ISBN XXX-XX-XXX-XXXX-X-XXXXX

웹 어플리케이션 취약점 조치방법 (Ⅰ)

2016. 11

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

웹 어플리케이션 취약점 조치방법 (Ⅰ)

2016. 11

부 서 : 첨단연구망센터 첨단연구망정보보호실
제출자 : 이행곤 (hglee@kisti.re.kr)
최장원 (jwchoi@kisti.re.kr)
정용환 (paul7931@kisti.re.kr)
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 구조적인 취약한 특성으로 인해 악의적인 공격자에 의한 공격 타겟이 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 문제해소를 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 실제 운영되고 있는 웹사이트 환경에서 블랙박스 테스트를 통해 주로 탐지되는 취약점 패턴을 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.

1. 웹 취약점 주요 탐지 유형

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 전체 17개 취약점 유형 중 7개의 유형만 다루도록 한다.

순번	취약점 유형	주요 탐지패턴
1	관리자 페이지 노출 취약점	[1-1] 관리자 페이지 노출
2	디렉터리 나열 취약점	[2-1] 디렉토리 목록화 패턴 발견
		[2-2] Microsoft FrontPage 디렉토리 목록화
		[2-3] Microsoft FrontPage '_vti_cnf' 정보유출
3	시스템 관리 취약점	[3-1] Apache Multivies Attack
		[3-2] HTTP Strict-Transport-Security 헤더 누락
		[3-3] 캐시화 가능한 SSL 페이지 발견
		[3-4] 세션 쿠키에서 HttpOnly 속성 누락
		[3-5] 암호화된 세션(SSL) 쿠키의 누락된 보안 속성
		[3-6] HTTP.sys 원격 코드 실행
		[3-7] 영구적 쿠키에 중요 세션 정보 포함
		[3-8] Microsoft FrontPage Extensions사이트 손상
4	WEBDAV 취약점	[4-1] WebDAV 취약점
5	불필요한 Method 허용 취약점	[5-1] TRACE 및 TRACK HTTP 메소드 사용
		[5-2] 안전하지 않은 HTTP 메소드 사용
6	취약한 파일 존재 취약점	[6-1] 파일 대체 버전, 애플리케이션 테스트 스크립트 발견
		[6-2] CMME 정보 유출
		[6-3] URL 경로 재지정을 통한 피싱
		[6-4] PHP phpinfo.php 정보 유출
		[6-5] 임시파일 및 아카이브 파일 다운로드
7	계정 관리 취약점	[7-1] 올바르지 않은 계정 잠금

제3장 웹 취약점 유형 별 상세 조치방안

제1절 관리자 페이지 노출

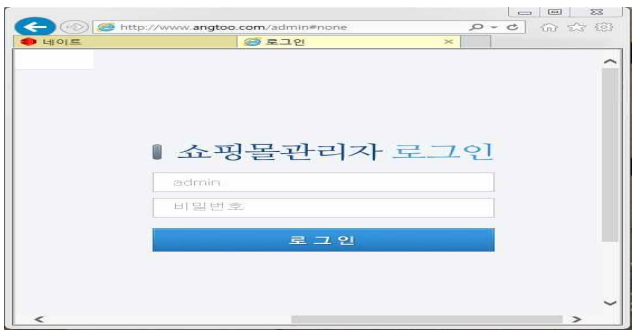
일반적으로 추측하기 쉬운 URL(ex: /admin, /manager)을 사용하고 있어, ID/패스워드에 대한 크랙 또는 접근 허가 정책에 대해 요청하는 부분의 정보를 변경함으로써 접근이 가능한 경우가 존재하는데, 웹 관리자의 권한이 노출될 경우 홈페이지의 위·변조 뿐만 아니라 취약점 정도에 따라서 웹 서버의 권한까지도 노출될 위험성이 존재함

<관리자 페이지 노출을 통한 비인가자 접근>

[1-1] 관리자 페이지 노출

◎ 개요

웹 서비스 관리자 페이지 경로가 외부로 노출되는 취약점으로, URL 주소창에 추측 가능한 페이지 주소(예시 : admin, adm, cms 등)로 관리자 페이지가 구성될 경우 관리자 계정 탈취로 인한 권한상승의 위험이 있음



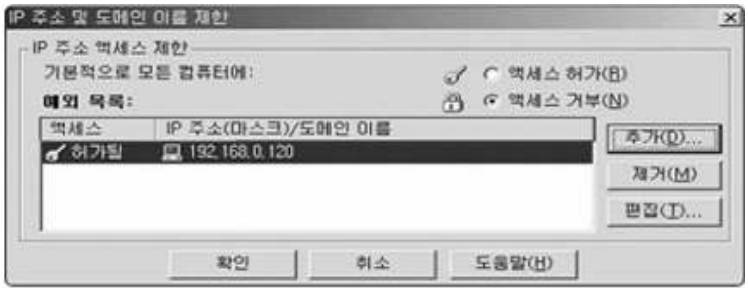
<관리자 전용 로그인 페이지>

◎ 조치방안

1)웹 서버 내에서의 특정 IP주소에서만 접근 허용

① 윈도우 IIS에서 관리자 IP 설정방법

- [설정]→[제어판]→[관리도구]→[인터넷 서비스 관리자]→[인터넷정보서비스]→[관리자 디렉터리 선택 후 마우스 우 클릭]→[등록정보]→[디렉터리 보안]→[IP주소 및 도메인 이름제한]→[편집]



<IIS 관리자 IP 설정>

② 리눅스 및 유닉스의 "아파치(Apache)" 환경 설정방법

- 아파치 웹 서버의 설정 파일 httpd.conf → "Directory"내의 AllowOverride 옵션에서 AutoConfig 또는 All을 추가하여 관리자 IP만 접근 가능하도록 제한

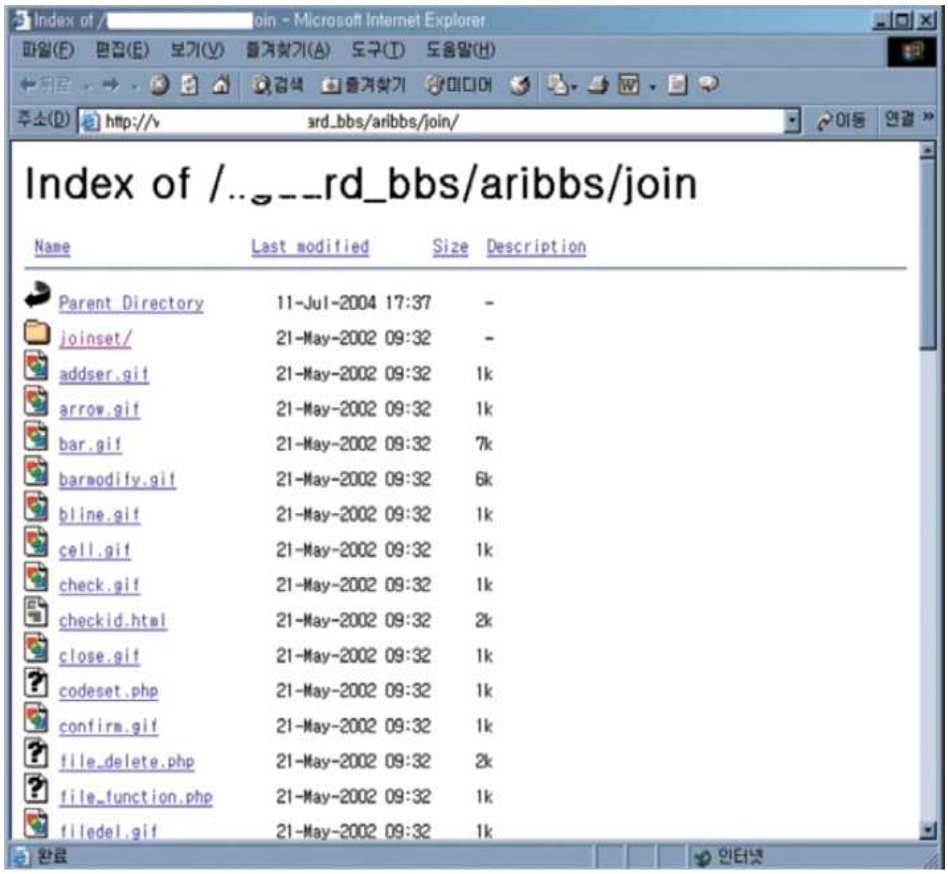
예시) 관리자 폴더 /usr/local/www/admin에 192.168.1.1만 접근할 수 있도록 설정한 경우

```
〈Directory /home/www/admin/〉
    AllowOverride AuthConfig (또는 All)
    Order deny, allow
    Deny from all
   Allow from 10.10.100.7 10.10.2.1/24
〈/Directory〉
# 먼저 접근을 제어하고자 하는 디렉토리에 대한 상위 디렉토리 정의에
# AllowOverride 부분이 'All', 'AuthConfig', 'FileInfo' 등으로 설정되어 있어야 한다.
〈Directory "접근을 제어하고자 하는 디렉토리"〉
    ......
    AllowOverride FileInfo AuthConfig Limit
    ......
〈/Directory〉
......
AccessFileName .htaccess
〈Files ~ "^\ht"〉
    Order allow, deny
    Deny from all
〈/Files〉

〈.htaccess〉
    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
```

제2절 디렉터리 나열 취약점

디렉터리 나열 취약점 유형은 웹 브라우저에서 URL 입력란에 파일명 이하를 삭제하고 바로 디렉토리 경로로 접근을 시도하였을 경우 디렉토리의 하위 내용이 나열되는 취약점으로, 공격자는 내부에 적재된 파일정보와 구성 정보 획득을 통해 웹 어플리케이션의 구조 파악을 통한 민감정보 유출의 위험성이 존재



<디렉터리 구조 노출>

[2-1] 디렉토리 목록화 패턴 발견

◎ 개요

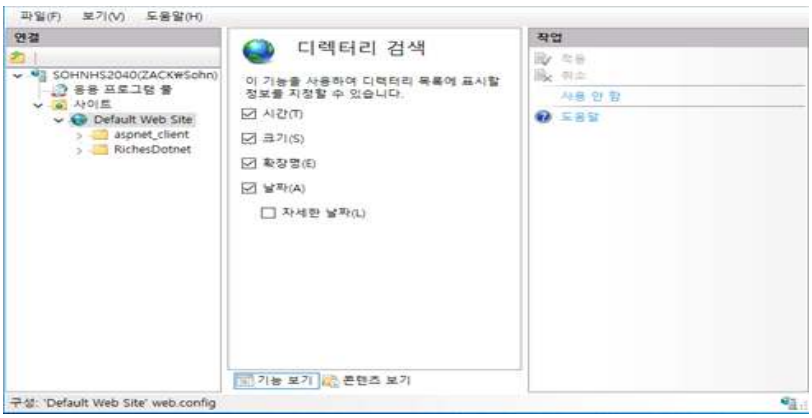
디렉터리 인덱싱 기능이 활성화되어 있을 경우, 비인가자가 외부에서 웹 서버 내 모든 디렉토리 및 파일에 접근이 가능하여 어플리케이션 및 서버의 중요정보 노출로 인한 추가공격에 악용될 수 있는 취약점

◎ 조치방안

1) Windows

[IIS]

제어판 > 관리도구 > IIS(인터넷 정보 서비스)관리자 매뉴에서 구축사이트 선택 > 화면 중앙 IIS 박스의 디렉터리검색 선택 사용안함 클릭



<IIS 디렉터리 검색기능 비활성화>

2) Apache

아파치(Apache)를 웹서버로 사용하는 리눅스 및 유닉스 OS는 아래와 같은 설정을 통해 디렉토리 리스팅 취약점을 차단 가능

```
<Directory "/var/www/html"> // 디렉토리 경로
    Options Indexes FollowSymLinks // 인덱스 활성
    → Options FollowSymLinks // 인덱스 비활성
```

3) Tomcat

톰캣(Tomcat)을 웹서버로 사용하는 리눅스 및 유닉스 OS는 아래와 같은 설정을 통해 디렉토리 리스팅 취약점을 차단 가능

- web.xml 파일 설정 예시

```
<init-param>
    <param-name>listings</param-name>
    <param-value>false</param-value>
</init-param>
```

※ 서비스 설정 후 데몬 재시작 필수

[2-2] Microsoft FrontPage 디렉토리 목록화

◎ 개요

Frontpage는 MicroSoft office에서 기본적으로 제공하는 웹 에디터로, 서버 관리기능(sercer extention) 보안설정이 올바르지 않을 경우, 가상 디렉토리 및 파일에 접근이 가능하여 어플리케이션 및 서버의 중요정보 노출로 인한 추가 공격에 악용될 수 있는 취약점

◎ 조치방안

- 불필요할 경우 Frontpage 서버관리 기능(Extensions) 제거
- C: ₩Program Files ₩Common Files ₩Microsoft Shared ₩Web server Extension 서브 디렉터리에서 아래의 디렉토리 삭제

₩isapi
₩_vit_bin₩_vti_adm
₩_vit_bin₩_vti_aut
₩_vit_bin
₩admisapi
₩admincgi

[2-3] Microsoft FrontPage '_vti_cnf' 정보유출

◎ 개요

Frontpage는 MicroSoft office에서 기본적으로 제공하는 웹 에디터로, 서버관리 기능(server-extention) 보안설정이 올바르지 않을 경우, 비인가자가 내부 디렉터리 정보나 파일경로 등 내부정보를 가지고 있는 "_vti_cnf" 폴더로 접근가능한 취약점

group.chunjae.co.kr - /images/login/_vti_cnf/

[To Parent Directory]

2005-05-20 오전 3:45	335	b_login_p.gif
2007-03-25 오전 12:00	466	bg-botton.gif
2007-03-25 오전 12:00	471	bg-r.gif
2005-05-20 오전 3:45	300	bg_green.gif
2007-03-25 오전 12:00	471	bg_right.jpg
2007-03-25 오전 12:00	466	checkbox.gif
2005-05-20 오전 3:45	301	copyright_green.gif
2005-05-20 오전 3:45	298	dot_white.gif
2005-05-20 오전 3:45	304	girl.gif
2007-03-25 오전 12:00	481	go.gif
2005-05-20 오전 3:45	335	id_p.gif
2005-05-20 오전 3:45	339	login1_p.jpg
2005-05-20 오전 3:45	339	login2_p.jpg
2005-05-20 오전 3:45	302	login_green.gif
2005-05-20 오전 3:45	302	logo_green.gif
2006-06-04 오전 12:00	448	main.gif
2007-03-25 오전 12:00	279	main.swf
2005-05-20 오전 3:45	305	main2.gif
2005-05-20 오전 3:45	305	main3.gif
2005-05-20 오전 3:45	301	more_orange.gif
2005-05-20 오전 3:45	301	notice_green.gif
2007-03-25 오전 12:00	451	over_checkbox.gif
2005-05-20 오전 3:45	335	pw_p.gif
2005-05-20 오전 3:45	303	typo.gif
2005-05-20 오전 3:45	297	underline.gif

<Frontpage 기본경로 유출>

◎ 조치방안

- Microsoft Update 서비스를 최신버전으로 유지하도록 권장
- 불필요할 경우 Frontpage 서버관리 기능(Extensions) 제거

제3절 디렉터리 나열 취약점

[3-1] Apache Multivies Attack

◎ 개요

Apache에서 제공되는 Multiviews는 다중언어지원 옵션으로 웹브라우저 또는 웹문서의 종류에 따라서 가장 적합한 페이지를 보여주는 기능이며, 공격자가 디렉터리에 숨겨진 파일을 찾아 중요 정보를 수집할 수 있는 취약점

◎ 조치방안

- Apache의 구성 파일에서 Multiviews 옵션 기능 제거 여부 확인
  파일경로 : <apache dir>/htdocs/httpd.conf

```
<Directory "/htdocs">
    Options –MultiViews // multiviews 비활성화
    AllowOverride All
    Order allow,deny
    Allow from all
</Directory>
```

- Apache 2.2 이후 버전

```
<FilesMatch ₩.php$> // 파일확장자명
    SetHandler application/x-httpd-php
<FilesMatch>
```

[3-2] HTTP Strict-Transport-Security 헤더 누락

◎ 개요

응용 애플리케이션에서 HTTP Strict Transport Security(HSTS) 헤더가 적용되지 않아 공격자가 HTTPS 스트리핑 공격(HTTPS 요청을 HTTP로 전환)즉, SSL/TLS 연결을 일반 HTTP 연결로 변경하여 민감한 정보가 평문형태로 탈취될 수 있는 취약점

◎ 조치방안

[Windows IIS]

- Windows IIS 경로 : C:₩inetpub₩wwwroot₩web.config
- 아래의 볼드체 구문 추가
- 배포환경에서 설정하기 위해서는 web.Release.config에 아래와 같은 내용 설정

```
<configuration>
    <system.webServer>
        <directoryBrowse enabled="true" />
    </system.webServer>
        <httpProtocol>
            <customHeaders>
             <add name="Strict-Transport-Security" value="max-age=31536000;
includeSubDomains" />
            </customHeaders>
        </httpProtocol>
  </configuration>
```

[Apache]

- Apache 경로 : /htdocs/httpd.conf
- website.conf 및 httpd.conf 설정파일에서 아래와 같은 내용 설정

```
<VirtualHost xxx.xxx.xxx.xxx:443>
    Header always set Strict-Transport-Security "max-age=63072000; includeSubdomains;"
  </VirtualHost> // max-age 값은 밀리세컨드 단위, 해당 설정은 2년임. 즉 2년동안
                    해당설정이 유효함.
```

[3-3] 캐시화 가능한 SSL 페이지 발견

◎ 개요

로그인 시 사용자명, Password 등 민감한 정보들을 보호하기 위해 SSL을 이용하여 암호화를 수행하는데, 사용 중인 브라우저에서 민감한 정보를 캐싱하여 발생하는 취약점

◎ 조치방안

민감한 데이터가 있는 모든 페이지에서 캐싱기능을 사용하지 않도록 설정

[시큐어코딩]

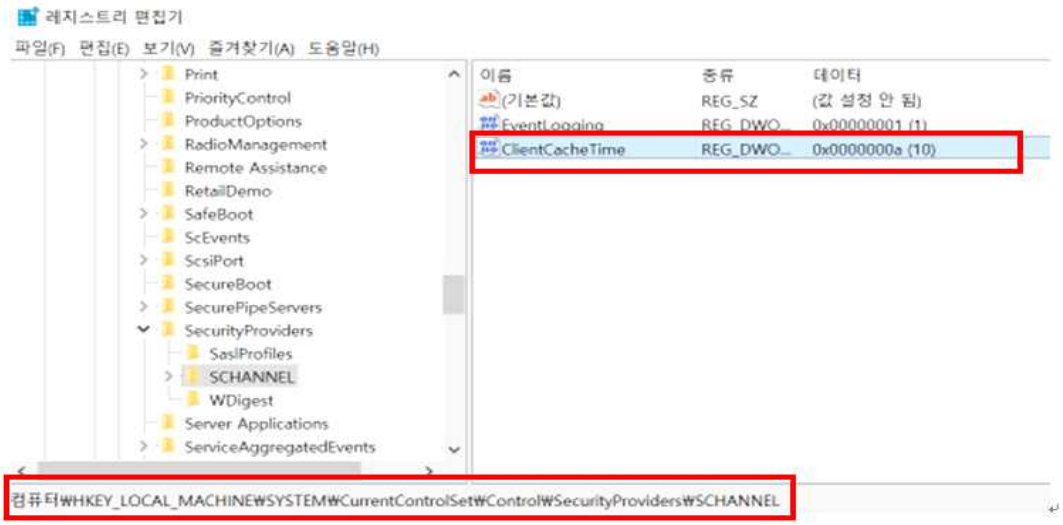
```
response.setHeader("Cache-Control", "no-cache");
response.setHeader("Pragma", "no-store, no-cache, must-revalidate");
```

[WINDOWS]

SSL 세션 정보의 캐시유지 시간 값 수정

예시) ClientCacheTime 설정 방법

십진수 데이터 값 선택 -> 밀리세컨드 단위 (60000 = 1분) -> 비활성화는 0값 처리



<세션 유지 시간 값 수정>

[3-4] 세션 쿠키에서 HttpOnly 속성 누락

◎ 개요

HttpOnly는 악의적인 자바스크립트의 접근을 차단하기 위한 기능으로 HttpOnly 속성이 누락될 경우 XSS(Cross Site Scripting)와 같은 공격자의 변조된 자바스크립트 요청 값에 대한 응답을 통해 세션 하이재킹이 발생할 수 있는 위험이 존재

◎ 조치방안

스크립트가 세션 쿠키에 접근하는 것을 보호하기 위한 "HttpOnly" 속성 설정

[ASP.NET]

파일경로 : systemroot₩Microsoft.NET₩Framework₩versionNumber₩CONFIG₩web.config

```
Response.Cookies.Add(new     HttpCookie("mycookie")
{
     Value = "쿠키 값",
     Secure    = true
}
또는
HttpCookie    myCookie = new HttpCookie("myCookie");
myCookie.HttpOnly = true;
Response.AppendCookie(myCookie);
```

```
Web.config
<httpCookies requireSSL="true"    />
```

※ 2.0 환경에서는 httponly가 기본으로 설정됨

[PHP]

파일경로 : /usr/local/lib/php.ini

```
Session.cookie_httponly    = True;
```

[TOMCAT]

파일경로 : /webapps/app/META-INF/context.xml

```
context.xml
<?xml    version="1.0" encoding="UTF-8"?>
<Context    path="/어플리케이션경로" useHttpOnly="true">
```

```
Servlet    3.0 이후 web.xml설정

    <session-config>
        <cookie-config>
            <http-only>true</http-only>
        </cookie-config>
    </session-config>
```

[3-5] 암호화된 세션(SSL) 쿠키의 누락된 보안 속성

◎ 개요

암호화 된 HTTPS 구간에서 개인정보를 쿠키로 저장할 때 쿠키객체의 보호를 위한 보안 속성을 적용하지 않을 경우 공격자에게 단순한 평문 형태로 노출되는 취약점

◎ 조치방안

HTTPS로만 서비스하는 경우 Cookie 객체의 setSecure(true) 메소드를 호출하여 브라우저 쿠키에 데이터를 저장하도록 설정

[안전한 코드 예제]

```
1: ……
2: private final String ACCOUNT_ID = "account";
3:
4: public void setupCookies(ServletRequest r, HttpServletResponse response) {
5: String acctID = r.getParameter("accountID");
6: // 계정 유효성 점검
7: if (acctID == null || "".equals(acctID)) return;
8: String filtered_ID = acctID.replaceAll("₩r", "");
9:
10: Cookie c = new Cookie(ACCOUNT_ID, filtered_ID);
11: // 민감한 정보를 가진 쿠키를 전송할때에는 보안 속성을 설정하여야 한다.
12: c.setSecure(true);
13: response.addCookie(c);
14: }
```

[3-6] HTTP.sys 원격 코드 실행

◎ 개요

HTTP 프로토콜 스택(HTTP.sys)이 특수하게 조작된 HTTP 요청의 구문을 검증하지 못하는 취약점으로 IIS 서버로 http header의 range 필드를 특수하게 조작한 패킷을 전송할 경우 블루스크린 유발 및 공격자가 원하는 원격코드의 실행이 가능함

◎ 조치방안

[취약한 OS 버전 사용 시 보안패치 수행]

Windows 7	Windows 7(32비트 시스템용) 서비스 팩 1
	Windows 7(x64 기반 시스템용) 서비스 팩 1
Windows Server 2008 R2	Windows Server 2008 R2(x64 기반 시스템용) 서비스 팩 1
	Windows Server 2008 R2(Itanium 기반 시스템용) 서비스 팩 1
Windows 8	Windows 8
	Windows 8.1
Windows Server 2012	Windows Server 2012
	Windows Server 2012 R2
Server Core 설치 옵션	Windows Server 2008 R2 (x64 기반 시스템용) 서비스 팩 1 (Server Core 설치)
	Windows Server 2012(Server Core 설치)

[안전한 코드 예제]

```
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <netdb.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <arpa/inet.h>

int connect_to_server(char *ip)
{
                        int sockfd = 0, n = 0;
        struct sockaddr_in serv_addr;
        struct hostent *server;

        if((sockfd = socket(AF_INET, SOCK_STREAM, 0)) < 0)
        {
                printf(" \n Error : Could not create socket \n");
                return 1;
        }
        memset(&serv_addr, '0', sizeof(serv_addr));
        serv_addr.sin_family = AF_INET;
        serv_addr.sin_port = htons(80);
        if(inet_pton(AF_INET, ip, &serv_addr.sin_addr)<=0)
        {
                printf(" \n inet_pton error occured\n");
                return 1;
        }

        if( connect(sockfd, (struct sockaddr *)&serv_addr, sizeof(serv_addr)) < 0)
        {
                printf(" \n Error : Connect Failed \n");
                return 1;
        }

        return sockfd;
        }


int main(int argc, char *argv[])
        {
        int n = 0;
        int sockfd;
        char recvBuff[1024];
        // Check server
        char request[] = "GET / HTTP/1.0 \r \n \r \n";
        // our evil buffer
        char request1[] = "GET / HTTP/1.1 \r \nHost: stuff \r \nRange: bytes=0-18446744073709551615
\r \n \r \n";

        if(argc != 2)
        {
        printf(" \n Usage: %s <ip of server> \n",argv[0]);
        return 1;
        }
        printf("[*] Audit Started \n");
        sockfd = connect_to_server(argv[1]);
        write(sockfd, request, strlen(request));
        read(sockfd, recvBuff, sizeof(recvBuff)-1);

        if (!strstr(recvBuff,"Microsoft"))
        {
        printf("[*] NOT IIS \n");
        exit(1);
        }

        sockfd = connect_to_server(argv[1]);
        write(sockfd, request1, strlen(request1));
        read(sockfd, recvBuff, sizeof(recvBuff)-1);
        if (strstr(recvBuff,"Requested Range Not Satisfiable"))
        {
        // 해당 취약점에 취약할 경우 "Looks VULN" 화면이 보임
        printf("[!!] Looks VULN \n");
        exit(1);
        } else if(strstr(recvBuff,"The request has an invalid header name"))
        {
        // 해당 취약점에 취약하지 않을 경우 "Looks Patched" 화면출력
        printf("[*] Looks Patched");
        } else
        // IIS 서버인지 확인 필요
        printf("[*] Unexpected response, cannot discern patch status");
}
```

[3-7] 영구적 쿠키에 중요 세션 정보 포함

◎ 개요

영구적 보관이 되는 쿠키는 파일 형태로 디스크에 저장되는데, 취약한 시스템은 쿠키 파일경로가 노출되어 디스크에 저장된 사용자의 쿠키정보가 탈취될 위험성이 있으며, 이를 통해 권한상승 및 주요 정보 탈취가 가능함

◎ 조치방안

사용자 인증정보와 세션 토큰과 같은 중요한 세션 정보가 유지되지 않도록 "non-permanent" 쿠키로 저장 (설정을 위해서는 쿠키의 "Expires" 필드를 설정하지 않아야 함)

[JAVA]

```
[JAVA]
Cookie tempCookie = new Cookie();
tempCookie.setMaxAge(60*60*24*365); // 밀리세컨드 단위, 1년동안 쿠키값 유효함. , BAD Code

tempCookie.setMaxAge(-1); // 파라미터 음수값 설정, 브라우저종료시 쿠키삭제, GOOD Code
```

[3-8] Microsoft FrontPage Extensions사이트 손상

◎ 개요

Frontpage Extention은 MicroSoft office에서 제공하는 웹 에디터인 Frontpage의 관리기능을 제공하며 모든 사용자가 접근이 가능하도록 초기 설정되어 있는데 공격자는 이를 통해 웹 페이지에 접근하여 악의적인 웹 서버 권한탈취 및 위·변조가 가능한 취약점

◎ 조치방안

취약한 파일 접근 권한 설정 (AUATHOR.DLL, ADMIN.DLL)

IIS 관리 콘솔 -> 웹사이트의 _VTI_BIN 디렉토리 -> 하위 파일선택(_VTI_AUTH/AUATHOR.DLL, _VTI_ADMIN/ADMIN.DLL) -> 파일의 등록정보 -> 보안탭 선택 -> 익명 액세스 체크 해제

<FrontPage 접근권한 설정>

제4절 WebDAV 취약점

[4-1] WebDAV 취약점

◎ 개요

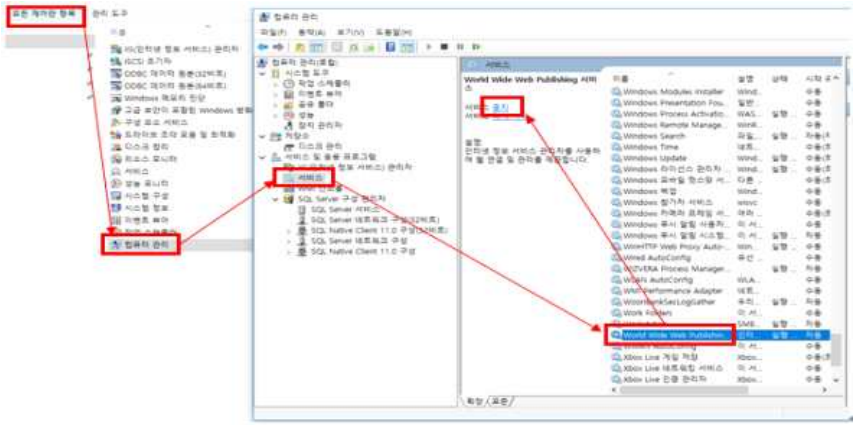
원격지에서 웹 서버 상의 컨텐츠를 조작할 수 있는 기능을 제공하는 WebDAV 라이브러리 파일의 속성에 읽기/쓰기 권한이 모두 허용되어 공격자가 원격에서 임의조작을 통한 웹페이지 위변조가 가능한 취약점

◎ 조치방안

[IIS를 사용하지 않을 경우]

서비스 상태를 '중지', 시작유형을 '사용 안함' 설정

IIS 경로 : [제어판]-[컴퓨터 관리]-[서비스]-[World Wide Web Publishing Service]

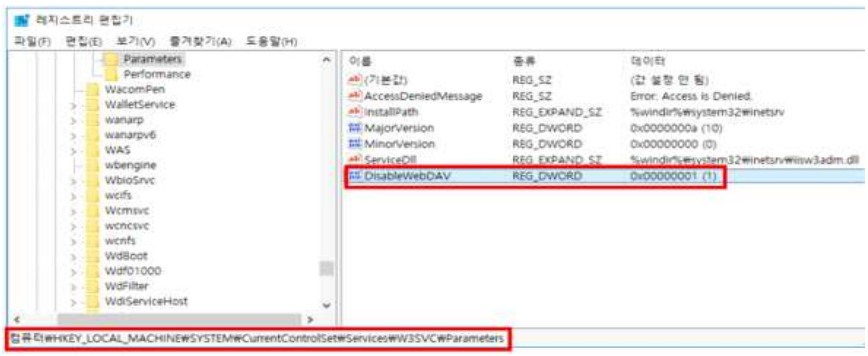


<IIS 비활성화 설정>

[WebDAV 기능이 필요하지 않을 경우-①]

경로 : HKEY_LOCAL_MACHINE₩SYSTEM₩CurrentControlSet₩Services₩W3SVC₩Parameter

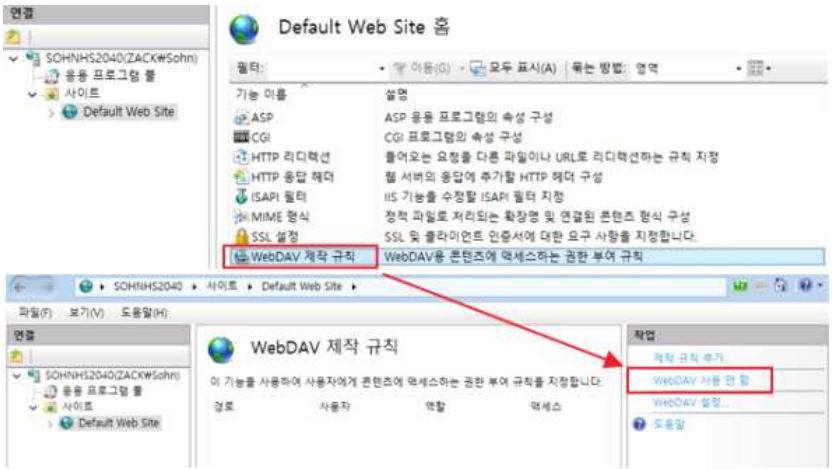
설정 방법 : DisableWebDAV의 DWORD 값을 만들어 '1'로 설정 후 IIS를 재시작



<WebDAV 실행 서비스 비활성화 설정>

[WebDAV 기능이 필요하지 않을 경우-②]

경로 : [제어판]-[관리도구]-[IIS 관리자]-[WebDAV 제작 규칙]-'WebDAV 사용 안함' 설정

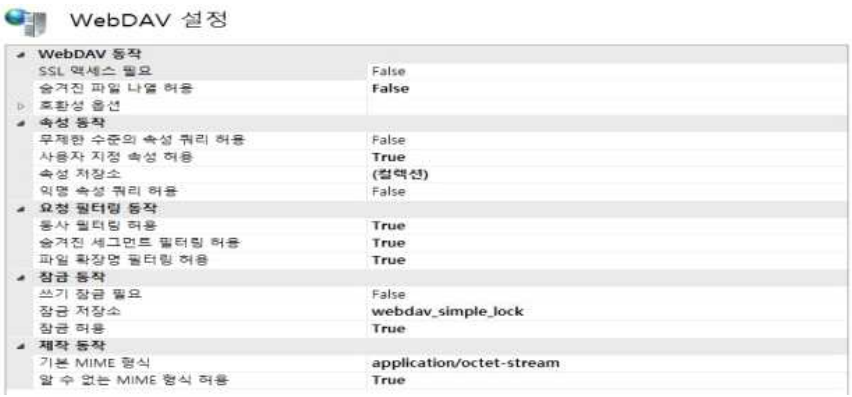


<WebDAV 비활성화 설정>

[WebDAV 기능을 사용할 경우]

경로 : [제어판]-[관리도구]-[IIS 관리자]-[WebDAV 제작 규칙]-[WebDAV 설정]

방법 : <WebDAV 주요 설정>표를 참고하여 권한 설정 적용



항목	설명
SSL 액세스 필요	SSL 인증서 필수 사용 여부
숨겨진 파일 나열 허용	숨겨진 파일 공개 여부
무제한 수준의 속성 쿼리 허용	지정된 경로 외에 다른 경로로의 접근허용
익명 속성 쿼리 허용	익명 사용자 접속을 허용
쓰기 잠금 필요	파일 쓰기(업로드) 제한
알 수 없는 MIME 형식 허용	알 수 없는 확장자 형식 허용

<WebDAV 설정관련 주요 변수>

제5절 불필요한 Method 허용 취약점

[5-1] TRACE 및 TRACK HTTP 메소드 사용

◎ 개요

TRACE 메소드 요청 시 서버에서 요청받은 메시지를 사용자에게 그대로 반환하는 특성을 악용하여 공격자가 악성 스크립트를 통한 사용자의 쿠키 및 중요 정보를 탈취할 수 있는 취약점

◎ 조치방안

1) CONNECT, PUT, DELETE, TRACE 메서드 비활성화

[IIS]

윈도우IIS 관리창의 서비스 확장에 있는WebDAV 허용을 비활성으로 변경

IIS Lockdown 툴 사용

ㄱ. IIS Lockdown Tool 설치

ㄴ. Lockdown Tool에 포함된 URLScan 설치
(URLScan은HTTP 요청을 블로킹 처리함으로 IIS 서버를 보호)

ㄷ. %windows directory%₩system32₩inetsrv₩urlscan 경로 이동

ㄹ. 해당 경로의 urlscan.ini 파일을 아래와 같이 설정

```
[options]
UseAllowVerbs=1

[AllowVerbs]
GET
HEAD
POST
또는
[options]
UseAllowVerbs=0

[AllowVerbs]
TRACE
TRACK
CONNECT
DELETE
```

[APACHE]

Httpd.conf 그리고 httpd-ssl.conf 아래와 같이 설정

```
<Directory /> //도메인 경로
 <LimitExcept GET POST> // 허용하는 메서드
  Order allow,deny
  deny from all
 </LimitExcept>
</Directory>
//get, post이외의 메서드는 모두 비활성화
```

[TOMCAT]

Web.xml 아래와 같이 설정

```
<security-constraint>
        <web-resource-collection>
        <web-resource-name></web-resource-name>
        <url-pattern>/*</url-pattern>
        <http-method>HEAD</http-method>
        <http-method>DELETE</http-method>
        <http-method>PUT</http-method>
        <http-method>OPTIONS</http-method>
        </web-resource-collection>
</security-constraint>
```

2) 크로스사이트 스크립트에 준하는 시큐어코딩

- 사용자 입력UI에서 스크립트 입력 거부 및 방지, 필터링

[5-2] 안전하지 않은 HTTP 메소드 사용

◎ 개요

웹 어플리케이션에서 일반적으로 사용하는 GET, POST 메소드 이외의 불필요한 메소드를 허용하였을 경우 공격자는 이를 이용하여 파일삭제, 파일 업로드 등 웹 서버를 인증 없이 조작이 가능

◎ 조치방안

GET 및 POST를 제외한 메서드 비활성화 조치

제6절 취약한 파일 존재 취약점

[6-1] 파일 대체 버전, 애플리케이션 테스트 스크립트, 압축된 디렉터리 발견

◎ 개요

웹서버 구성 시 기본 설치 파일, 테스트용 파일 등 웹 서버 운영에 사용되지 않은 불필요한 파일들이 외부로 노출되어 해당 파일을 통해 공격자가 내부 구성 정보수집, 테스트용 파일 조작 등 2차 공격에 악용될 수 있는 취약점

◎ 조치방안

1) 설치 시 자동 배포되는 샘플 파일 삭제
   - 사용하지 않는 파일 및 테스트 용도 디렉토리/파일 삭제 후 배포
     (예 : ASP, JSP, PHP, JAVA, PERL, txt, html ,js ,zip, .war , tar 등)
   - 소스코드내 사용하지 않는 코드 삭제(예 : 테스트/디버깅 용도 main 함수)

2) 파일 대체 버전 발견 조치방안
   - 가상 웹 서버 루트 아래에 존재하는 파일의 대체 버전 조회 후 삭제
   - 가상 루트 아래에는 항상 사용 중인 파일만 있음을 확인

3) 애플리케이션 테스트 스크립트 발견 조치방안
   - 서버에서 test/temporary 스크립트를 삭제
   - 서버 오퍼레이션에 필수적인 또 다른 스크립트가 서버에 존재하는지 확인

4) 압축된 디렉토리 발견
   - 압축된 디렉토리 파일에 대한 액세스 제한 또는 제거

[6-2] CMME 정보 유출

◎ 개요

CMME(Content Management Made Easy)는 PHP기반 웹서버의 관리편의를 위해 사용하는 웹 컨텐츠 관리시스템으로 웹서버의 구성정보에 관한 중요 정보를 포함하고 있으며 접근권한이 미흡할 경우 공격자는 이에 접근하여 서버에 대한 내부정보를 수집할 수 있는 취약점

◎ 조치방안

설정파일("data/admin/users", "info.php")에 대하여 통해 신뢰된 사용자 액세스가 가능하도록 정책설정

[6-3] URL 경로 재지정을 통한 피싱

◎ 개요

공격자가 HTTP 매개변수 조작을 통해 악의적인 사이트로 리다이렉션하여 사용자를 피싱 사이트로 접속하도록 유인하는 취약점으로 사용자명, Password, 신용카드번호, 주민등록번호 등 민감한 정보의 탈취가 가능한 위험성이 존재함

예시) http://example.com/example.php?url=http://malicious.example.com

◎ 조치방안

허용가능한 URL과 도메인들의 화이트리스트를 설정하여 악의적인 사이트 접근 차단

[안전한 코드 예제]

```
1: ……
2: protected void doGet(HttpServletRequest request, HttpServletResponse response)
3: throws ServletException, IOException {
4: String query = request.getQueryString();
5:
6: // 다른 페이지 이동하는 URL 리스트를 만든다.
7: String allowURL[] = { "url1", "url2", "url3" };
8: ArrayList arr = new ArrayList();
9: for ( int i = 0; i < allowURL.length; i++ )
10: arr.add(allowURL[i]);
11:
12: if (query.contains("url")) {
13: String url = request.getParameter("url");
14: // url에 대한 유효성 점검을 한다. 만약 http://가 있으면 다른 도메인으로 URL을
redirect로 의심된다.
15: if (url != null && url.indexOf("http://") != -1 ) {
16: url = url.replaceAll("₩r", "").replaceAll("₩n", "");
17: // URL 목록에 있지 않으면 요청을 거부한다.
18: if ( !arr.contains(url) ) throw new MyException("에러");
19: response.sendRedirect(url);
20: }
21: }
22: ……
```

[6-4] PHP phpinfo.php 정보 유출

◎ 개요

서버의 환경 설정에 대한 많은 정보를 포함하고 있는 phpinfo.php 페이지가 공격자에게 노출되어 예제 및 샘플 페이지, 설치 정보 등 웹서버의 구성정보를 열람 가능한 취약점으로 2차 공격 수행에 필요한 정보가 노출되는 위험성 존재

PHP Version 5.2.12

System	Windows NT DEV1-1 6.2 build 9200
Build Date	Dec 16 2009 17:01:16
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-snapshot-template=d:₩php-sdk₩snap_5_2₩vc6₩x86₩template" "--with-php-build=d:₩php-sdk₩snap_5_2₩vc6₩x86₩php_build" "--with-pdo-oci=D:₩php-sdk₩oracle₩instantclient10₩sdk,shared" "--with-oci8=D:₩php-sdk₩oracle₩instantclient10₩sdk,shared" "--without-pi3web"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:₩Windows
Loaded Configuration File	C:₩APM_Setup₩php.ini
Scan this dir for additional .ini files	(none)
additional .ini files parsed	(none)
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	enabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	php, file, data, http, ftp, compress.zlib
Registered Stream Socket Transports	tcp, udp
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, zlib.*

<PHP 구성정보 노출>

◎ 조치방안

1) phpinfo.php 페이지 삭제

```
rm /var/www/html/phpinfo.php (예시)
```

2) phpinfo() 비활성화

PHP 경로 : /usr/local/apache/conf/php.ini

```
disable_functions = phpinfo()
```

[6-5] 임시파일 및 아카이브 파일 다운로드

◎ 개요

URI에 노출되는 상대경로 조작을 통해 임시파일에 대한 다운로드나 열람이 가능한 취약점으로 시스템 구성 시 자동으로 생성되는 기본 설치경로 및 임시파일이 외부 비인가자에게 노출 될 경우 공격자가 경로를 유추하여 내부 핵심 정보를 획득할 수 있는 취약점

◎ 조치방안

- 임시파일 삭제 또는 웹 시스템의 가상 디렉토리 외부로 이동
- 다운로드를 허용할 디렉토리를 지정하여 해당 디렉토리를 벗어나는 위치의 다운로드 요청에 대해서는 경고메시지와 함께 다운로드를 금지하도록 함

제7절 계정 관리 취약점

[7-1] 올바르지 않은 계정 잠금

◎ 개요

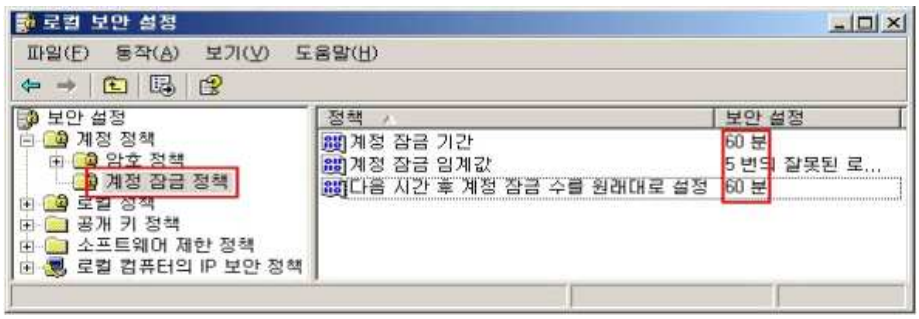
취약한 계정 정책 또는 로그인 실패에 대한 상세한 오류메시지는 공격자에게 유용한 정보를 제공하며 별도의 인증수단이 없을 경우 무차별 대입공격이나 사전대입공격에 노출되어 관리자 권한을 획득 가능

◎ 조치방안

1) 계정 잠금 정책 설정

[WINDOWS]

경로 : [시작]-[설정]-[제어판]-[관리도구]-[로컬 보안 설정]-[계정정책]-[계정 잠금 정책]
계정 잠금 기간 : 계정 잠금 임계값에 도달했을 경우 계정 잠금 상태를 유지할 기간 계정 잠금 임계값 : 사전 공격 방지를 위하여 지정 횟수 이상 로그인 실패 시 계정잠금 기간 동안 계정 사용 불가



<계정 잠금정책 설정>

[CenOS 5]

경로 : ₩etc₩pam.d₩password-auth
no_magic_root : root는 패스워드 잠금 설정 제외 deny=5 : 5회 입력 실패시 패스워드 잠금 unlock_time : 계정 잠김 후 잠금 해제되는 시간(초) reset : 접속 시도 성공시 실패한 횟수 초기화

```
CentOS 5
[root@localhost ~]# /etc/pam.d/system-auth
...
auth     required   pam_tally.so deny=5 unlock_time=60 no_magic_root
...
account  required   pam_tally.so no_magic_root reset
...
```

2) 오류메시지 노출 설정

[IIS]

경로 : [시작]-[설정]-[제어판]-[관리도구]-[인터넷 서비스 관리자]-[등록정보]-[사용자 정의] 오류 등록 정보 편집을 통해 사용자 정의 에러페이지를 지정
계정 잠금 기간 : 계정 잠금 임계값에 도달했을 경우 계정 잠금 상태를 유지할 기간 계정 잠금 임계값 : 사전 공격 방지를 위하여 지정 횟수 이상 로그인 실패 시 계정 잠금 기간 동안 계정사용 불가

[Apache]

경로 : /etc/httpd/conf/httpd.conf 유효하지 않은 요청은 별도로 만든 사용자 정의 에러페이지로 Redirect 설정
<httpd.conf> ErrorDocument 404/error_page.html ServerSignature off //Error 페이지 등에서 노출되는 웹서버 버전정보를 나타내지 않도록 설정

[Tomcat]

경로 : <Tomcat home directory>₩conf₩web.xml 에러코드에 따른 포워딩 페이지 설정
<web.xml> 404 /error_page/404.jsp

제4장 결론

최근 들어 인터넷 기술이 고도화되면서, 정보개방의 필요성이 급증하고 있다. 이에 공공정보들이 외부로 오픈된 인터넷을 통해 정보유통을 통한 소통의 장으로 활용되어 사용자들은 언제 어디서나 원하는 정보를 편리하게 확인할 수 있는 환경이 제공되고 있다. 하지만 누구에게나 접근이 가능한 인터넷의 구조적인 취약점은 상당히 많은 위험성을 가지고 있으며, 실제 웹 어플리케이션을 활용한 침해사고가 매년 지속적으로 증가하고 있는 추세이다.

특히 애플리케이션 계층은 네트워크 계층이나 시스템 계층에 비해 기술적으로 고도화되어 있고 애플리케이션의 종류도 다양하기 때문에 대부분의 보안 관리자들이 보안 정책을 수립하고 적용함에 있어 가장 많은 어려움을 겪는 실정이다. 또한 대부분의 운영자들이 홈페이지 구축 시 사용자의 편의성에만 주안점을 두는 경향이 있어 보안대책에 크게 관심을 갖지 않는 것도 웹 어플리케이션 분야의 침해사고 발생 증가의 원인이 되고 있는 것이다.

현재 이루어지고 있는 웹 공격의 90% 이상이 웹 애플리케이션을 노린 공격이라고 해도 과언이 아니다. 결국 안전한 웹 보안을 구축하고자 한다면 홈페이지 구축단계에서의 철저한 보안 코딩을 비롯하여 근본적인 취약점을 최소화가 필요하며, 지속적인 취약점 점검 및 보안조치 수행을 통해 보다 안전한 웹 어플리케이션 환경을 마련하여야 할 것이다.

참고 자료

[1] Secure coding, http://cwe.mitre.org

[2] JAVA, http://wikisecurity.net/guide:java

[3] MS, https://technet.microsoft.com/library/security/ms13-078

[4] MS, http://insecure.org/sploits/Microsoft.frontpage.insecurities.html

[5] REDHAT, https://securityblog.redhat.com/2014/10/15/poodle-a-ssl3-vulnerability-cve-2014-3566/

[5] APACHE, http://www.apache.org/