP.NET 디버그 모드가 활성화 되어 있을 경우 악성명령에 취약하므로 비활성화 설정 필요하다. ASP.NET의 디버깅을 사용 불가능하게 하려면 web.config 또는 Machine.config 파일의 디버깅 값을 true에서 false로 변경

* web.config 또는 Machine.config 파일 Debug 설정 수정
  - %SystemRoot%₩Microsoft.NET₩Framework₩%VersionNumber%₩CONFIG₩web.config
  - %SystemRoot%₩Microsoft.NET₩Framework₩%VersionNumber%₩CONFIG₩Machine.config

스크립트 수정 전

```
<compilation debug="true"/>
```

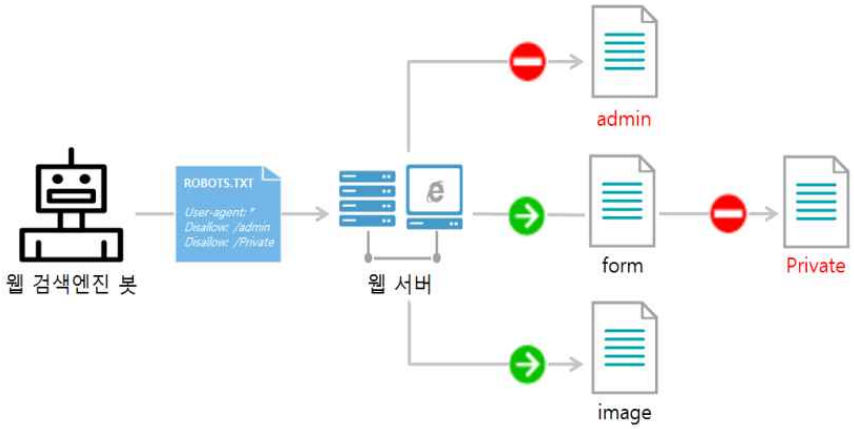
스크립트 수정 후

```
<compilation debug="false"/>
```

[2-8] Robots.txt 파일 웹 사이트 구조 노출

◎ 개요

robots.txt 파일은 웹 사이트에 정보수집 로봇이 접근하는 것을 방지하기 위한 규약으로 접근제한에 대한 정책을 기술한다. 하지만 검색을 허용하거나 불허하는 정책이 잘못되어 있을 경우 관리자페이지 혹은 개인정보 페이지 등 웹 서비스 해킹에 필요한 민감정보가 검색되어 해킹에 악용될 수 있는 정보가 유출될 수 있다.



<Robots.txt 정상적인 탐색경로>

◎ 조치방안

* "robots.txt" 파일은 정보를 보호하거나 숨기기 위한 목적으로 사용 권장하지 않음
* 웹 robot 검색에서 해당 디렉터리를 배제하기 위해서 중요한 파일이나 디렉터리를 다른 분리된 서브-디렉터리로 이동

설정 예시 특정 파일을 'folder'와 같은 특별한 의미를 갖지 않는 디렉터리로 이동

```
/folder/passwords.txt
/folder/sensitive_folder/
New robots.txt:
User-agent: *
Disallow: /folder/
```

* 디렉토리 구조를 변경할 수 없다면, "robot.txt" 파일에서 그 이름의 일부만을 사용하여, 공격자로 하여금 전체 파일 이름이나 디렉토리 이름을 유추하기 어렵게 설정

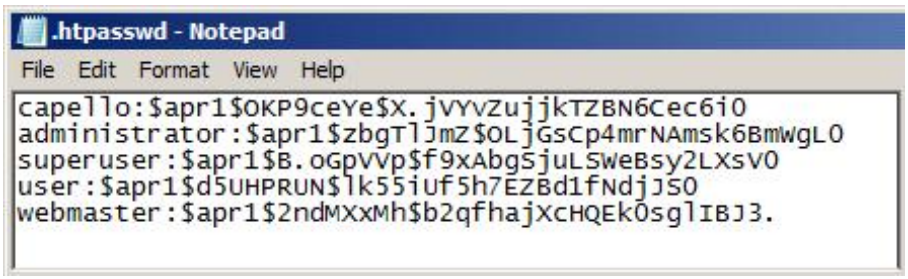
설정 예시 'sensitive_folder'과 'passwords.txt' 파일에 대한 robot 검색을 배제하기 위한 방법

```
robots.txt:
User-agent: *
Disallow: /se
Disallow: /pa
```

[2-9] 웹 서버 액세스 제어 파일 올바르지 않은 사용 권한 설정

◎ 개요

일부 웹 서버는 접근제어 설정의 세분화를 지원하기 위해 3개의 부분 설정 파일(.htaccess, .htpasswd, .htgroup)을 사용하는데, 이 파일에 대해 잘못된 접근제어 정책이 적용될 경우 비인가자에 의해 사용자 이름, 관리자 비밀번호, 머신이름 및 민감한 파일 위치 등과 같이 웹 애플리케이션에 대한 민감한 정보가 유출될 위험성이 존재한다.



<.htpasswd 파일의 노출된 사용자 중요 정보>

◎ 조치방안

* .htaccess 파일에 접근제어 설정

설정 예시 'sensitive_folder'과 'passwords.txt' 파일에 대한 robot 검색을 배제하기 위한 방법

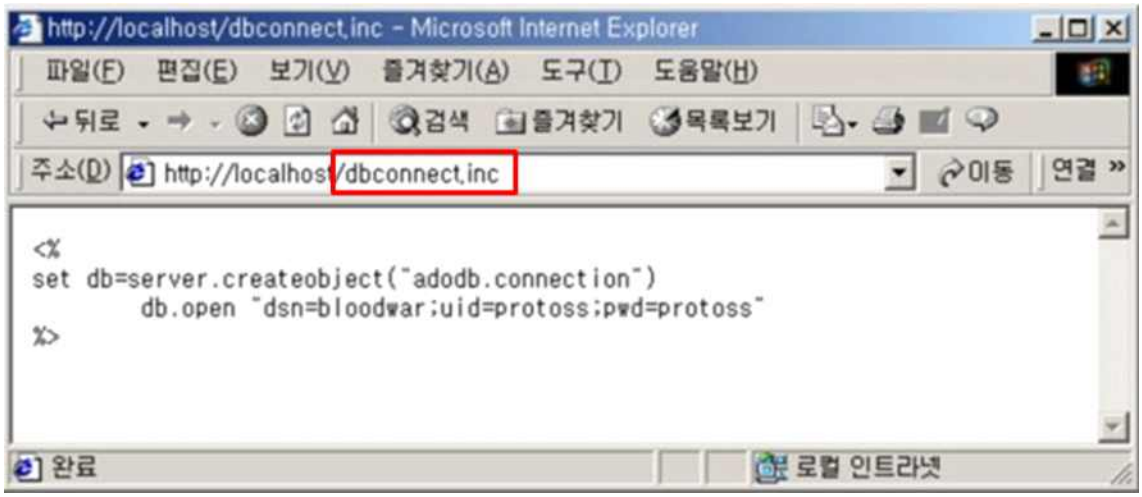
```
AuthName "인증이 필요한 관리자 페이지 입니다."
AuthType Basic
AuthUserFile /home/www/admin/.htpasswd
AuthGroupFile /dev/null
require valid-user
Order deny, allow
Deny from all
Allow from 10.10.100.7 10.10.2.1/24
```

* ".htaccess"와 ".htgroup" 파일에 대해 적절한 액세스 권한을 설정
* .htpasswd에 의해 관리되는 민감한 웹 비밀번호 파일은 외부노출을 금지해야 하며 비인가자에 의한 비정상적인 접근을 차단하기 위해 접근제어 설정 필요

[2-10] PCCS MySQL Database Admin Tool 관리자 비밀번호 유출

◎ 개요

PCCS-MySQL Database Admin Tool은 PHP로 작성된 MySQL 관리 프로그램이다. 이 프로그램은 서버 확장기능을 위한 다양한 파일이 설치되는데, 특히 관리자 비밀번호가 평문으로 저장되는 dbconnect.inc 파일은 악의적인 사용자의 GET 요청 조작으로 인해 외부로 파일 다운로드가 가능하다. 이러한 민감파일이 공격자에게 유출될 경우 DB접근을 통한 기밀정보 유출, 불법 다운로드, 삭제 등 다양한 공격에 악용될 수 있다.



<웹 브라우저를 통한 dbconnect.inc 노출 예시>

◎ 조치방안

* PCCS-MySQL database Admin Tool 최신 버전으로 업그레이드
* pccsmysqladm 루트 디렉터리에 .htaccess 파일을 생성 후 .inc 파일에 대한 액세스 설정

.htaccess

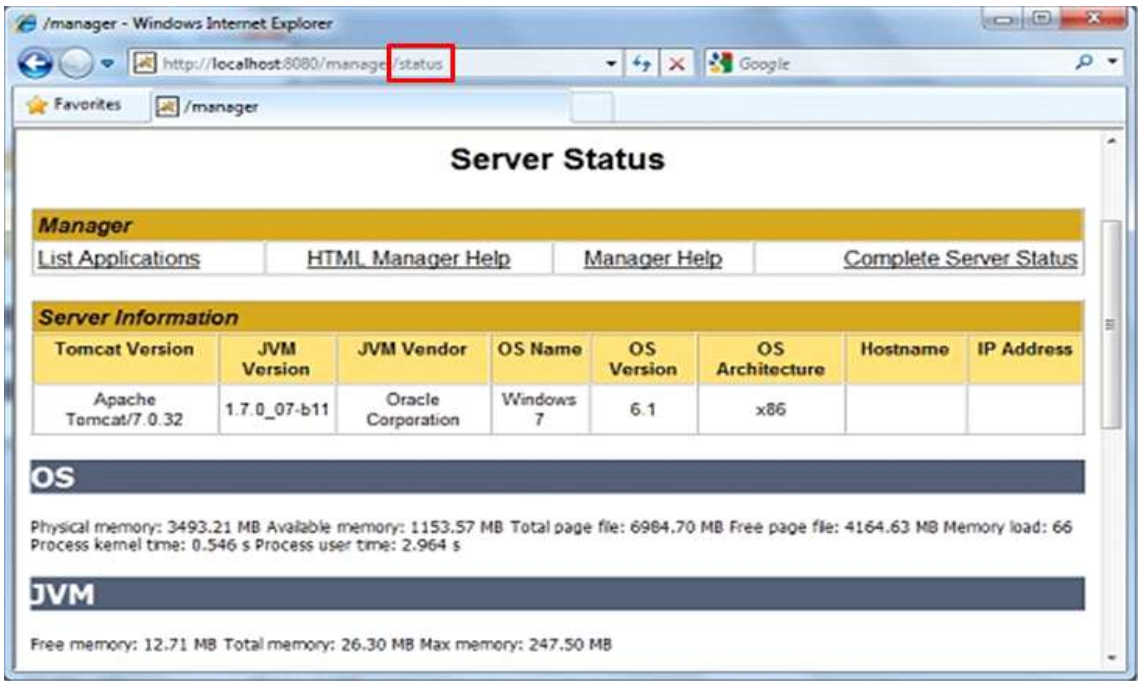
```
<Files ~ "₩..inc$">
Order allow, deny
Deny from all
</Files>
```

* 비밀번호 정보가 평문으로 저장되는 dbconnect.inc파일에 대하여 보안조치 수행 또는 접근제한 설정

[2-11] Apache server-status 정보 유출

◎ 개요

Apache 웹 서버에서 mod_status 모듈이 httpd.conf 파일에 설정되면, 현재 작동중인 서버의 상태정보를 보여주는 "server-status" HTML 페이지가 제공된다. 이러한 페이지에 대한 접근제어가 적절하게 설정되어 있지 않은 경우 공격자는 웹 브라우저를 통해 외부로 노출된 HTML페이지에서 서버 구성 및 버전정보를 획득하고, 사이트에 대한 추가적인 공격을 수행할 수 있다.



<서버정보를 보여주는 server-status>

◎ 조치방안

* Apache 웹 서버의 동작상태를 알 수 있는 server-status html 페이지가 노출되지 않도록 httpd.conf 파일을 통해 접근 제어 설정

Apache

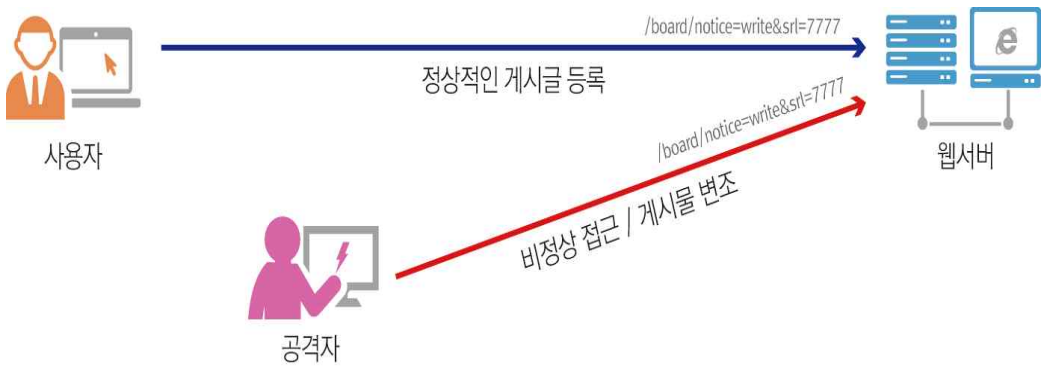
httpd.conf 파일 설정

```
#<Location /server-status>
# SetHandler server-status
# Order deny,allow
# Deny from all
# Allow from .your_domain.com
#</Location>
```

제3절 권한인증 취약점

취약점 설명

일반적으로 홈페이지 운영 시 사용자들의 접근제어를 목적으로 관리자와 일반 사용자에 대한 권한을 별도로 분리하여 관리하는데 권한관리 정책이 미흡할 경우 비인가자도 취약한 URL 매개변수 조작을 통해 권한이 없는 영역의 페이지에 접근할 수 있다. 특히, 관리자 권한으로 접근이 가능한 공지사항 글쓰기 메뉴에 허가되지 않은 사용자가 접근하여 임의의 글쓰기가 가능할 경우 홈페이지 관리자로 위장하여 악성코드 배포, 홈페이지 위·변조가 가능하다.



<경로조작을 통한 시스템 내부파일 다운로드>

사전예방조치 방안

* 타 사용자가 게시한 글에 대한 접근제어 설정을 권장
* 게시글 작성 및 수정, 삭제 시 비밀번호 설정을 이용한 접근제어 수행
* 관리자 권한의 모든 페이지에 구현된 Cilent Side Script 형식의 검증
* 홈페이지에서 로그인을 한 사용자의 경우 세션인증을 통하여 해당 글에 대한 수정, 삭제 권인 검증 필요
* Secure Coding

Tomcat

· web.xml 파일에서 아래와 같이 설정

```
<session-config>
<session-timeout>10</session-timeout> #분 단위
</session-config>
```

Apache

· /etc/httpd/conf/httpd.conf 파일에서 timeout 옵션에 아래와 같이 설정

```
Timeout 300 //기본 값 300초
```

JAVA

· 세션 타임아웃 설정

```
public void doGet(HttpServletRequest request, HttpServletResponse response)throws
IOException
{
.....
PrintWriter out = response.getWriter();
HttpSession session = request.getSession();
session.setAttribute("foo", "42");
session.setMaxInactiveInterval(600); //초단위
}
```

JAVA/JSP

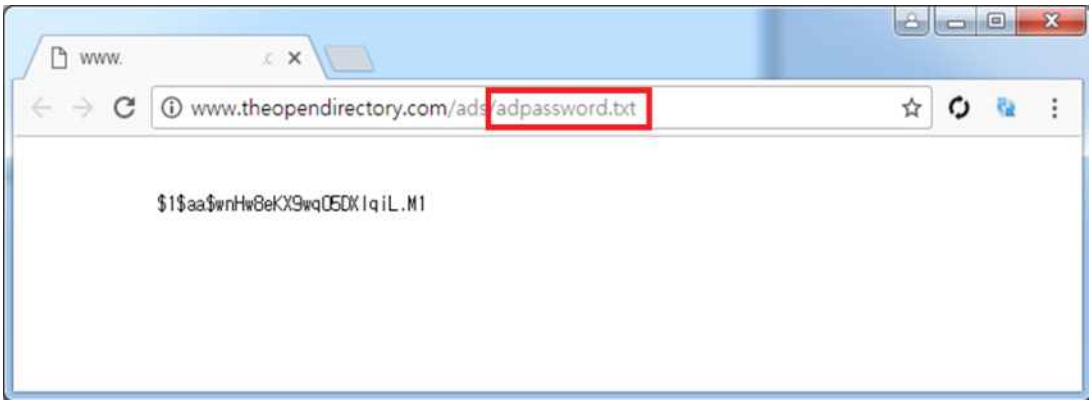
· 세션 파기 구현

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException
{
 request.getSession().invalidate();
 //세션 파기 invalidate Method 호출로 세션 파기됨.
response.sendRedirect(request.getContextPath() + "/Login.html");
 }
```

[3-1] Banner Rotating 01 권한 상승

◎ 개요

"Banner rotating 01"은 배너를 제작할 수 있는 프리웨어 CGI 스크립트로, 기본 구성되는 파일 중 암호정보를 담고 있는 adpassword.txt이 존재 하는데, 이 암호 파일에 대하여 접근을 취약하게 설정하였을 경우, 공격자는 파일의 내용을 읽을 수 있으며 DES 암호 크래커 응용 프로그램을 사용하여 콘텐츠를 검색하고 오프라인으로 해독 할 수 있다.



<adpssword.txt 경로를 통한 암호화된 구성정보 노출 예시>

◎ 조치방안

* 비인가자로부터 adpassword.txt에 접근할 수 없도록 .htaccess 파일 접근권한 수정

Apache

.htaccess 파일 설정

```
<Files "adpassword.txt">
Order allow, deny
Deny from all
</Files>
```

제4장 결론

‘17년도 3분기에 웹 어플리케이션에 대한 블랙박스 및 화이트박스 테스트 결과 주로 웹서버를 구성하는 제반 소프트웨어에서 설치파일 기본경로 노출, 패킷 요청·응답에 관한 취약한 기본설정, 관리자 기능 접근권한 설정 미흡, 소스정보 유출 등 웹서버를 구성할 때 초기설정 및 설계결함으로 인해 발생하는 취약점이 다수 발견되었다.

최근, 날이 갈수록 사이버 공격 기술이 점차 지능화 되면서 고차원적인 침해 시도가 급증하고 있으나, 대부분의 침해사고는 기존에 이미 알려진 오래된 취약점 사례를 통해 발생한다. 해당 원인은 서비스 편의성 측면에 주안점을 두는 운영자와 개발자의 안일한 대처에서 발생한다고 할 수 있다.

침해사고가 발생하게 되면 취약점을 유발하는 설정 값이나 소스, 더 나아가 제반 소프트웨어의 재구성, 재설계 등 취약점 개선작업에 요구되는 비용과 시간이 요구되는데 이는 웹사이트를 새로 구축하는 정도로 많이 소요될 수 있어 기회비용 측면에서도 비효율적이다.

보안사고 발생 후 패치를 개발하고 대응하는 것은 사후약방문에 불과하다. 따라서 향후에는 홈페이지 구축단계에서부터 이러한 침해사고를 미연에 방지할 수 있도록 심도있는 사전점검 및 선제적 예방조치를 반드시 수행해야 할 것이다.

참고 자료

[1] https://www.koreahtml5.kr/download/WebDev_w3c_html5_technology.pdf

[2] https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Access-Control-Allow-Origin

[3] http://help.adobe.com/ko_KR/ActionScript/3.0_ProgrammingAS3/WS5b3ccc516d4fbf351e63e3d118a9b90204-7c9b.html

[4] https://docs.oracle.com/cd/E11882_01/network.112/e10835/sqlnet.htm#NETRF250

[5] https://support.microsoft.com/ko-kr/help/815157/how-to-disable-debugging-for-asp-net-applications

[6] Redhat, https://access.redhat.com/solutions/1232233

[7] MS, https://msdn.microsoft.com/en-us/library/aa479501.aspx

[8] Adobe, http://help.adobe.com/ko_KR/ActionScript/3.0_ProgrammingAS3/WS5b3ccc516d4fbf351e63e3d118a9b90204-7e08.html