




# UIUCTF log-action Write-up

32기 유연서

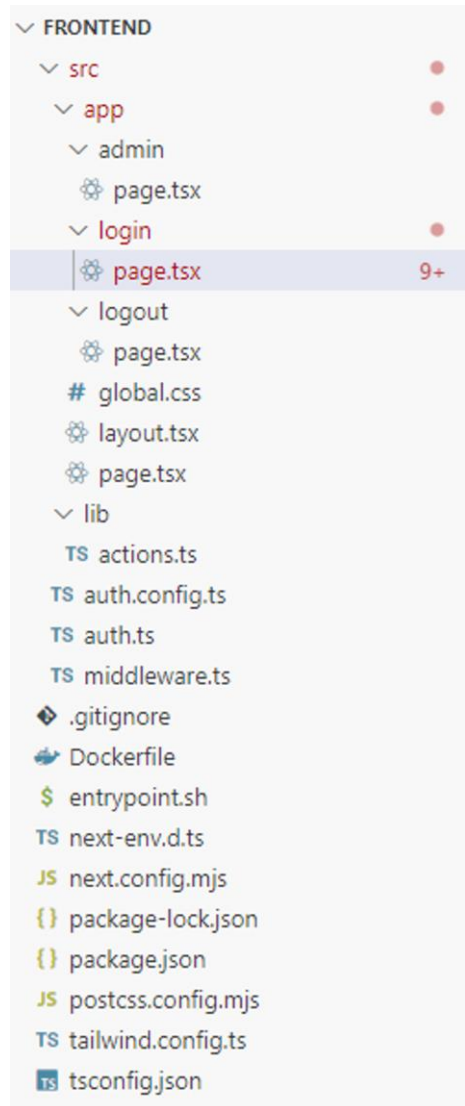
내가 풀어볼 ctf는 uiuctf의 web부분 log-action이다. log-action을 고른 이유는 저번 드림핵에서 풀어봤던 ctf문제도 log-action과 관련된 문제였기 때문에 그나마 가장 익숙해 보였기 때문이다.

<input type="checkbox"/> 이름	수정한 날짜	유형	크기
 backend	2024-06-28 오후 9:53	파일 폴더	
 frontend	2024-06-28 오후 9:53	파일 폴더	
 docker-compose	2024-06-29 오후 1:53	Yaml 원본 파일	1KB

파일을 다운받고 압축을 해제해주었다. backend와 frontend파일이 보인다.

일단! 프론트엔드는 직접적으로 이용자들의 눈에 보이는 영역, 백엔드는 사용자가 원하는 일을 에러가 나지 않고 원활하게 실행되도록 하는 영역이다.

이 파일들을 장고 스터디를 위해 깔았었던 비주얼스튜디오 코드로 열어보았다. 저번 스터디에서 처음으로 비주얼스튜디오 코드를 사용해보았었는데, 웹페이지 코드를 만들때 폴더와 파일단위로 뭔가 코드를 작성해서 뼈대를 이루는 것 같았기 때문에.. 이 폴더를 비주얼 스튜디오 코드로 열면 더 보기 편하지 않을까! 싶었기 때문이다. 그리고 애초에 web부분 문제이기때문이라는 이유도 있었다.



src는 source의 줄임말이다. 따라서 이 밑의 폴더들은 모두 소스코드일것이다. 이 아래에는 admin,login,logout이라는 이름의 폴더와 코드들이 있다. 각각의 이름에 맞는 일을 수행하는 코드이려나? 싶었다.

일단 이 ctf문제의 소개글이 로그인에 자꾸 오류난다! 였기 때문에 login코드부터 살펴보았다.

```

1  "use client";
2  import { useState } from "react";
3  import { authenticate } from "@lib/actions";
4
5  export default function LoginPage() {
6    const [pending, setPending] = useState(false);
7    const [error, setError] = useState(null);
8
9    const handleSubmit = async (event) => {
10      event.preventDefault();
11      setPending(true);
12      setError(null);
13
14      const formData = new FormData(event.target);
15
16      try {
17        const result = await authenticate(formData);
18        if (result !== 'Something went wrong.') {
19          window.location.href = "/admin";
20        } else {
21          setError(result);
22        }
23      } catch (error) {
24        setError('An unexpected error occurred.');
```

```

25      } finally {
26        setPending(false);
27      }
28    };
29
30    return (
31      <div className="max-w-prose mx-auto flex flex-col gap-4">
32        <p className="text-2xl font-bold">
33          Login
34        </p>
35        <form onSubmit={handleSubmit} className="flex flex-col gap-4">
36          <label className="flex flex-col gap-1">
37            <span>Username</span>
38            <input
39              type="text"
40              id="username"
41              name="username"
42              placeholder="Enter your username"
43              required
44              minLength={3}
45            />
46          </label>
47          <label className="flex flex-col gap-1">
48            <span>Password</span>
49            <input
50              type="password"
51              id="password"
52              name="password"
53              placeholder="Enter your password"
54              required
55              minLength={10}
56            />
57          </label>
58          <button type="submit" disabled={pending}>
59            {pending ? 'Logging in...' : 'Log in'}
60          </button>
61          {error && (
62            <span className="text-sm text-red-500">
63              {error}
64            </span>
65          )}
66        </form>
67      </div>
68    );
69  }

```

을 살펴보았는데, 사용자가 로그인 정보를 입력하고 제출하면 해당 정보를 서버로 보내서 인증을 시도하는 코드인 것 같았고 수상한 부분이 딱히 없다고 느껴졌다. 다음으로 로그아웃 코드를 살펴보았다.

```
1  import Link from "next/link";
2  import { redirect } from "next/navigation";
3  import { signOut } from "@/auth";
4
5  export default function Page() {
6    return (
7      <>
8        <h1 className="text-2xl font-bold">Log out</h1>
9        <p>Are you sure you want to log out?</p>
10       <Link href="/admin">
11         Go back
12       </Link>
13       <form
14         action={async () => {
15           "use server";
16           await signOut({ redirect: false });
17           redirect("/login");
18         }}
19     >
20       <button type="submit">Log out</button>
21     </form>
22   </>
23 )
24 }
```

사용자가 로그아웃을 원하면 log out 버튼을 클릭하여 폼이 제출되고 이후 서버 측에서 signOut 함수가 호출되고 로그아웃 처리가 되는 것 같다. 이후 완료되면 사용자가 /login 페이지로 리디렉션되도록 한다.

이 문제의 목표는 로그인 과정에서 잠재적인 보안취약점을 찾기이다. 목표를 달성하진 못했지만 이번 시도에서 배운 점과 느낀 점을 써보도록 하자

1. 로그와 관련된 취약점들엔 무엇이 있는가에 대해 알아보아야한다.

(들어는 본것: \*\*SQL Injection 등등 이것 말고도 굉장히 많은 것들이 있었다. XSS CSRF 등등 정말 많은 보안관련 공격 기법들이 있었는데 이런것들을 차근차근 알아가야만 한다.)\*\*

## 1. 앞으로 웹 ctf문제를 풀기 위해서 필요한 역량

### - 웹 기술 기초

—> **\*\*HTML/CSS\*\*** 웹 페이지의 구조와 스타일을 이해하기 위한 기본 지식

—> **\*\*JavaScript\*\*** 웹 애플리케이션의 동작을 이해하고 분석하기 위해

—> **\*\*HTTP/HTTPS\*\*** 웹 통신 프로토콜의 동작 방식을 이해하고, 요청과 응답을 분석하기 위해!

제발 ctf문제에 대한 역량을 길러야 할 것 같다. ctf문제를 풀고 아무것도 모르는 상태로 라이트업을 작성하고 싶지 않다