□ 단국대학교 > 사이버보안 학과 > 인터넷 보안 > 보고서 템플릿

25년 1학기 인터넷 보안 수업

세션 고정 및 파라미터 변조

2025년 5월 20일

학번 : 32230324
이름 : 김가람

과정 설명

<세션 고정>

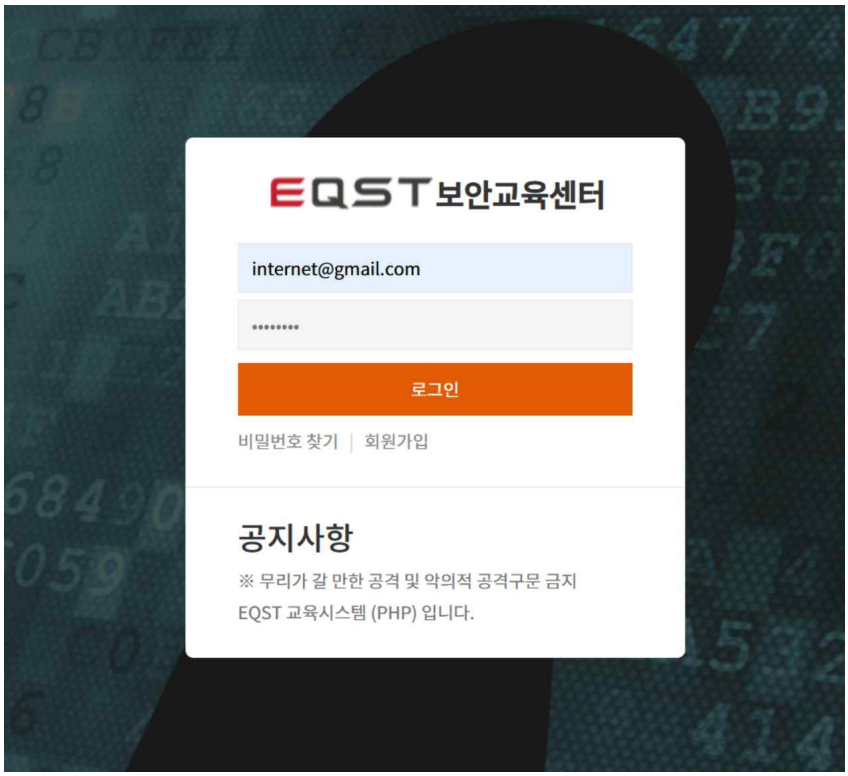
세션이란 웹 애플리케이션에서 사용자 상태를 유지하기 위해 쓰이는 것으로 이를 노린 다양한 취약점이 존재함. 대표적인 취약점으로는 사용자의 세션 ID가 로그인 후에도 변경되지 않는 경우와 같은 세션 고정 취약점, 로그아웃 후에도 세션 ID가 유지되는 경우와 같은 불충분한 세션 만료 취약점, 세션 ID가 단순하거나 패턴이 예측 가능할 경우의 세션 예측 등이 있음. 이러한 세션 취약점을 대응하는 방법으로는 로그인 시 세션 ID 변경, 로그아웃 시 세션 ID 변경(제거), https 사용을 통한 통신 구간 보호, 유추하기 어려운 세션 ID 사용, Cookie 보안 설정(HTTPOnly(서버가 요청할 때만 서버로 보냄) 등)가 있음.

<실습>

<Q1 - 1004번째 사용자의 세션으로 접근하기>

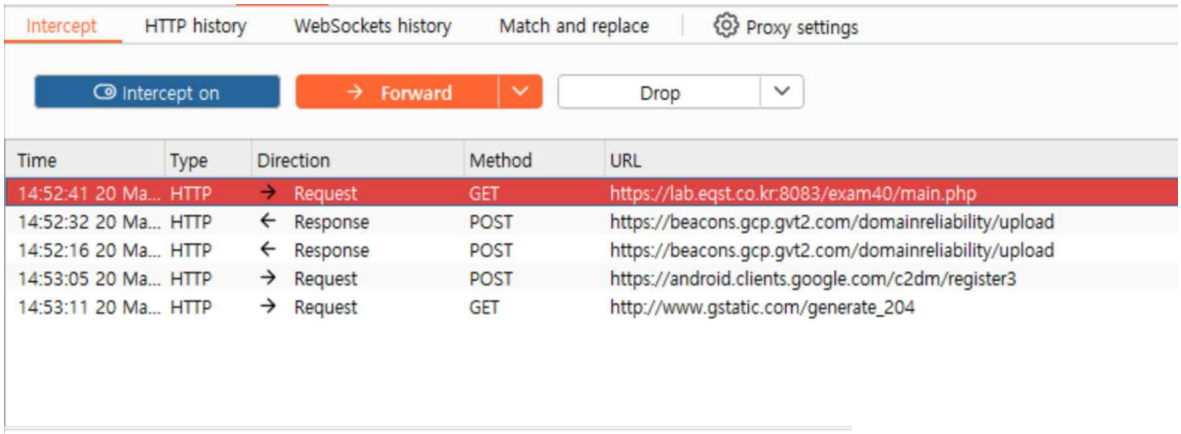
Step 1. 로그인 하기

: 기존에 알고 있는 계정으로 사이트에 로그인 하면 로그인에 성공하였다는 페이지로 연결됨.



Step 2. Intercept

: Step 1에서 접속한 로그인 성공 페이지를 BurpSuite에서 Intercept하여 Request를 확인해보면 sessionID가 포함되어 있는 것을 확인할 수 있음.

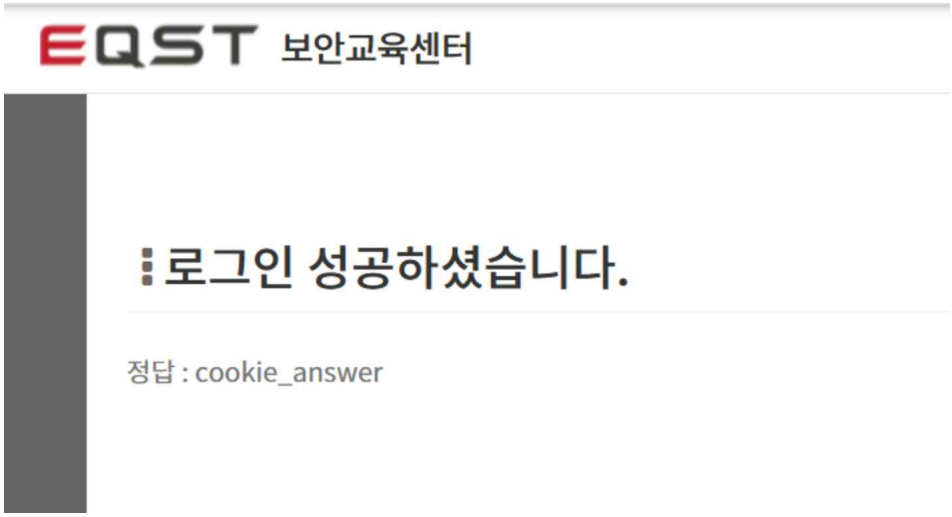


Step 3. SessionID 조작

: Step 2에서 발견한 SessionID의 값을 살펴보면 eqst에 6자리 숫자(회원 번호)가 결합된 형식임을 추정할 수 있음. 이에 따라 접근하고자 하는 1004번째 사용자의 sessionID는 eqst001004로 판단되며 기존 sessionID인 eqst261158을 해당 값으로 변경 후 Intercept를 off하여 요청을 전송함.

Step 4. 정답 확인

: Step 3의 결과로 로그인 성공 페이지에 정답 (cookie_answer)이 나타남.



<파라미터 변조>

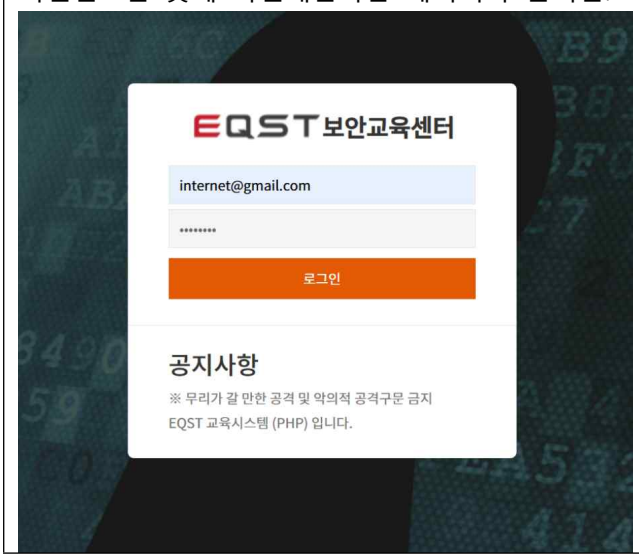
파라미터 변조란 클라이언트 측에서 서버로 전송되는 파라미터 값을 공격자가 임의로 조작하여 시스템의 정상 작동을 우회하거나 정보를 탈취하는 취약점임. url 접속, 패스워드 입력 우회, 프로세스 검증 누락과 같은 불충분한 사용자 인증, url 접속을 이용한 서비스 사용, 권한 파라미터 변조와 같은 불충분한 사용자 인가가 대표적임. 이에 대한 보안 대책으로는 세션에서 사용자가 로그인된 사용자인지 검증하는 것, 주요 페이지 접근 전에 재인증하는 것, HTML에 주요 정보를 임시로 저장하게 될 경우 암호화하는 것 등이 있음.

<실습>

<Q3 - 로그인이 불가한 사이트에서 인증 취약점을 통해 main 페이지의 정답 획득>

Step 1. 로그인 시도

: 로그인을 시도할 경우 ID와 비밀번호를 맞게 입력하였음에도 불구하고 이메일 주소와 비밀번호를 맞게 확인해달라는 메시지가 출력됨.



Step 2. 소스 코드 분석

: 개발자 도구(F12)를 통해 로그인 페이지(login.php)의 소스 코드를 살펴보면 아래와 같은 조건문이 포함되어 있는 것을 확인할 수 있음. 해당 조건문을 분석해보면 다음과 같은 동작을 수행함을 알 수 있음.

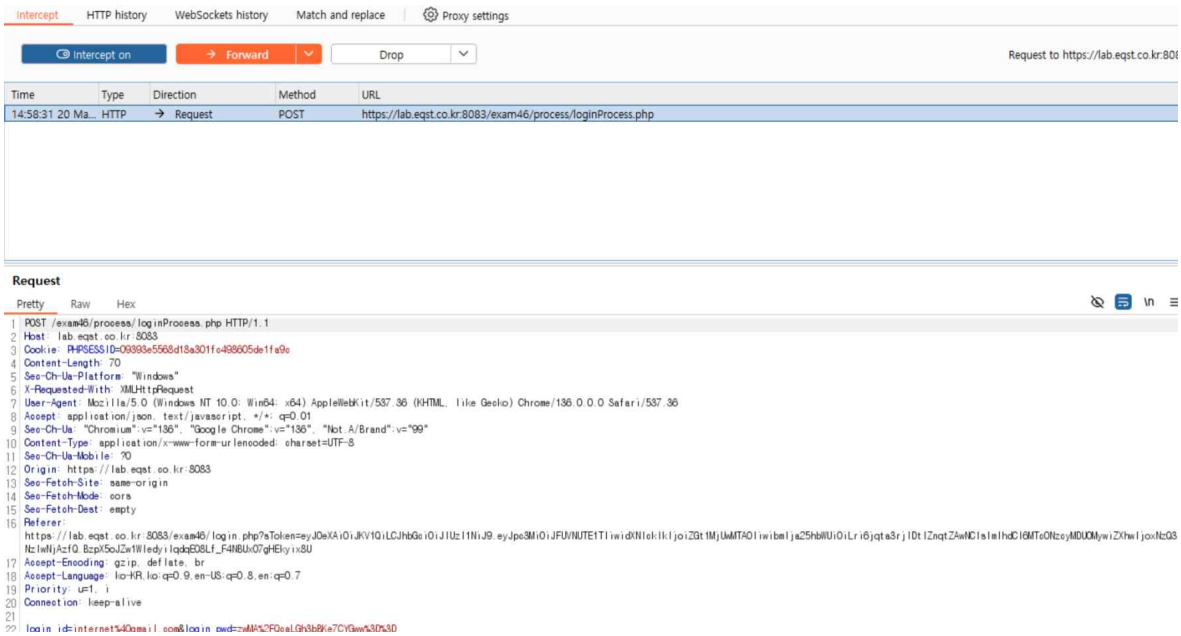
- result 값이 "Y"인 경우, 사용자를 main.php로 이동함.
- result 값이 "YP"인 경우, 비밀번호를 변경해달라는 메시지 출력과 함께 mypass.php로 이동함.
- result 값이 "N1"이나 "N2" 등의 경우, 5회 비밀번호 오류 시 잠긴다는 안내 메시지와 함께 로그인 시도 횟수가 출력됨.
- result 값이 "NC"인 경우, 5번 오류로 비밀번호 찾기를 진행한다는 메시지와 함께 findpw.php로 이동함.
- 위 경우들 모두 해당되지 않는 경우, 이메일 주소 또는 비밀번호를 확인해달라는 메시지가 출력됨.

```
47      if (statusText == "success"){
48          if (responseText.result == "Y")
49          {
50              /*
51              if (returnurl != ""){
52                  locationPath = returnurl;
53              } else {
54                  locationPath = "/community/free";
55              }
56              */
57              locationPath = "main.php";
58              document.location.href = locationPath;
59          }else if(responseText.result == "YP"){
60              alert("비밀번호를 변경해주세요.");
61              locationPath = "mypass.php";
62              document.location.href = locationPath;
63          }else if(responseText.result == "N1"){
64              alert("5회 비밀번호 오류 시 잠깁니다. (1/5)");
65          }else if(responseText.result == "N2"){
66              alert("5회 비밀번호 오류 시 잠깁니다. (2/5)");
67          }else if(responseText.result == "N3"){
68              alert("5회 비밀번호 오류 시 잠깁니다. (3/5)");
69          }else if(responseText.result == "N4"){
70              alert("5회 비밀번호 오류 시 잠깁니다. (4/5)");
71          }else if(responseText.result == "NC"){
72              alert("5번 오류로 비밀번호 찾기를 진행합니다.");
73              locationPath = "findpw.php";
74              document.location.href = locationPath;
75          }else if(responseText.result == "N5"){
76              alert("5번 오류로 비밀번호 찾기를 진행합니다.");
77              locationPath = "findpw.php";
78              document.location.href = locationPath;
79          }else{
80              alert("이메일 주소 또는 비밀번호를 확인해주세요.");
81              return false;
82          }
83      }
```

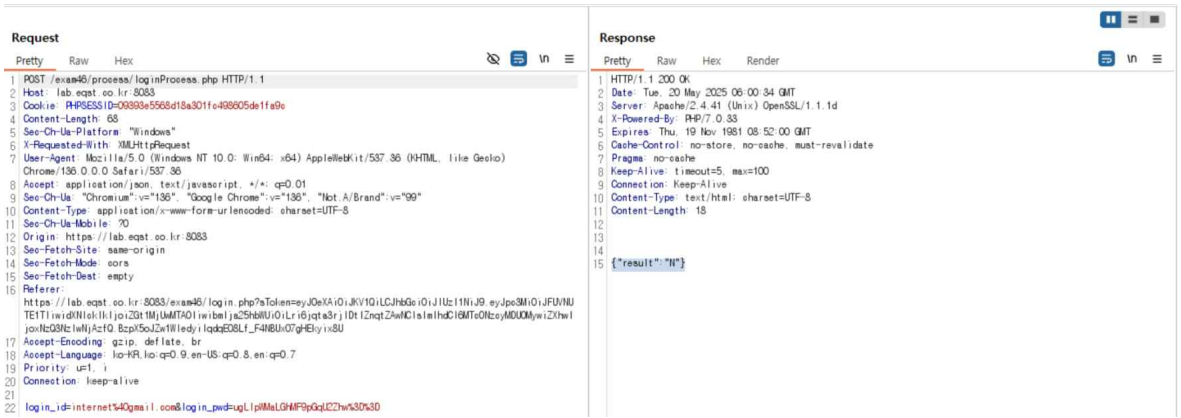
Step 3. 파라미터 변조

3-1. Intercept

: 로그인 시도 과정을 BurpSuite로 Intercept하면 Request만 존재하는 것을 확인할 수 있음. 이 상태에서 Forward 버튼을 눌러 요청을 전송하면 Response가 반환되며, 이를 확인해보면 {"result", "N"}이라는 부분을 확인할 수 있음. 이 값은 결과가 N임을 의미하며, 이는 Step 2에서 분석한 조건문에 따라 Y, YP, N1, N2, N3, N4, NC 중 어느 것에도 해당하지 않는 경우로 이메일 주소 또는 비밀번호를 확인해달라는 메시지를 출력하게 됨.



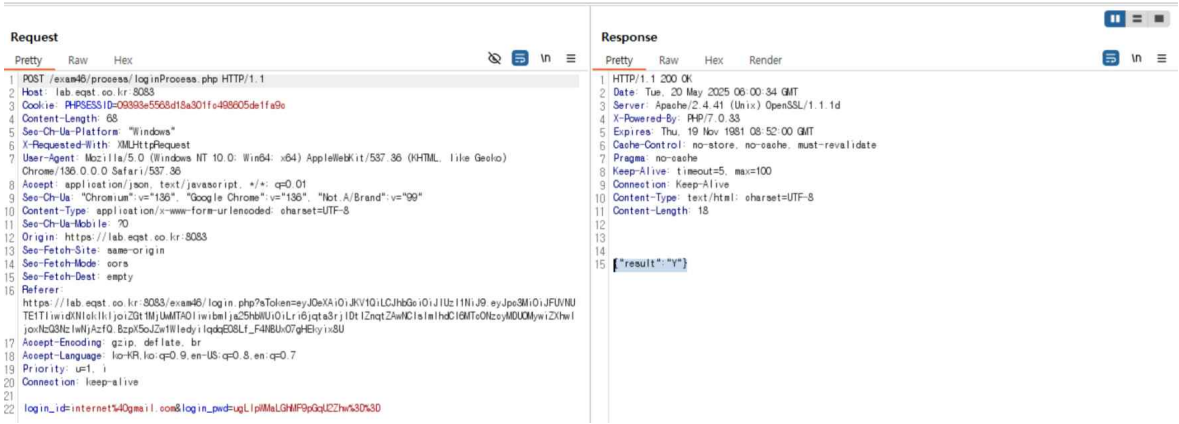
- Forward 결과 (Response 확인 가능)



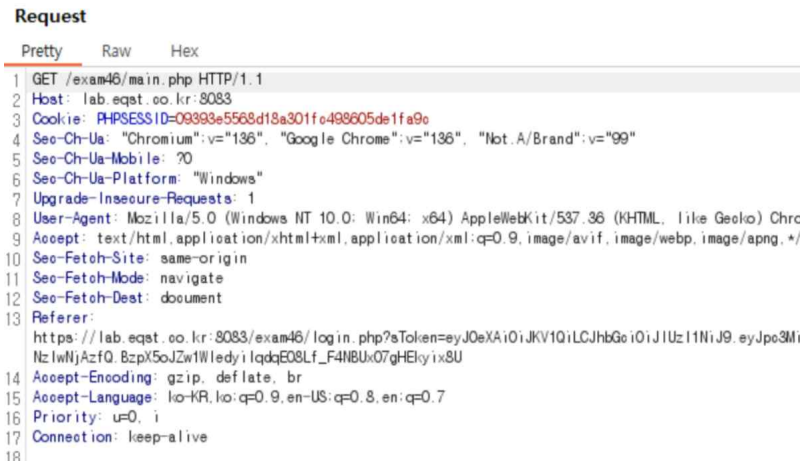
(다음 페이지에 이어서)

3-2. 응답 수정

: 정답을 획득하기 위해선 main.php로 이동해야 하므로 현재 응답인 "N"을 "Y"로 수정하여 main.php로 연결되도록 조작한 뒤 Forward함. 그 결과, Request에서 main.php로 정상 연결되는 것을 확인할 수 있음.

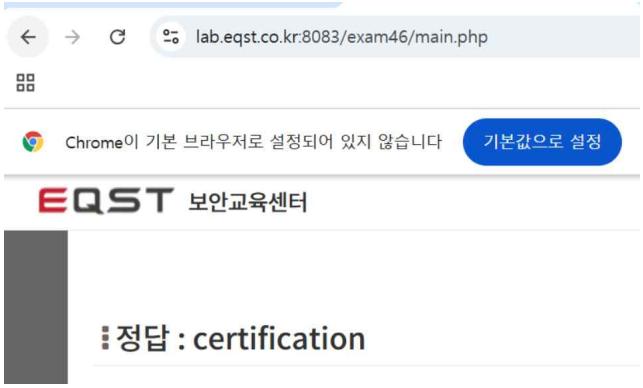


- Forward 결과 (main.php로 이동)



Step 4. 정답 확인

: Step 3-2에서 Forward한 결과를 확인 후 Intercept를 off 한 후 브라우저를 통해 확인해보면 main.php로 연결되며 정답(certification)을 확인할 수 있음.



(다음 페이지에 이어서)

성명	프로젝트 후 소감
김가람	이번 실습을 통해 단순히 세션ID 형식을 유추하여 조작하거나 응답 값을 바꾸는 것만으로도 인증을 우회할 수 있다는 사실에서 생각보다 웹 보안이 굉장히 허술할 수 있다는 것을 느꼈고 복잡하거나 어려운 기술만이 아닌 이런 단순한 취약점을 노리는 공격이 더 현실적이고 위험할 수 있겠다는 생각이 들었음. 또 파라미터 변조 실습에서는 클라이언트 측에서 서버의 응답을 조건문으로 처리하고 그 결과에 따라 페이지에 연결되도록 하는 구조가 취약점이 될 수 있음을 알게 되었고, 검증은 반드시 클라이언트가 아닌 서버에서 처리되어야 한다는 원칙의 중요성을 직접 느낄 수 있었음. 이처럼 단순히 기능 구현만을 고려한 개발은 어떤 보안적 문제가 발생할 수 있는지를 놓치기 쉽겠다는 것도 느낌. 따라서 단지 기능이 잘 동작하는 것에 만족하는 것이 아니라 이 기능이 안전하게 동작하는가에 대해서도 함께 고민하는 보안 전문가가 되어야겠다는 생각을 함.