

□ 단국대학교 > 사이버보안 학과 > 인터넷 보안 > 보고서 템플릿

25년 1학기 인터넷 보안 수업

세션 고정 & 파라미터 변조

2025년 5월 20일

학번 : 32231594
이름 : 박기쁨

1. 워게임(세션·쿠키 & 정보 누출) 1 번 문항 : 타 사용자의 세션으로 변조

<과정 설명>

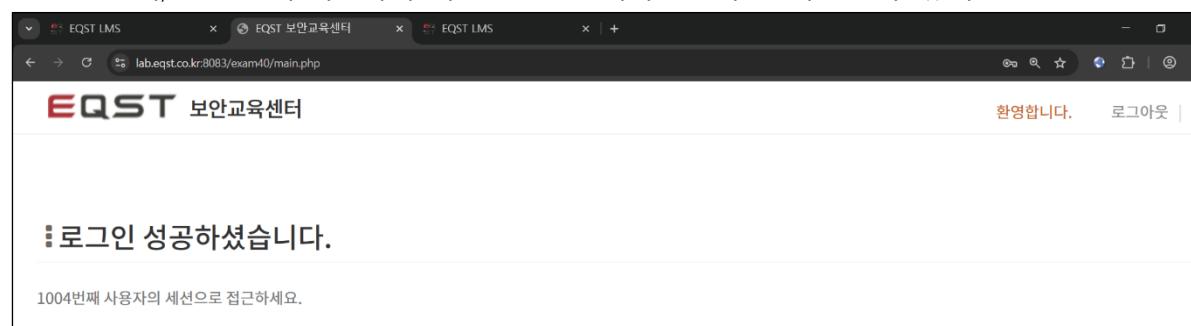
쿠키(Cookie)는 Stateless 프로토콜인 HTTP 통신 시, 서버가 사용자 정보를 유지하기 위하여 응답 헤더에 붙여 클라이언트에게 전송하는 정보를 뜻한다. 세션·쿠키 취약점 공격은, 쿠키 속 타 사용자의 세션 ID를 탈취하여 해당 사용자의 권한을 획득하는 공격이다.

본 실습에서는, 쿠키 속 세션 ID를 변조하여 타 사용자의 세션으로 페이지에 접근해보고자 한다.

Step 1. Cookie 의 session ID 변조하기

Step 1-1. Cookie 의 session ID 확인

: 로그인 시, 1004 번째 사용자의 세션으로 접근하라는 문구를 확인할 수 있다.



: Burp Suite 를 이용하여 해당 페이지를 intercept 하여 Cookie 의 session ID 를 확인한다.

(session ID=eqst001822)

A screenshot of the Burp Suite interface. The "Proxy" tab is selected. A request for "https://lab.eqst.co.kr:8083/exam40/main.php" is shown in the list. The "Inspector" panel on the right shows the "Selected text" field containing "eqst001822". Below it, the "Decoded from" dropdown is set to "URL encoding" and shows "eqst001822". Other panels like "Request" and "Response" are also visible.

Step 1-2. session ID 를 1004 번째 사용자의 세션으로 변조

: intercept 한 Cookie 의 session ID 를 1004 번째 사용자의 세션으로 변조한다.

(eqst001822 -> eqst001004)

The screenshot shows the Burp Suite interface in the 'Proxy' tab. A specific cookie entry for session 1004 is highlighted in the list. The cookie value 'eqst001004' is selected in the 'Selected text' field of the 'Inspector' panel. The 'Decoded from' dropdown is set to 'URL encoding'. The URL in the list is 'https://lab.eqst.co.kr:8083/exam40/main.php'.

: 로그인에 성공하고 정답을 획득한다. (정답: cookie_answer)

The screenshot shows a browser window with three tabs: 'EQST LMS', 'EQST 보안교육센터', and 'EQST LMS'. The middle tab displays the login page for 'EQST 보안교육센터'. The status bar at the bottom of the browser says '환영합니다.' (Welcome). The page content includes a message '로그인 성공하셨습니다.' (Login successful) and the text '정답: cookie_answer'.

2. 워게임(파라미터 변조) 3 번 문항 : 인증 취약점을 통한 main 페이지 접속

<과정 설명>

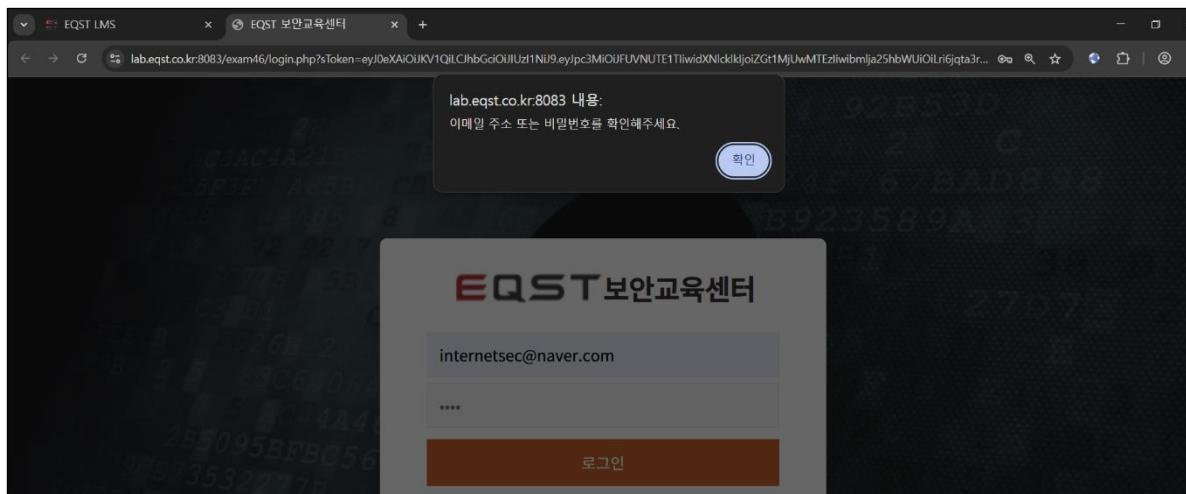
파라미터 변조 취약점 공격은, 패킷을 통해 전달되는 여러 파라미터들을 변조하여 정상 경로 및 인증 과정을 우회하는 공격이다.

본 실습에서는 인증 취약점을 이용하여, login 페이지의 정상적인 동작 경로를 피해 로그인에 성공해보자 한다.

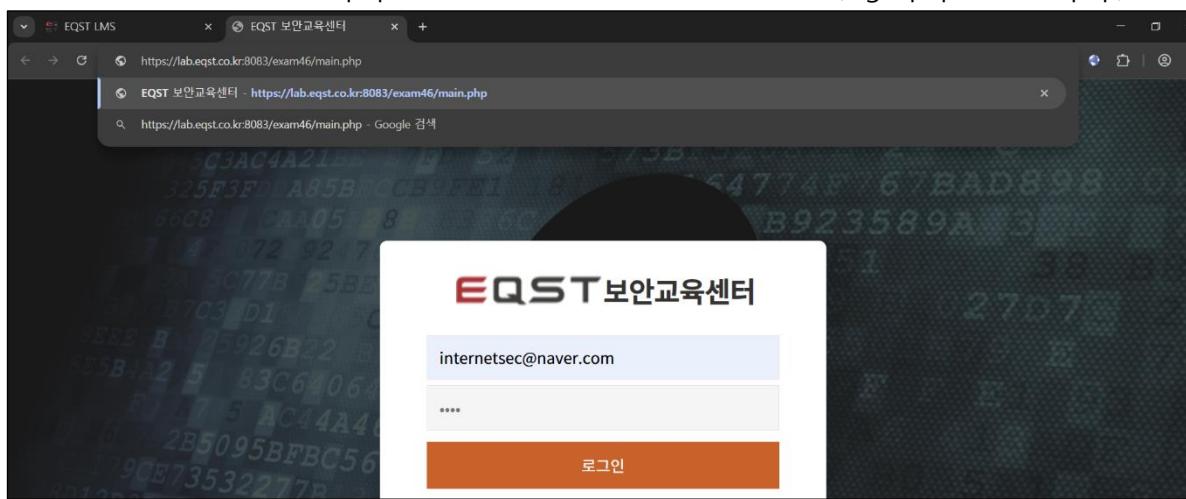
Step 1. 인증 취약점을 통한 main 페이지 접속

Step 1-1. 인증 취약점을 통한 main 페이지 접속

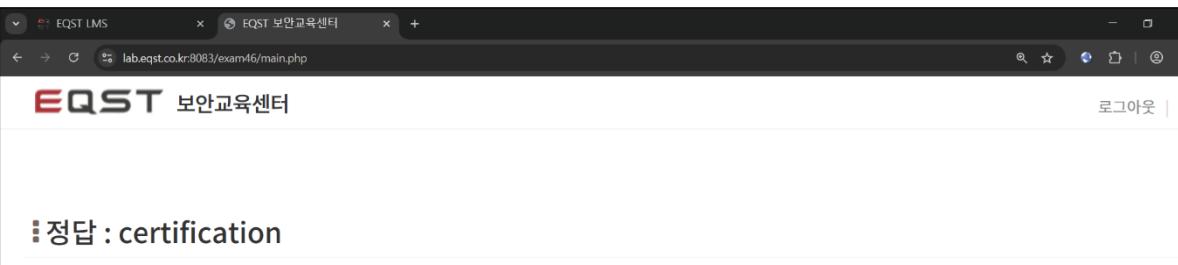
: 정상적인 로그인 ID 와 PW 를 입력하였을 때, 로그인에 실패하며 알림창을 띄운다.



: URL 창을 이용하여 main.php 페이지로 직접 접근을 시도해본다. (login.php -> main.php)



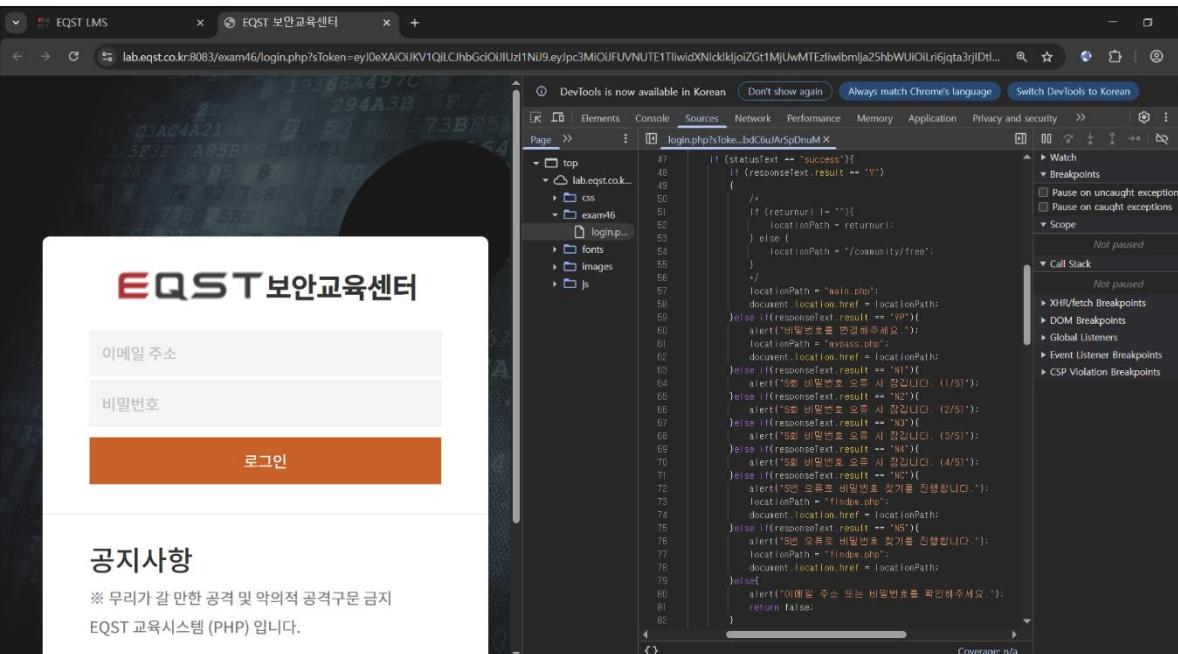
: main 페이지 접속에 성공하여 정답을 획득한다. (정답: certification)



Step X. 기타 방식

Step X-1. login.php 페이지의 소스 코드를 확인한다.

: result 가 Y이면 main.php로, 이외의 경우 이하 코드로 이어지는 것을 확인할 수 있다.



Step X-2. result 값 변조

: 해당 페이지를 intercept 하여 response의 result 값을 확인한다. (현재 result 값은 N)

Request

```
POST /exam46/process/loginProcess.php HTTP/1.1
Host: lab.eqst.co.kr:8083
Cookie: PHPSESSID=00040574c08a470c7080832ad18e9cd5
Content-Length: 59
Sec-CH-Ua-Platform: "Windows"
Sec-CH-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36
Accept: application/json, text/javascript, */*; q=0.01
Sec-Ch-Ua: "Chromium";v="108", "Google Chrome";v="108", "Not/A/Brand";v="99"
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Sec-Ch-Ua-View-Width: 768
Origin: https://lab.eqst.co.kr:8083
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer:
https://lab.eqst.co.kr:8083/exam46/login.php?Tokenkey=J0XW1Q2PQ1Q1QJh6c1Q1U1U1I1j9j eyj pcSMiOJFpNUtESTIiinid001ch1k1jpi2t1#jh1Ec1iinib1jx25hM0iQ1r1digt1a1j1D12nq12k1My1s1m1 hc1C1M1ch1fc1z1j4s8n1zKh1jpi2t1#h238ckz#0Q21Q_H0BEJ1dqEnY800tr1f0#1w0d0s1ST1n8PHg1-Dr1f0
Accept-Encoding: gzip, deflate, br
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
Priority: u1, i
Connection: keep-alive
login_id=internetsec40naver.com@login_gud+np@9100LH1u09kc
```

Response

```
HTTP/1.1 200 OK
Date: Wed, 21 May 2025 02:04:18 GMT
Server: Apache/2.4.41 (Ubuntu) OpenSSL/1.1.1d
X-Powered-By: PHP/7.0.33
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
Content-Length: 18
14
15 (*result*: "N")
```

Inspector

Selected text
"N"

Decoded from: HTML encoding
"N"

: result 값을 Y로 변조한다. (N -> Y)

Request

```
POST /exam46/process/loginProcess.php HTTP/1.1
Host: lab.eqst.co.kr:8083
Cookie: PHPSESSID=00040574c08a470c7080832ad18e9cd5
Content-Length: 59
Sec-CH-Ua-Platform: "Windows"
Sec-CH-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36
Accept: application/json, text/javascript, */*; q=0.01
Sec-Ch-Ua: "Chromium";v="108", "Google Chrome";v="108", "Not/A/Brand";v="99"
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Sec-Ch-Ua-View-Width: 768
Origin: https://lab.eqst.co.kr:8083
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer:
https://lab.eqst.co.kr:8083/exam46/login.php?Tokenkey=J0XW1Q2PQ1Q1QJh6c1Q1U1U1I1j9j eyj pcSMiOJFpNUtESTIiinid001ch1k1jpi2t1#jh1Ec1iinib1jx25hM0iQ1r1digt1a1j1D12nq12k1My1s1m1 hc1C1M1ch1fc1z1j4s8n1zKh1jpi2t1#h238ckz#0Q21Q_H0BEJ1dqEnY800tr1f0#1w0d0s1ST1n8PHg1-Dr1f0
Accept-Encoding: gzip, deflate, br
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
Priority: u1, i
Connection: keep-alive
login_id=internetsec40naver.com@login_gud+np@9100LH1u09kc
```

Response

```
HTTP/1.1 200 OK
Date: Wed, 21 May 2025 02:04:18 GMT
Server: Apache/2.4.41 (Ubuntu) OpenSSL/1.1.1d
X-Powered-By: PHP/7.0.33
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
Content-Length: 18
14
15 (*result*: "Y")
```

Inspector

Selected text
"Y"

Decoded from: HTML encoding
"Y"

: 로그인에 성공하여 result 값을 획득한다. (정답: certification)

EQST 보안교육센터

로그아웃

정답 : certification

성명	프로젝트 후 소감
박기쁨	<p>평소 인터넷을 사용하며 쿠키가 정확히 어떤 역할을 하는지, 세션이 만료되었다는 것은 어떤 의미인지 궁금했는데, 이번 수업과 실습을 통해 쿠키와 세션에 대해 정확히 배우고 이해할 수 있었던 것 같다. 주요정보를 저장해두는 등 웹을 편리하게 사용하고자 하면 할수록, 보안성은 현저히 떨어진다는 것을 직접 깨달은 것 같다.</p> <p>조금 다른 이야기이지만, 각 취약점에 대한 보안 대책을 배운 후에 항상 드는 생각이 있다. '이미 안전한 보안 대책을 찾아버린 취약점에 대해서는 배운 공격 방법을 적용할 수 없을 것이고, 대다수의 웹 페이지들은 잘 알려진 취약점과 보안 대책을 이미 고려한 후 개발되었을 텐데, 실무에서 내가 과연 무엇을 할 수 있을까?'</p>