

취약점 진단

결과 보고서

『주요정보통신기반시설 기술적 진단 컨설팅』

2025.12.01

한국폴리텍대학 대전캠퍼스
클라우드보안과 김민진

- 목 차 -

1. 개 요

가. 목 적	3
나. 진단 대상	3
다. 진단 방법	3
라. 진단 기준	3

2. 취약점 진단 결과

가. 취약점 진단 결과	4
나. 취약점 진단 항목 및 결과 요약표	5

3. 상세 진단 결과

1. 개요

1. 목적

- 주요정보통신기반시설 기술적 취약점 분석·평가 가이드에 따라 정보통신기반 보호법 기준으로 적용되어 있는지 진단하는 것이 목적임

2. 진단 대상

- Web Application을 대상으로 이행진단을 수행함

서비스명	세부 정보
Web Application	IP : 192.168.30.144 (진단 실습용 웹서버)

3. 진단 방법

Task	진단 수행 방법	일정
취약점 진단 계획수립	<ul style="list-style-type: none">취약점 진단 대상 및 진단항목 선정진단환경 구축	2025.11.24 ~ 2025.11.25
진단 수행	<ul style="list-style-type: none">Web Application 진단 수행취약점 진단 결과보고서 작성	2025.11.26 ~ 2025.12.01

4. 진단 기준

- 진단 등급

구 분	진단 방법
양호	진단 항목의 보안 설정이 잘 적용되어 있음
취약	진단 항목의 보안 설정이 적용되어 있지 않음

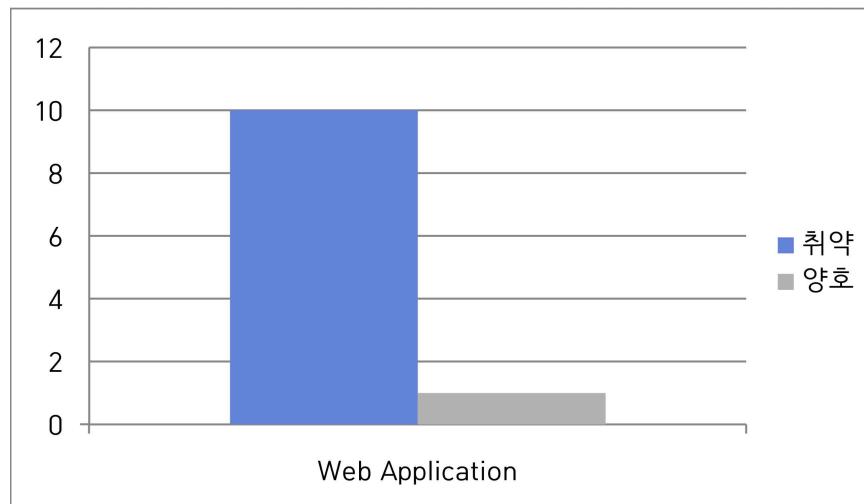
- 위험도

구 분	설 명
상	관리자 권한 획득 및 시스템 파괴, 정지 등 운영에 절대적인 위험
중	일반 권한을 획득하거나 중요한 정보유출로 추가적인 위험이 상당히 내포
하	시스템 정보 등의 위협은 내포되어 있으나, 공격 영향력은 가장 낮음

2. 취약점 진단 결과

1. 취약점 진단 결과 종합

- Web Application 진단 실시 결과, 도출된 취약점 총 10개, 안전 항목 1개가 확인되었으며, 취약 항목의 비율은 90.9%임.



구 분	진단항목	진단 결과		취약수준
		취약	양호	
web application	11	10	1	90.9%

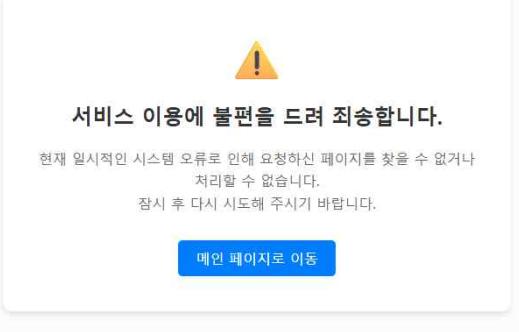
취약점 진단 항목 및 결과 요약표

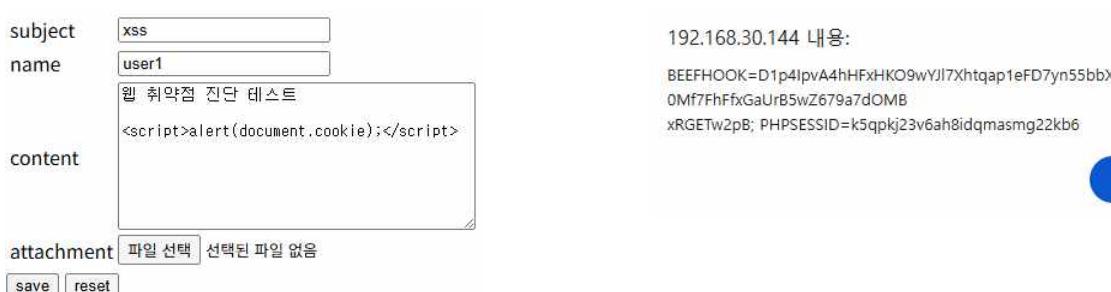
대상	취약점 진단 항목	항목 중요도	진단결과
Web Application	버퍼 오버플로우	상	양호
	SQL 인젝션	상	취약
	디렉터리 인덱싱	상	취약
	정보 누출	상	취약
	크로스사이트 스크립팅	상	취약
	약한 문자열 강도	상	취약
	크로스사이트 리퀘스트 변조 (CSRF)	상	취약
	불충분한 인가	상	취약
	세션 고정	상	취약
	파일 업로드	상	취약
	파일 다운로드	상	취약

2. 취약점 진단 결과

SI (상)	[Web] SQL 인젝션
취약점 개요	
점검내용	■ 웹페이지 내 SQL 인젝션 취약점 존재 여부 점검
점검목적	■ 대화형 웹 사이트에 비정상적인 사용자 입력 값 허용을 차단하여 악의적인 데이터베이스 접근 및 조작을 방지하기 위함
점검대상 및 판단기준	
판단기준	양호 : 임의로 작성된 SQL 쿼리 입력에 대한 검증이 이루어지는 경우
	취약 : 임의로 작성된 SQL 쿼리 입력에 대한 검증이 이루어지지 않는 경우
진단결과	취약
점검 및 조치사례	
<p>■ 진단순서</p> <p>Step 1) 사용자 입력 값에 특수문자나 임의의 SQL 쿼리를 삽입하여 DB 에러 페이지가 반환되는지 확인</p> <p>Step 2) 사용자 입력 값에 임의의 SQL 참, 거짓 쿼리를 삽입하여 참, 거짓 쿼리에 따라 반환되는 페이지가 다른지 확인</p> <p>Step 3) 로그인 페이지에 참이 되는 SQL 쿼리를 전달하여 로그인되는지 확인 (비밀번호에 'or'1'='1 입력)</p>	
<p>Username : <input type="text" value="admin"/> Password : <input type="password"/> <input type="button" value="Login"/></p> <p>Welcome admin <input type="button" value="Logout"/> <input type="button" value="Write"/></p>	
조치방안	<p>1) SQL 쿼리에 사용되는 문자열의 유효성을 검증하는 로직 구현</p> <p>2) ', ;, --, #, /*, */ 과 같은 특수문자를 사용자 입력 값으로 지정 금지</p> <p>3) Dynamic SQL 구문 사용을 지양하며 파라미터에 문자열 검사 필수 적용</p> <p>4) 시스템에서 제공하는 에러 메시지 및 DBMS에서 제공하는 에러 코드가 노출되지 않도록 예외 처리</p> <p>5) 웹 방화벽에 인젝션 공격 관련 설정</p> <p>설정 전</p> <pre>\$result = mysqli_query(\$con, "SELECT * FROM users WHERE username='\$username' AND password='\$password'");</pre> <p>설정 후 (prepared statement 설정)</p> <pre>\$stmt = mysqli_prepare(\$con, "SELECT * FROM users WHERE username=? AND password=?"); if(\$stmt){ mysqli_stmt_bind_param(\$stmt, "ss", \$username, \$password); mysqli_stmt_execute(\$stmt); \$result = mysqli_stmt_get_result(\$stmt);</pre>

DI (상)	[WEB] 디렉터리 인덱싱 취약점 개요																																				
점검내용	■ 웹 서버 내 디렉터리 인덱싱 취약점 존재 여부 점검																																				
점검목적	■ 디렉터리 인덱싱 취약점을 제거하여 특정 디렉터리 내 불필요한 파일 정보의 노출을 차단																																				
점검대상 및 판단기준																																					
판단기준	양호 : 디렉터리 파일 리스트가 노출되지 않는 경우																																				
	취약 : 디렉터리 파일 리스트가 노출되는 경우																																				
진단결과	취약																																				
점검 및 조치사례																																					
<p>■ 진단순서</p> <p>Step 1) URL 경로 중 확인하고자 하는 디렉터리까지만 주소창에 입력하여 인덱싱 여부 확인</p>																																					
<h3>Index of /tmp</h3> <table border="1"> <thead> <tr> <th>Name</th> <th>Last modified</th> <th>Size</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Parent Directory</td> <td>-</td> <td>-</td> <td></td> </tr> <tr> <td>1764213079@webshell.php</td> <td>2025-11-27 12:11</td> <td>31</td> <td></td> </tr> <tr> <td>1764213327@aa.txt</td> <td>2025-11-27 12:15</td> <td>5</td> <td></td> </tr> <tr> <td>1764213496@webshell2.php</td> <td>2025-11-27 12:18</td> <td>90</td> <td></td> </tr> <tr> <td>1764213611@webshell2.php</td> <td>2025-11-27 12:20</td> <td>90</td> <td></td> </tr> <tr> <td>1764213861@webshell2.php</td> <td>2025-11-27 12:24</td> <td>90</td> <td></td> </tr> <tr> <td>1764221537@weevely.php</td> <td>2025-11-27 14:32</td> <td>519</td> <td></td> </tr> <tr> <td>test.txt</td> <td>2025-11-27 14:42</td> <td>12</td> <td></td> </tr> </tbody> </table> <p>Apache/2.4.52 (Ubuntu) Server at 192.168.30.144 Port 80</p>		Name	Last modified	Size	Description	Parent Directory	-	-		1764213079@webshell.php	2025-11-27 12:11	31		1764213327@aa.txt	2025-11-27 12:15	5		1764213496@webshell2.php	2025-11-27 12:18	90		1764213611@webshell2.php	2025-11-27 12:20	90		1764213861@webshell2.php	2025-11-27 12:24	90		1764221537@weevely.php	2025-11-27 14:32	519		test.txt	2025-11-27 14:42	12	
Name	Last modified	Size	Description																																		
Parent Directory	-	-																																			
1764213079@webshell.php	2025-11-27 12:11	31																																			
1764213327@aa.txt	2025-11-27 12:15	5																																			
1764213496@webshell2.php	2025-11-27 12:18	90																																			
1764213611@webshell2.php	2025-11-27 12:20	90																																			
1764213861@webshell2.php	2025-11-27 12:24	90																																			
1764221537@weevely.php	2025-11-27 14:32	519																																			
test.txt	2025-11-27 14:42	12																																			
조치방안	<ul style="list-style-type: none"> Apache /etc/apache2/apache2.conf 파일 내 DocumentRoot 항목의 Options에서 Indexes 제거 설정 전 <pre><Directory /var/www/> Options FollowSymLinks Indexes AllowOverride None Require all granted </Directory></pre> 설정 후 <pre><Directory /var/www/> Options FollowSymLinks AllowOverride None Require all granted </Directory></pre> 																																				

IL (상)	[WEB] 정보 누출	
취약점 개요		
점검내용	■ 웹 서비스 시 불필요한 정보가 노출되는지 여부 점검	
점검목적	■ 웹 서비스 시 불필요한 정보가 노출되는 것을 방지함으로써 2차 공격에 활용될 수 있는 정보 노출을 차단하기 위함	
점검대상 및 판단기준		
판단기준	<p>양호 : 웹 사이트에 중요정보가 노출되지 않고, 에러 발생 시 과도한 정보가 노출되지 않는 경우</p> <p>취약 : 웹 사이트에 중요정보가 노출되거나, 에러 발생 시 과도한 정보가 노출되는 경우</p>	
진단결과	취약	
점검 및 조치사례		
<p>■ 진단순서</p> <p>Step 1) 웹 사이트에 중요정보가 평문으로 노출되고 있는지 확인 Step 2) 웹페이지에 마스킹 된 중요정보가 웹페이지 소스에 평문으로 노출되고 있는지 확인 Step 3) 에러 메시지 또는 에러 페이지에서 과도한 정보가 노출되는지 확인</p> 		
<h2>Not Found</h2> <p>The requested URL was not found on this server.</p> <p>Apache/2.4.52 (Ubuntu) Server at 192.168.30.144 Port 80</p>		
조치방안	<ul style="list-style-type: none"> Apache /etc/apache2/sites-available/000-default.conf 파일 내 별도의 에러 페이지 추가 <p>설정 후</p> <pre>ErrorDocument 403 /errors/error_default.html ErrorDocument 404 /errors/error_default.html ErrorDocument 500 /errors/error_default.html</pre> 	

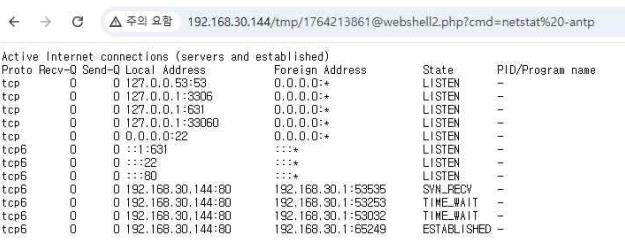
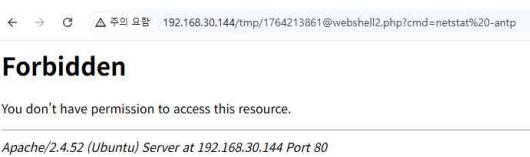
XS (상)	[Web] 크로스사이트 스크립팅												
취약점 개요													
점검내용	■ 웹 사이트 내 크로스사이트 스크립팅 취약점 존재 여부 점검												
점검목적	■ 웹 사이트 내 크로스사이트 스크립팅 취약점을 제거하여 악성 스크립트의 실행을 차단												
점검대상 및 판단기준													
판단기준	<p>양호 : 사용자 입력 값에 대한 검증 및 필터링이 이루어지는 경우</p> <p>취약 : 사용자 입력 값에 대한 검증 및 필터링이 이루어지지 않으며, HTML 코드가 입력·실행되는 경우</p>												
진단결과	취약												
점검 및 조치사례													
<p>■ 진단순서</p> <p>Step 1) 사용자 입력 값을 전달받는 애플리케이션 (회원정보 변경, 게시판, 댓글, 자료실 등)에 스크립트 입력 후 실행되는지 확인</p> 													
<p>1) 사용자 측에서 넘어오는 값을 신뢰하는 모든 form과 파라미터 값에 대해서 필터링을 수행함</p> <p>2) URLDecoder 클래스에 존재하는 decode 메소드를 통해 URL 인코딩이 적용된 사용자 입력 값을 디코딩함으로써 우회 공격 차단</p> <p>※ 필터링 조치 대상 입력 값</p> <ul style="list-style-type: none"> 스크립트 정의어 : <SCRIPT>, <OBJECT>, <APPLET>, <EMBED>, <FORM>, <IFRAME> 등 특수문자 : <, >, ", ', &, %, %00(null) 등 <p>조치방안</p> <p>설정 후 (str_replace 설정)</p> <pre>\$title = str_replace("&", "&amp;", \$title); \$title = str_replace("<", "&lt;", \$title); \$title = str_replace(">", "&gt;", \$title); \$title = str_replace("\\"", "&quot;", \$title); \$title = str_replace("\'", "&#39;", \$title); \$comment = str_replace("&", "&amp;", \$comment); \$comment = str_replace("<", "&lt;", \$comment); \$comment = str_replace(">", "&gt;", \$comment); \$comment = str_replace("\\"", "&quot;", \$comment); \$comment = str_replace("\'", "&#39;", \$comment);</pre> <table border="1" style="float: right;"> <tr><td>subject</td><td>xss 적용 후</td></tr> <tr><td>name</td><td>user1</td></tr> <tr><td>date</td><td>2025-11-30</td></tr> <tr><td>content</td><td>웹 취약점 진단 테스트 <script>alert(document.cookie);</script></td></tr> <tr><td>attachment</td><td></td></tr> <tr><td colspan="2">back</td></tr> </table>		subject	xss 적용 후	name	user1	date	2025-11-30	content	웹 취약점 진단 테스트 <script>alert(document.cookie);</script>	attachment		back	
subject	xss 적용 후												
name	user1												
date	2025-11-30												
content	웹 취약점 진단 테스트 <script>alert(document.cookie);</script>												
attachment													
back													

BF (상)		[Web] 약한 문자열 강도		
취약점 개요				
점검내용	■ 웹페이지 내 로그인 폼 등에 약한 강도의 문자열 사용 여부 점검			
점검목적	■ 유추 가능한 취약한 문자열 사용을 제한하여 계정 및 패스워드 추측 공격을 방지하기 위함			
점검대상 및 판단기준				
판단기준	양호 : 관리자 계정 및 패스워드가 유추하기 어려운 값으로 설정되어 있으며, 일정 횟수 이상 인증 실패 시 로그인을 제한하고 있는 경우 취약 : 관리자 계정 및 패스워드가 유추하기 쉬운 값으로 설정되어 있으며, 일정 횟수 이상 인증 실패 시 로그인을 제한하고 있지 않은 경우			
진단결과	취약			
점검 및 조치사례				
<p>■ 진단순서</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 <p>Username : <input type="text" value="root"/> Password : <input type="password" value="...."/> <input type="button" value="Login"/></p> <p>Step 2) 일정 횟수(3~5회) 이상 인증 실패 시 로그인을 제한하는지 확인</p>				
조치방안	<p>1) 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</p> <p>설정 후 (5회 이상 로그인 실패 시 계정 잠금 설정)</p> <pre> else { if(\$row['try_count'] >= 5) { echo "<script>alert('계정이 잠겼습니다. 관리자에게 문의하세요.');" . "</script>"; echo "<meta http-equiv='refresh' content='0; url=index.php'>"; exit; } if(\$row['password'] === \$password) { \$stmt_reset = mysqli_prepare(\$con, "UPDATE users SET try_count=0 WHERE username=?"); mysqli_stmt_bind_param(\$stmt_reset, "s", \$username); mysqli_stmt_execute(\$stmt_reset); session_start(); \$_SESSION['username'] = \$username; } else { \$stmt_fail = mysqli_prepare(\$con, "UPDATE users SET try_count = try_count + 1 WHERE username=?"); mysqli_stmt_bind_param(\$stmt_fail, "s", \$username); mysqli_stmt_execute(\$stmt_fail); echo "<script>alert('Invalid username or password (Fail: ." . (\$row['try_count']+1) . ". /5)'%;" . "</script>"; } } </pre> <p>192.168.30.144 내용: 계정이 잠겼습니다. 관리자에게 문의하세요.</p> <p style="text-align: right;">확인</p>			

CF (상)		[Web] 크로스사이트 리퀘스트 변조 (CSRF)												
취약점 개요														
점검내용	<ul style="list-style-type: none"> ■ 사용자의 신뢰(인증) 정보의 변조 여부 점검 													
점검목적	<ul style="list-style-type: none"> ■ 사용자 입력 값에 대한 적절한 필터링 및 인증에 대한 유효성을 검증하여 신뢰(인증) 정보 내의 요청(Request)에 대한 변조 방지 													
점검대상 및 판단기준														
판단기준	<p>양호 : 사용자 입력 값에 대한 검증 및 필터링이 이루어지는 경우</p>													
	<p>취약 : 사용자 입력 값에 대한 필터링이 이루어지지 않으며, HTML 코드(또는 스크립트)를 입력하여 실행되는 경우</p>													
진단결과	취약													
점검 및 조치사례														
<p>■ 진단순서</p> <p>Step 1) XSS 취약점이 존재하는지 확인</p> <p>Step 2) 등록 및 변경 등의 데이터 수정 기능의 페이지가 있는지 조사함</p> <p>Step 3) 데이터 수정 페이지에서 전송되는 요청(Request) 정보를 분석하여 임의의 명령을 수행하는 스크립트 삽입 후 해당 게시글을 타 사용자가 열람하였을 경우 스크립트가 실행되는지 확인</p>														
<ul style="list-style-type: none"> • 공격 시 실행될 링크 파일 <pre><!DOCTYPE html> <html> <body> <form action="http://192.168.30.144/write.php" method="POST" enctype="multipart/form-data"> <input type="hidden" name="title" value="CSRF attack success!!" /> <input type="hidden" name="username" value="admin" /> <input type="hidden" name="comment" value="This page is written by CSRF attack ...!" /> <input type="hidden" name="write" value="save" /> </form> <script> document.forms[0].submit(); </script> </body> </html></pre> <p>content Global Hacking issue..! attachment 파일 선택 선택된 파일 없음 save reset</p> <table border="1"> <tr> <td>16</td> <td>CSRF attack success!!</td> <td>admin</td> <td>2025-11-27</td> </tr> <tr> <td>15</td> <td>CSRF attack success!!</td> <td>admin</td> <td>2025-11-27</td> </tr> <tr> <td>14</td> <td>csrf</td> <td>user1</td> <td>2025-11-27</td> </tr> </table>			16	CSRF attack success!!	admin	2025-11-27	15	CSRF attack success!!	admin	2025-11-27	14	csrf	user1	2025-11-27
16	CSRF attack success!!	admin	2025-11-27											
15	CSRF attack success!!	admin	2025-11-27											
14	csrf	user1	2025-11-27											
조치방안	<ul style="list-style-type: none"> • 사용 중인 프레임워크에 기본적으로 제공되는 CSRF 보호 기능 사용 • 사용자가 정상적인 프로세스를 통해 요청하였는지 Referer 검증 로직 구현 • 정상적인 요청(Request)과 비정상적인 요청(Request)를 구분할 수 있도록 Hidden Form 을 사용하여 임의의 암호화된 토큰(세션 ID, Timestamp, nonce 등)을 추가하고 이 토큰을 검증하도록 설계 • HTML이나 자바스크립트에 해당되는 태그 사용을 사전에 제한하고, 서버 단에서 사용자 입력 값에 대한 필터링 구현 • XSS 조치 방안 참조 													

IN (상)		[Web] 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											
점검대상 및 판단기준												
판단기준	양호 : 접근제어가 필요한 중요 페이지의 통제수단이 적절하여 비인가자의 접근이 불가능한 경우 취약 : 접근제어가 필요한 중요 페이지의 통제수단이 미흡하여 비인가자의 접근이 가능한 경우											
진단결과	취약											
점검 및 조치사례												
<p>■ 진단순서</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 것으로 다른 사용자의 비밀 게시글 (또는 개인정보 변경, 패스워드 변경 등)에 접근 가능한지 확인</p> <table border="1"> <thead> <tr> <th>Time</th> <th>Type</th> <th>Direction</th> <th>Method</th> <th>URL</th> </tr> </thead> <tbody> <tr> <td>02:21:18 D...</td> <td>H...</td> <td>Response</td> <td>GET</td> <td>http://192.168.30.144/view.php?id=7</td> </tr> </tbody> </table> <p>Request</p> <pre> 1 GET /view.php?id=22 HTTP/1.1 2 Host: 192.168.30.144 3 Cache-Control: max-age=0 4 Accept-Language: en-US,en;q=0.9 5 Upgrade-Insecure-Requests: 1 6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36 7 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image /avif,image/webp,image/apng,*/*;q=0.8,application/signed-ex change;v=b3;q=0.7 8 Referer: http://192.168.30.144/index.php 9 Accept-Encoding: gzip, deflate, br 10 Cookie: PHPSESSID=8pmrskpl0u9u9u9mjqjrs075g5 11 Connection: keep-alive 12 13 </pre> <p>Response</p> <pre> 28 <td align="center"> 29 subject 30 </td> 31 <td align="center"> 32 IN 33 </td> 34 <td align="center"> 35 name 36 </td> 37 <td align="center"> 38 user1 39 </td> 40 <td align="center"> 41 content 42 </td> 43 <td align="center"> 44 Insufficient Authorization 45 </td> 46 </tr> 47 <tr> 48 <td align="center"> 49 content 50 </td> 51 <td align="center"> 52 back 53 </td> 54 <td align="center"> 55 location.href='index.php' 56 </td> 57 </tr> 58 </table> 59 <input type="button" value="back" onclick= 60 location.href='index.php'> 61 </body> 62 </html> </pre>			Time	Type	Direction	Method	URL	02:21:18 D...	H...	Response	GET	http://192.168.30.144/view.php?id=7
Time	Type	Direction	Method	URL								
02:21:18 D...	H...	Response	GET	http://192.168.30.144/view.php?id=7								
조치방안	<ul style="list-style-type: none"> 접근제어가 필요한 중요 페이지는 세션을 통한 인증 등 통제수단을 구현하여 인가된 사용자 여부를 검증 후 해당 페이지에 접근할 수 있도록 함 페이지별 권한 매트릭스를 작성하여 접근제어가 필요한 모든 페이지에서 권한 체크가 이뤄지도록 구현하여야 함 											

SF (상)	[Web] 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								
점검대상 및 판단기준									
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								
진단결과	취약								
점검 및 조치사례									
<p>■ 진단순서</p> <p>Step 1) 로그인 시(1) 세션 ID가 발행되는지 확인하고 로그아웃 후 다시 로그인(2)할 때 예측 불가능한 새로운 세션 ID가 발급되는지 확인</p> <p>(1) Request cookies</p> <table border="1" style="margin-bottom: 10px;"> <thead> <tr> <th>Name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>PHPSESSID</td> <td>8pnrskp10u9u9u9nmqjrs075g5</td> </tr> </tbody> </table> <p>(2) Request cookies</p> <table border="1" style="margin-top: 10px;"> <thead> <tr> <th>Name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>PHPSESSID</td> <td>8pnrskp10u9u9u9nmqjrs075g5</td> </tr> </tbody> </table>		Name	Value	PHPSESSID	8pnrskp10u9u9u9nmqjrs075g5	Name	Value	PHPSESSID	8pnrskp10u9u9u9nmqjrs075g5
Name	Value								
PHPSESSID	8pnrskp10u9u9u9nmqjrs075g5								
Name	Value								
PHPSESSID	8pnrskp10u9u9u9nmqjrs075g5								
조치방안	<ul style="list-style-type: none"> 로그인할 때마다 예측 불가능한 새로운 세션 ID를 발급받도록 해야 하고 기존 세션 ID는 파기해야 함 <p>설정 후 (session_regenerate_id(true); 설정)</p> <pre>session_start(); session_regenerate_id(true); \$_SESSION['username'] = \$username;</pre> <p>Request cookies</p> <table border="1" style="margin-top: 10px;"> <thead> <tr> <th>Name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>PHPSESSID</td> <td>v2u8tfpvqkq8tfqjc248ekf99h</td> </tr> </tbody> </table>	Name	Value	PHPSESSID	v2u8tfpvqkq8tfqjc248ekf99h				
	Name	Value							
PHPSESSID	v2u8tfpvqkq8tfqjc248ekf99h								

FU (상)	[Web] 파일 업로드										
취약점 개요											
점검내용	■ 웹 사이트의 게시판, 자료실 등에 조작된 Server Side Script 파일 업로드 및 실행 가능 여부 점검										
점검목적	■ 업로드되는 파일의 확장자에 대한 적절성 여부를 검증하는 로직을 통해 공격자가 조작된 Server Side Script 파일 업로드 방지 및 서버상에 저장된 경로를 유추하여 해당 Server Side Script 파일 실행을 불가능하게 하기 위함										
점검대상 및 판단기준											
판단기준	양호 : 업로드되는 파일에 대한 확장자 검증이 이루어지는 경우 취약 : 업로드되는 파일에 대한 확장자 검증이 이루어지지 않는 경우										
진단결과	취약										
점검 및 조치사례											
<p>■ 진단순서</p> <p>Step 1) 웹 사이트에 파일 업로드 기능이 존재하는 경우, 확장자가 jsp, php, asp, cgi 등 Server Side Script 파일들이 업로드 가능한지 확인</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="padding: 2px;">subject</td><td style="padding: 2px;">testy23</td></tr> <tr><td style="padding: 2px;">name</td><td style="padding: 2px;">user1</td></tr> <tr><td style="padding: 2px;">date</td><td style="padding: 2px;">2025-11-27</td></tr> <tr><td style="padding: 2px;">content</td><td style="padding: 2px;">ytr</td></tr> <tr><td style="padding: 2px;">attachment</td><td style="padding: 2px;">webshell2.php</td></tr> </table>		subject	testy23	name	user1	date	2025-11-27	content	ytr	attachment	webshell2.php
subject	testy23										
name	user1										
date	2025-11-27										
content	ytr										
attachment	webshell2.php										
<p>Step 2) 웹 사이트에 있는 디렉터리 정보를 이용하여 첨부한 Server Side Script 파일의 위치를 조사한 후 브라우저 주소창에 해당 경로를 입력하여 실행 가능한지 확인</p> <p></p>											
조치방안	<p>1) 화이트 리스트 방식으로 허용된 확장자만 업로드 가능토록 서버 측 통제 적용</p> <p>2) 업로드되는 파일을 디렉터리에 저장할 때 파일명과 확장자를 외부 사용자가 추측할 수 없는 문자열로 변경하여 저장 (파일 이름은 DB에 저장)</p> <ul style="list-style-type: none"> • Apache <p>FileMatch 지시자를 이용하여 *.ph, *.inc, *lib 등의 Server Side Script 파일에 대해서 직접 URL 호출을 금지시킴</p> <p></p> <p></p>										

FD (상)		[Web] 파일 다운로드		
취약점 개요				
점검내용	■ 웹 사이트에서 파일 다운로드 시 허용된 경로 외 다른 경로의 파일 접근이 가능한지 여부 점검			
점검목적	■ 파일 다운로드 시 허용된 경로 외 다른 경로의 파일 접근을 방지하여 공격자가 임의의 위치에 있는 파일을 열람하거나 다운받는 것을 불가능하게 하기 위함			
점검대상 및 판단기준				
판단기준	양호 : 다운로드 파일이 저장된 디렉터리 이외에 접근이 불가능한 경우 취약 : 다운로드 파일이 저장된 디렉터리 이외에 접근이 가능한 경우			
진단결과	취약			
점검 및 조치사례				
<p>■ 진단순서</p> <p>Step 1) 웹 사이트에 cgi, jsp, php 등의 애플리케이션을 이용하여 파일을 다운받는 페이지가 있는지 조사</p> <p>Step 2) 웹 사이트에서 파일 다운로드 시 요청(Request) 정보에 파일 경로를 웹 서버(웹 사이트 포함) 중요 파일(winnt\win.ini, /etc/passwd 등)의 상대 경로(..)로 치환한 후 전송했을 때 해당 경로 파일들을 다운로드 가능한지 확인</p> <p>ex) ../../../../../../../../../../etc/passwd</p> <p>Request</p> <pre>Pretty Raw Hex 1 GET /download.php?file=/var/www/html/tmp/../../../../../../../../etc/passwd HTTP/1.1 2 Host: 192.168.30.144</pre> <p>download.php ~Downloads</p> <pre>Open Save ... 1 root:x:0:0:root:/root:/bin/bash 2 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin 3 bin:x:2:2:bin:/bin:/usr/sbin/nologin</pre>				
<ul style="list-style-type: none"> 다운로드 애플리케이션 소스 파일을 수정하여 파일을 다운받을 수 있는 디렉터리를 특정 디렉터리로 한정하고 이 외의 다른 디렉터리에서는 파일을 다운받을 수 없도록 설정해야 함 PHP를 사용하는 경우 php.ini에서 magic_quotes_gpc를 On으로 설정하여 .\./ 와 같은 역슬러시 문자 입력 시 치환되도록 설정 <p>설정 후</p> <pre><?php require_once("./file.php"); if(isset(\$_GET['file'])){ \$file_name = \$_GET['file']; \$filter = array('\\', '/', '%'); \$filtered_name = str_replace(\$filter, '', \$file_name); \$path = "./tmp/" . \$filtered_name; if(file_exists(\$path)){ file_download(\$filtered_name); } else{ echo "<script>alert('Wrong Filename'); history.back();</script>"; } } ?></pre> <p>192.168.30.144 says Wrong Filename</p> <p style="text-align: center;">OK</p>				
조치방안				