
Web Vulnerability Diagnostic Results Report

웹 취약점 점검 보고서

본 문서는 웹 취약점 점검 세부 작업 수행결과를 담고 있는 문서입니다. 본 문서의 유출 또는 분실 등의 사고로 인해 본 정보가 노출될 경우 해당 정보자산을 이용하고 있는 사용자의 손실 및 피해를 초래할 수 있으며, 사회적인 혼란 및 피해를 야기할 수 있습니다.

따라서 사전 승인 없이 본 내용의 전부 또는 일부에 대한 복사, 전제, 배포, 사용 등을 금합니다.

Web 취약점 분석·평가 항목

점검항목	항목 중요도	항목코드
버퍼 오버플로우	상	BO
포맷스트링	상	FS
LDAP 인젝션	상	LI
운영체제 명령 실행	상	OC
SQL 인젝션	상	SI
SSI 인젝션	상	SS
XPath 인젝션	상	XI
디렉터리 인덱싱	상	DI
정보 누출	상	IL
악성 콘텐츠	상	CS
크로스사이트 스크립팅	상	XS
약한 문자열 강도	상	BF
불충분한 인증	상	IA
취약한 패스워드 복구	상	PR
크로스사이트 리퀘스트 변조(CSRF)	상	CF
세션 예측	상	SE
불충분한 인가	상	IN
불충분한 세션 만료	상	SC
세션 고정	상	SF
자동화 공격	상	AU
프로세스 검증 누락	상	PV
파일 업로드	상	FU
파일 다운로드	상	FD
관리자 페이지 노출	상	AE
경로 추적	상	PT
위치 공개	상	PL
데이터 평문 전송	상	SN
쿠키 변조	상	CC

XS (상)		11. 크로스사이트 스크립팅
취약점 개요		
점검내용	■	웹 사이트 내 크로스사이트 스크립팅 취약점 존재 여부 점검
점검목적	■	웹 사이트 내 크로스사이트 스크립팅 취약점을 제거하여 악성 스크립트의 실행을 차단
보안위협	■	웹 애플리케이션에서 사용자 입력 값에 대한 필터링이 제대로 이루어지지 않을 경우, 공격자는 사용자 입력 값을 받는 게시판, URL 등에 악의적인 스크립트(Javascript, VBScript, ActiveX, Flash 등)를 삽입하여 게시글이나 이메일을 읽는 사용자의 쿠키(세션)를 탈취하여 도용하거나 악성코드 유포 사이트로 Redirect 할 수 있음
참고	※	크로스사이트 스크립팅: 악의적인 사용자가 공격하려는 사이트에 스크립트를 넣는 기법으로 공격 방식은 크게 stored 공격 방식과 reflected 공격 방식으로 나누어 짐 ※ OWASP - XSS 필터링 관련 참고사항 https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet ※ 소스코드 및 취약점 점검 필요
점검대상 및 판단기준		
대상	■	웹 애플리케이션 소스코드, 웹 방화벽
판단기준	양호 :	사용자 입력 값에 대한 검증 및 필터링이 이루어지는 경우
	취약 :	사용자 입력 값에 대한 검증 및 필터링이 이루어지지 않으며, HTML 코드가 입력·실행되는 경우
조치방법		웹 사이트의 게시판, 1:1 문의, URL 등에서 사용자 입력 값에 대해 검증 로직을 추가하거나 입력되더라도 실행되지 않게 하고, 부득이하게 웹페이지에서 HTML을 사용하는 경우 HTML 코드 중 필요한 코드에 대해서만 입력되게 설정
점검 및 조치 사례		
■ 점검방법 ※ XSS 취약 유형		
XSS에 취약한 페이지 유형		1. HTML을 지원하는 게시판 2. Search Page 3. Join Form Page 4. Referrer를 이용하는 Page 5. 그 외 사용자로부터 입력받아 화면에 출력하는 모든 페이지에서 발생 가능
XSS를 유발할 수 있는 스크립트		<pre><script> ... </script> <div style="background-image:url(javascript:...)"> </div> <embed> ...</embed></pre>

XS (상)

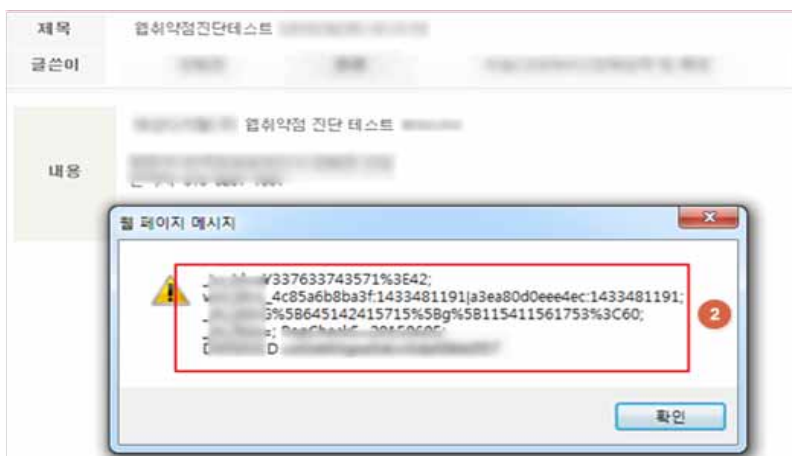
11. 크로스사이트 스크립팅

	<p><iframe></iframe></p> <p>※ Filtering을 우회하기 위해 다양한 표현 가능</p> <p>◆ %3Cscript%3E.....%3Cscript%3E ◆ Jav&#97;script;</p> <p>◆ Java&#13;script ◆ Java&#0013;script</p>
--	--

Step 1) 사용자 입력 값을 전달받는 애플리케이션(회원정보 변경, 게시판, 댓글, 자료실 등)에 스크립트 입력 후 실행되는지 확인



[게시글에 스크립트 삽입(stored)]



[스크립트 코드 동작]

XS (상)

11. 크로스사이트 스크립팅

Step 2) 사용자 입력 값을 전달받는 애플리케이션(검색, URL)에 스크립트 입력 후 실행되는지 확인



■ 보안설정방법

- * 웹 사이트에 사용자 입력 값이 저장되는 페이지는 공격자가 웹 브라우저를 통해 실행되는 스크립트 언어(HTML, Javascript, VBScript 등)를 사용하여 공격하므로 해당되는 태그 사용을 사전에 제한하고, 사용자 입력 값에 대한 필터링 작업이 필요함
- * 게시물의 본문뿐만 아니라 제목, 댓글, 검색어 입력 창, 그 외 사용자 측에서 넘어오는 값을 신뢰하는 모든 form과 파라미터 값에 대해서 필터링을 수행함
- * 입력 값에 대한 필터링 로직 구현 시 공백 문자를 제거하는 trim, replace 함수를 사용하여 반드시 서버 측에서 구현되어야 함
- * URLDecoder 클래스에 존재하는 decode 메소드를 통해 URL 인코딩이 적용된 사용자 입력 값을 디코딩함으로써 우회 공격 차단
- * 웹 방화벽에 모든 사용자 입력 품(회원정보 변경, 게시판, 댓글, 자료실, 검색, URL 등)을 대상으로 특수문자, 특수 구문 필터링하도록 룰셋 적용

※ 필터링 조치 대상 입력 값

- 스크립트 정의어 : <SCRIPT>, <OBJECT>, <APPLET>, <EMBED>, <FORM>, <IFRAME> 등
- 특수문자 : <, >, ", ', &, %, %00(null) 등

※ 웹 애플리케이션 별 상세 설정

■ ASP

```
<%
... 중략 ...
If use_HTML Then
    content = Server.HtmlEncode(content)
... 중략 ...

Sub ReplaceStr(content, byref str)
    content = replace(content, "<", "&lt;")
    content = replace(content, "&", "&amp;")
    content = replace(content, """, "&quot;")
    content = replace(content, "<", "&lt;")
    content = replace(content, ">", "&gt;")
    str = content
```

XS (상)

11. 크로스사이트 스크립팅

```
End Sub
... 종략 ...
%>
```

■ PHP

```
... 종략 ...
if($use_html == 1) // HTML tag를 사용해야 하는 경우 부분 허용
    $memo = str_replace("<", "&lt;", $memo);// HTML TAG 모두 제거
    $tag = explode(" ", $use_tag);

    for($i=0; $i<count($tag); $i++) { // 허용할 TAG만 사용할 수 있도록 변경
        $memo = eregi_replace("&lt;".$tag[$i].", " "<".$tag[$i].", " ", $memo);
        $memo = eregi_replace("&lt;".$tag[$i].">", "<".$tag[$i].">", $memo);
        $memo = eregi_replace("&lt;".$tag[$i].", "</".$tag[$i], $memo); }
    else // HTML tag를 사용하지 못하게 할 경우
        $memo = str_replace("<", "&lt;", $memo);
        $memo = str_replace(">", "&gt;", $memo);
... 종략 ...
```

■ JSP

```
<%
... 종략 ...
string subject = request.getParameter("subject_BOX");
subject = subject.replaceAll("<", "&lt;");
subject = subject.replaceAll(">", "&gt;");
... 종략 ...
%>
```

※ 참고: 필터링 대상

<	>	<	>	innerHTML
javascript	eval	onmousewheel	onactive	onfocusout
expression	charset	ondataavailable	oncut	onkeyup
applet	document	onafteripupdate	onclick	onkeypress
meta	string	onmousedown	onchange	onload
xml	create	onbeforeactivate	onbeforecut	onbounce
blink	append	onbeforecopy	ondbclick	onmouseenter
link	binding	onbeforedeactivate	ondeactivate	onmouseout
style	alert	ondataatchaged	ondrag	onmouseover
script	msgbox	cnbeforeprint	ondragend	onsubmit
embed	refresh	cnbeforepaste	ondragenter	onmouseend
object	void	onbeforeeditfocus	ondragleave	onresizestart
iframe	cookie	onbeforeunload	ondragover	onunload
frame	href	onbeforeupdate	ondragstart	onselectstart
frameset	onpaste	onpropertychange	ondrop	onreset
ilayer	onresize	ondatasetcomplete	onerror	onmove

XS (상)	11. 크로스사이트 스크립팅				
	layer	onselect	oncellchange	onfinish	onstop
	bgsound	base	onlayoutcomplete	onfocus	onrowexit
	title	onblur	onselectionchange	vbscript	onerrorupdate
	onbefore	onstart	onrowsinserted	onkeydown	onfilterchage
	onmouseup	onfocusin	oncontrolselected	onrowsdelete	onlosecapture
	onrowenter	onhelp	onreadystatechange	onmouseleave	onmousemove
	oncontextmenu				
조치 시 영향	일반적으로 영향 없음				

BF (상)	12. 약한 문자열 강도
취약점 개요	
점검내용	<ul style="list-style-type: none"> ■ 웹페이지 내 로그인 폼 등에 약한 강도의 문자열 사용 여부 점검
점검목적	<ul style="list-style-type: none"> ■ 유추 가능한 취약한 문자열 사용을 제한하여 계정 및 패스워드 추측 공격을 방지하기 위함
보안위협	<ul style="list-style-type: none"> ■ 해당 취약점 존재 시 유추가 용이한 계정 및 패스워드의 사용으로 인한 사용자 권한 탈취 위험이 존재하며, 해당 위험을 방지하기 위해 값의 적절성 및 복잡성을 검증하는 로직을 구현하여야 함
참고	<p>※ 약한 문자열 강도 취약점: 웹 사이트에서 취약한 패스워드로 회원가입이 가능할 경우 공격자는 추측 및 주변 정보를 수집하여 작성한 사전 파일로 대입을 시도하여 사용자 계정을 탈취할 수 있는 취약점</p> <p>※ 소스코드 및 취약점 점검 필요</p>
점검대상 및 판단기준	
대상	<ul style="list-style-type: none"> ■ 웹 애플리케이션 소스코드
판단기준	<p>양호 : 관리자 계정 및 패스워드가 유추하기 어려운 값으로 설정되어 있으며, 일정 횟수 이상 인증 실패 시 로그인을 제한하고 있는 경우</p>
	<p>취약 : 관리자 계정 및 패스워드가 유추하기 쉬운 값으로 설정되어 있으며, 일정 횟수 이상 인증 실패 시 로그인을 제한하고 있지 않은 경우</p>
조치방법	계정 및 비밀번호의 체크 로직 추가 구현
점검 및 조치 사례	
<p>■ 점검방법</p> <p>Step 1) 웹 사이트 로그인 페이지의 로그인 창에 추측 가능한 계정이나 패스워드를 입력하여 정상적으로 로그인되는지 확인</p> <ul style="list-style-type: none"> • 취약한 계정: admin, administrator, manager, guest, test, scott, tomcat, root, user, operator, anonymous 등 • 취약한 패스워드: Abcd, aaaa, 1234, 1111, test, password, public, blank 패스워드, ID와 동일한 패스워드 등 <div data-bbox="247 1150 871 1369" data-label="Image"> </div> <p>Step 2) 일정 횟수(3~5회) 이상 인증 실패 시 로그인을 제한하는지 확인</p>	

BF (상)

12. 약한 문자열 강도

■ 보안설정방법

- * 취약한 계정 및 패스워드를 삭제하고, 사용자가 취약한 계정이나 패스워드를 등록하지 못하도록 패스워드 규정이 반영된 체크 로직을 회원가입, 정보변경, 패스워드 변경 등 적용 필요한 페이지에 모두 구현하여야 함

※ 규정 예시



- Step 1) 다음 각 목의 문자 종류 중 2종류 이상을 조합하여 최소 10자리 이상 또는 3종류 이상을 조합하여 최소 8자리 이상의 길이로 구성
- (1) 영문 대문자(26개)
 - (2) 영문 소문자(26개)
 - (3) 숫자(10개)
 - (4) 특수문자(32개)
- Step 2) 연속적인 숫자나 생일, 전화번호 등 추측하기 쉬운 개인정보 및 아이디와 비슷한 비밀번호는 사용하지 않는 것을 권고
- Step 3) 비밀번호에 유효기간을 설정하여 반기별 1회 이상 변경
- Step 4) 최근 사용되었던 패스워드 재사용 금지

- * 로그인 시 패스워드 입력 실패가 일정 횟수(3~5회) 이상 초과할 경우 관리자에게 통보 및 계정 잠금

- ※ 인증 실패 횟수를 Client Side Script(Javascript, VBScript 등)를 사용하면 사용자가 임의로 수정할 수 있으므로 Server Side Script(PHP, ASP, JSP 등)를 통하여 구현

조치 시
영향

일반적으로 영향 없음

IA (상)	13. 불충분한 인증
취약점 개요	
점검내용	■ 중요 페이지 접근 시 추가 인증 요구 여부 점검
점검목적	■ 중요 페이지에 추가 인증으로 접근을 강화하여 불필요한 정보의 노출 및 변조를 차단하기 위함
보안위협	■ 중요정보(개인정보 변경 등) 페이지에 대한 인증 절차가 불충분할 경우 권한이 없는 사용자가 중요정보 페이지에 접근하여 정보를 유출하거나 변조할 수 있으므로 중요정보 페이지에는 추가적인 인증 절차를 구현하여야 함
참고	※ 소스코드 및 취약점 점검 필요
점검대상 및 판단기준	
대상	■ 웹 애플리케이션 소스코드
판단기준	양호 : 중요정보 페이지 접근 시 추가 인증을 하는 경우
	취약 : 중요정보 페이지 접근에 대한 추가 인증을 하지 않는 경우
조치방법	중요정보 페이지에 대한 추가 인증 로직 추가 구현
점검 및 조치 사례	
<p>■ 점검방법</p> <p>Step 1) 중요정보(개인정보 변경 등) 페이지 접근 시 재인증 여부 확인</p>  <p>Step 2) 인증 후 페이지에 아이디만을 인증 값으로 하여 변수로 관리되고 있는지 확인</p> 	

IA (상)

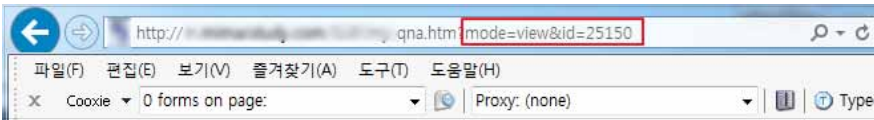

13. 불충분한 인증

■ 보안설정방법

- * 중요정보(개인정보 변경 등)를 표시하는 페이지에서는 본인 인증을 재확인하는 로직을 구현하고, 사용자가 인증 후 이용 가능한 페이지에 접근할 때마다 승인을 얻은 사용자인지 페이지마다 검증하여야 함
- * 접근 통제 정책을 구현하고 있는 코드는 구조화, 모듈화가 되어 있어야 함
- * 접근제어가 필요한 모든 페이지에 통제수단(로그인 체크 및 권한 체크)을 구현해야 하며 특히, 하나의 프로세스가 여러 개의 페이지 또는 모듈로 이루어져 있을 때 권한 체크가 누락되는 경우를 방지하기 위해서 공통 모듈을 사용하는 것을 권장함
- * 인증 과정을 처리하는 부분에 Client Side Script(Javascript, VBScript 등)를 사용하면 사용자가 임의로 수정할 수 있으므로 Server Side Script(PHP, ASP, JSP 등)를 통하여 인증 및 필터링 과정을 수행함

조치 시
영향

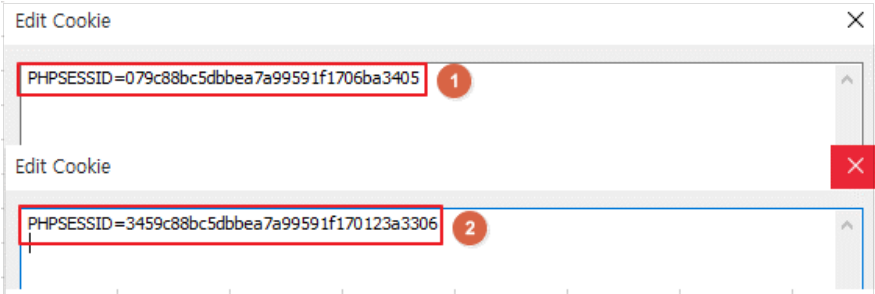
일반적으로 영향 없음

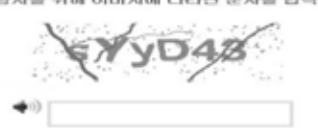
IN (상)	17. 불충분한 인가
취약점 개요	
점검내용	■ 민감한 데이터 또는 기능에 접근 및 수정 시 통제 여부 점검
점검목적	■ 접근 권한에 대한 검증 로직을 구현하여 비인가자의 악의적인 접근을 차단하기 위함
보안위협	■ 접근제어가 필요한 중요 페이지의 통제수단이 미흡한 경우, 비인가자가 URL 파라미터 값 변경 등의 방법으로 중요 페이지에 접근하여 민감한 정보 열람 및 변조 가능함
참고	※ 소스코드 및 취약점 점검 필요
점검대상 및 판단기준	
대상	■ 웹 애플리케이션 소스코드
판단기준	양호 : 접근제어가 필요한 중요 페이지의 통제수단이 적절하여 비인가자의 접근이 불가능한 경우
	취약 : 접근제어가 필요한 중요 페이지의 통제수단이 미흡하여 비인가자의 접근이 가능한 경우
조치방법	접근제어가 필요한 모든 페이지에 권한검증 로직 구현
점검 및 조치 사례	
<p>■ 점검방법</p> <p>Step 1) 비밀 게시글(또는 개인정보 변경, 패스워드 변경 등) 페이지에서 다른 사용자와의 구분을 ID, 일련번호 등의 단순한 값을 사용하는지 조사</p>  <p>Step 2) 게시글을 구분하는 파라미터 값을 변경하는 것만으로 다른 사용자의 비밀 게시글 (또는 개인정보 변경, 패스워드 변경 등)에 접근 가능한지 확인</p> 	

웹(Web)

IN (상)	17. 불충분한 인가
<p>■ 보안설정방법</p> <p>* 접근제어가 필요한 중요 페이지는 세션을 통한 인증 등 통제수단을 구현하여 인가된 사용자 여부를 검증 후 해당 페이지에 접근할 수 있도록 함</p> <p>* 페이지별 권한 매트릭스를 작성하여 접근제어가 필요한 모든 페이지에서 권한 체크가 이뤄지도록 구현하여야 함</p>	
조치 시 영향	일반적으로 영향 없음

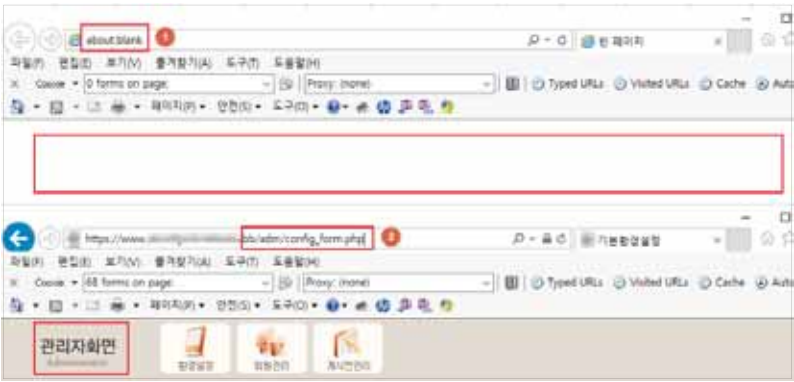
웹(Web)

SF (상)	19. 세션 고정
취약점 개요	
점검내용	■ 사용자 로그인 시 항상 일정하게 고정된 세션 ID 값을 발행하는지 여부 확인
점검목적	■ 로그인할 때마다 예측 불가능한 새로운 세션 ID를 발행하여 세션 ID의 고정 사용을 방지하기 위함
보안위협	■ 사용자 로그인 시 항상 일정하게 고정된 세션 ID가 발행되는 경우 세션 ID를 도용한 비인가자의 접근 및 권한 우회가 가능
참고	※ 소스코드 및 취약점 점검 필요
점검대상 및 판단기준	
대상	■ 웹 애플리케이션 소스코드
판단기준	양호 : 로그인할 때마다 예측 불가능한 새로운 세션 ID가 발행되고, 기존 세션 ID는 파기될 경우
	취약 : 로그인 세션 ID가 고정 사용되거나 새로운 세션 ID가 발행되지만 예측 가능한 패턴으로 발행될 경우
조치방법	사용자가 로그인할 때마다 예측 불가능한 새로운 세션 ID 생성 로직 구현하고 기존 세션 ID는 파기함
점검 및 조치 사례	
■ 점검방법 Step 1) 로그인 시(1) 세션 ID가 발행되는지 확인하고 로그아웃 후 다시 로그인(2)할 때 예측 불가능한 새로운 세션 ID가 발급되는지 확인	
	
■ 보안설정방법 * 로그인할 때마다 예측 불가능한 새로운 세션 ID를 발급받도록 해야 하고 기존 세션 ID는 파기해야 함	
조치 시 영향	일반적으로 영향 없음

AU (상)	20. 자동화 공격
취약점 개요	
점검내용	<ul style="list-style-type: none"> ■ 웹 애플리케이션의 특정 프로세스(로그인 시도, 게시물 등록, SMS 발송 등)에 대한 반복적인 요청 시 통제 여부 확인
점검목적	<ul style="list-style-type: none"> ■ 무차별 대입 공격 및 자동화 공격으로 웹 애플리케이션에 자원이 고갈되는 것을 방지하기 위함
보안위협	<ul style="list-style-type: none"> ■ 웹 애플리케이션의 특정 프로세스에 대한 반복적인 요청을 통제하지 않을 경우 무차별 대입 공격으로 인해 사용자 계정을 탈취할 수 있고, 자동화 공격으로 게시물 등록 또는 SMS 발송 요청을 반복하여 웹 애플리케이션 자원을 고갈시킬 수 있음
참고	※ 소스코드 및 취약점 점검 필요
점검대상 및 판단기준	
대상	<ul style="list-style-type: none"> ■ 웹 애플리케이션 소스코드, 웹 방화벽
판단기준	양호 : 웹 애플리케이션의 특정 프로세스에 대한 반복적인 요청 시 통제가 적절한 경우
	취약 : 웹 애플리케이션의 특정 프로세스에 대한 반복적인 요청 시 통제가 미흡한 경우
조치방법	웹 애플리케이션의 특정 프로세스에 대한 대량 사용 통제 로직 구현 및 웹 방화벽 룰셋 설정을 통해 대량의 불특정 프로세스 요청 차단
점검 및 조치 사례	
<p>■ 점검방법</p> <p>Step 1) 로그인 시도, 게시물 등록, SMS 발송 등에 대한 정상적인 요청 정보를 식별하여 반복적으로 요청 시 통제가 이루어지는지 확인</p> <p>■ 보안설정방법</p> <p>* 로그인 시도, 게시물 등록, SMS 발송 등에 대한 사용자 요청이 일회성이 될 수 있도록, 캡차(이미지를 이용하여 확인 값을 표시하고 사용자가 값을 등록하여 인증함) 등 일회성 확인 로직을 구현하여야 함</p> <p>※ 캡차(CAPTCHA): 자동화된 컴퓨터와 사람을 판별하기 위한 기술의 일종</p> <div data-bbox="280 1260 840 1444"> <p>자동 등록 절차를 위해 이미지에 나타난 문자를 입력해 주세요.</p>  <p>다른 이미지 보기 다음 단계로</p> </div>	

웹(Web)

AU (상)	20. 자동화 공격
	<p>* 자동화 공격을 시도하면 짧은 시간에 다량의 패킷(양)이 전송되므로 이를 공격으로 감지하고 방어할 수 있는 IDS/IPS 시스템을 구축하여야 함. 서버에 요청되는 패킷(양)의 모니터링이 불가능한 경우 적시에 적절한 대응이 어려움</p>
조치 시 영향	일반적으로 영향 없음

PV (상)	21. 프로세스 검증 누락
취약점 개요	
점검내용	<ul style="list-style-type: none"> ■ 인증이 필요한 웹 사이트의 중요(관리자 페이지, 회원변경 페이지 등) 페이지에 대한 접근제어 설정 여부 확인
점검목적	<ul style="list-style-type: none"> ■ 인증이 필요한 모든 페이지에 대해 유효 세션임을 확인하는 프로세스 및 주요 정보 페이지에 접근 요청자의 권한 검증 로직을 적용하여, 비인가자가 하위 URL 직접 접근, 스크립트 조작 등의 방법으로 중요한 페이지에 접근을 시도하는 것을 차단하기 위함
보안위협	<ul style="list-style-type: none"> ■ 인증이 필요한 웹 사이트의 중요(관리자 페이지, 회원변경 페이지 등) 페이지에 대한 접근 제어가 미흡할 경우 하위 URL 직접 접근, 스크립트 조작 등의 방법으로 중요한 페이지에 대한 접근이 가능함
참고	※ 소스코드 및 취약점 점검 필요
점검대상 및 판단기준	
대상	<ul style="list-style-type: none"> ■ 웹 애플리케이션 소스코드
판단기준	양호 : 인증 후에 접근해야 하는 웹 사이트의 하위 URL을 로그인하지 않고 직접 접근할 때 접근이 불가능한 경우
	취약 : 웹 사이트의 하위 URL을 로그인하지 않고 직접 접근할 때 접근이 가능한 경우
조치방법	인증이 필요한 페이지의 경우 페이지별 권한 체크 로직 구현
점검 및 조치 사례	
<p>■ 점검방법</p> <p>Step 1) 업무프로세스 파악</p> <p>Step 2) 권한의 종류 및 범위 파악</p> <p>Step 3) 페이지의 모든 기능을 수집하여 프로세스 상에 통제된 페이지 접근이 가능한지 확인</p> 	

PV (상)

21. 프로세스 검증 누락

■ 보안설정방법

- * 우회될 수 있는 플로우를 차단하여야 하며, 페이지별 권한 매트릭스를 작성하여 페이지에 부여된 권한의 타당성을 체크한 후 권한 매트릭스를 기준으로 전 페이지에서 권한 체크가 이뤄지도록 구현하여야 함
- * 인증이 필요한 모든 페이지에 대해 유효 세션임을 확인하는 프로세스 및 주요 정보 페이지에 접근 요청자의 권한 검증 로직을 적용함
- * 유효 세션의 검증 및 페이지에 대한 접근 권한을 Client Side Script에 의존할 경우 사용자가 임의로 수정할 수 있으므로 Server Side Script로 구현된 프로세스를 사용

※ 웹 애플리케이션 별 상세 설정

■ ASP

(예) 인증이 필요한 페이지 소스 코드

```
<% - 인증 성공 시 세션값 세팅
Session("sessionChk") = True
Session("UserID") = userID
Session("UserGrp") = userGrp
Session("UserIP") = Request.ServerVariables("REMOTE_ADDR")
... 중략 ...
- 사용자 그룹 리턴 함수
... 중략 ...
Function GetUserGroup(strUserID)
End function ... 중략 ...
- 페이지에 접근 가능한 UserGroup 설정값이 '100' 가정 시
ChkUserGrp = GetUserGroup(userID)
//세션 userID값을 통해 DB에 저장된 사용자 그룹 리턴 ... 중략 ...
If Session_Check and Session("UserGrp") = ChkUserGrp Then
If Session("UserGrp") <> 100 Then
Response.Write("권한이 없습니다.")
Response.End
End
Else
Response.Redirect "Login.asp"
Response.End
End if
... 중략 ... %>
```


■ JSP

(예) 인증이 필요한 페이지 소스 코드

```
<%
... 중략 ...
PortalSessionManager sessionMgr = (PortalSessionManager)
session.getAttribute("sessionMgr");
if (sessionMgr == null || sessionMgr.getUserId() == null) {
(new FailToAuthenticateCmd()).execute(request,response);
}
... 중략 ...
String usrGrp = session.getAttribute("Usrgrp") == null ?
```

PV (상)	21. 프로세스 검증 누락
<pre>"" : (String)session.getAttribute("Usrgrp"); if (!usrGrp.equals("") !userGrp.equals(Code.getMarket())) { // 접근 권한을 인가할 수 없음. (new FailToPermissionCmd()).execute(request,response); } 중략 ... %></pre>	
조치 시 영향	일반적으로 영향 없음

웹(Web)

AE (상)	24. 관리자 페이지 노출
취약점 개요	
점검내용	■ 유추하기 쉬운 URL로 관리자 페이지 및 메뉴 접근의 가능 여부 점검
점검목적	■ 관리자 페이지 URL이 유추하기 쉬운 이름(admin, manager 등) 및 웹 사이트 설계 오류를 수정하여 비인가자의 관리자 메뉴 접근을 방지하고자 함
보안위협	■ 웹 관리자의 권한이 노출될 경우 웹 사이트의 변조뿐만 아니라 취약성 정도에 따라서 웹 서버의 권한까지도 노출될 수 있음
참고	※ 소스코드 및 취약점 점검 필요
점검대상 및 판단기준	
대상	■ 웹 애플리케이션 소스코드, 웹 서버, 웹 방화벽
판단기준	양호 : 유추하기 쉬운 URL로 관리자 페이지 접근이 불가능한 경우
	취약 : 유추하기 쉬운 URL로 관리자 페이지 접근이 가능한 경우
조치방법	유추하기 어려운 이름(포트 번호 변경 포함)으로 관리자 페이지를 변경하여 비인가자가 관리자 페이지에 접근할 수 없도록 하고 근본적인 해결을 위해 지정된 IP만 관리자 페이지에 접근할 수 있도록 제한하여야 함 단, 부득이하게 관리자 페이지를 외부에 노출해야 하는 경우 관리자 페이지 로그인 시 2차 인증(otp, vpn, 인증서 등) 적용 필요함
점검 및 조치 사례	
■ 점검방법	
Step 1) 추측하기 쉬운 관리자 페이지 경로(/admin, /manager, /master, /system 등) 접근을 시도하여 관리자 페이지가 노출되는지 확인	
	

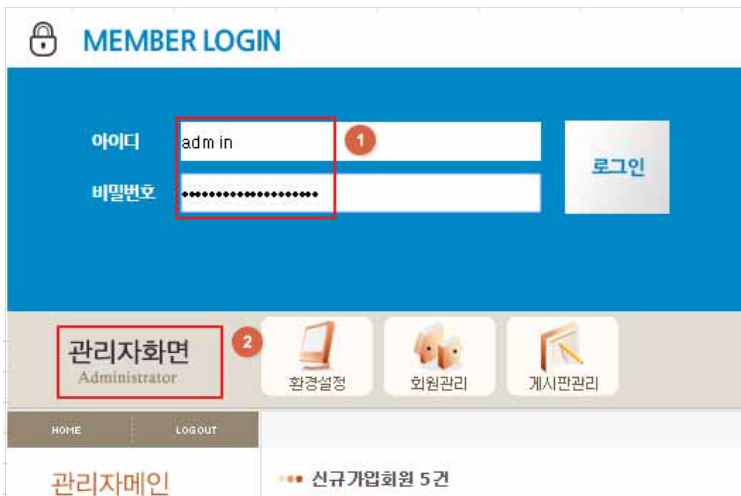
AE (상)

24. 관리자 페이지 노출

Step 2) 추측하기 쉬운 포트(7001, 8080, 8443, 8888 등) 접속을 시도하여 관리자 페이지가 노출되는지 확인



Step 3) 관리자 페이지의 로그인 창에 기본 관리자 계정(admin, administrator, manager 등) 및 패스워드를 입력하여 로그인 가능한지 확인



AE (상)

24. 관리자 페이지 노출

Step 4) 관리자 페이지 로그인 후 식별된 하위 페이지(/admin/main.asp, /admin/menu.html 등) URL을 새 세션에서 직접 입력하여 인증 과정 없이 접근 가능한지 확인



■ 보안설정방법

- * 일반 사용자의 접근이 불필요한 관리자 로그인 페이지 주소를 유추하기 어려운 이름으로 변경하고 관리자 페이지 접근 포트도 변경함
- * 관리자 페이지의 하위 페이지 URL을 직접 입력하여 접근하지 못하도록 페이지마다 세션 검증이 필요함
- * 관리자 페이지 이외에도 특정 사용자만 접근 가능한 페이지들은 정상적인 프로세스에 따라 접근할 수 있도록 페이지마다 세션 검증이 필요함
- * 웹 방화벽을 이용하여 특정 IP만 접근 가능할 수 있도록 룰셋 적용

조치 시
영향

일반적으로 영향 없음