

□ 단국대학교 > 사이버보안 학과 > 인터넷 보안 > 보고서 템플릿

25년 1학기 인터넷 보안 수업

인터넷 뱅킹 모의해킹

: 10억 대출 받기

2025년 5월 27일

학번 : 32231594

이름 : 박기쁨

본 실습은 연봉의 200%, 최대 1 억까지 대출해주는 금융 상품 페이지를 해킹하여 10 억을 대출받는 과정을 다룬다.

Step 1-1. 대출 과정 확인

: 상품 설명에서 연봉 5 천만원 이상은 5 천만원으로 계산된다는 내용을 확인할 수 있다.

: 신청 버튼을 누르면, 연봉과 계좌번호를 입력하는 가입 정보 입력 페이지로 넘어간다.
: 연봉을 999999999999999999 원으로 입력해본다.

가입 정보 입력

신청 상품	잘 찾아서 10억 벌자
신청 기간	1 년
연봉 정보	<div>99999999999999999999 원</div>
대출 금액	연봉의 200.0% (최대 100,000,000원)
입금 계좌번호	<div>21354 - 001 - 0479001</div>

다음으로

: 정보 입력 후 '다음으로' 버튼을 누르면, 계좌번호와 비밀번호, 보안카드 번호를 입력하는 사용자 인증 페이지로 넘어간다.

사용자 인증

계좌 인증

계좌번호

21354 - 001 - 0479001

비밀번호

....

계좌 비밀번호 인증

보안카드 인증

32번 앞 두자리

..

29번 뒤 두자리

..

보안카드 인증

공동인증서(구 공인인증서)
인증

: 정보 입력 후 공동인증서 인증을 마치면, 마지막 페이지로 넘어가며 대출이 완료된다.

: 거래 내역 조회 페이지에서 대출이 성공적으로 이루어졌음을 알 수 있다.

: 연봉을 999999999999999999 원으로 입력하였지만 5천만원으로 계산되어, 연봉의 200%인 1 억이 대출되었음을 알 수 있다.

거래 내역 조회

계좌번호

21354 - 001 - 0479001

시작 날짜

종료 날짜

거래 유형

검색

2025-05-27

2025-05-27

전체

거래일시	보낸분/받은분	금액	잔액	거래 유형
2025-05-27	대출금 입금	100,000,000원	2,701,000,000원	입금

Step 1-2. 패킷 확인

: 가입 정보 입력 페이지에서 연봉 정보가 패킷에서 어떻게 전달되는지 확인한다.

가입 정보 입력

신청 상품	잘 찾아서 10억 벌자
신청 기간	1 년
연봉 정보	99999999999999999999 원
대출 금액	연봉의 200.0% (최대 100,000,000원)
입금 계좌번호	21354 - 001 - 0479001

다음으로

: Burp 를 이용하여 패킷을 intercept 해보면, 파라미터와 파라미터 값들이 암호화되어 전달되고 있음을 알 수 있다.

The screenshot shows the Burp Suite interface with the Intercept tab selected. A request is intercepted from https://elms2.skinfosec.co.kr:8111/practice/practice13/applyproduct/loan/checkinfo. The request is a POST method. The Inspector panel on the right shows the request details, including the request body which is encrypted (e2e_data).

Request details:

- Time: 16:15:33 27 May
- Type: HTTP
- Direction: Request
- Method: POST
- URL: https://elms2.skinfosec.co.kr:8111/practice/practice13/applyproduct/loan/checkinfo
- Status code:
- Length:

Request body (e2e_data):

```
U2FsdGVkX1ZB/B8qY5hVkhU3R0RPFjkjD1H7Cob1r49B/1X2Fcd7:MM1Y4qjR6c3BqKFEVL8Iaz3TtpheBqRhe18I64DLnesdFSP1IhsaQmQ7aZ1iFDK7B4EDYH7Dau7e762SD1ED0p58YrEZ308Mk328b1z4PdlRyycDkgJ2hck34aksh8H7C  
RbNix:7MF1HqprQnYQLxKlPGDTyJL8KfCCLoAMjSHq2BhMAn7Aa6A53D
```

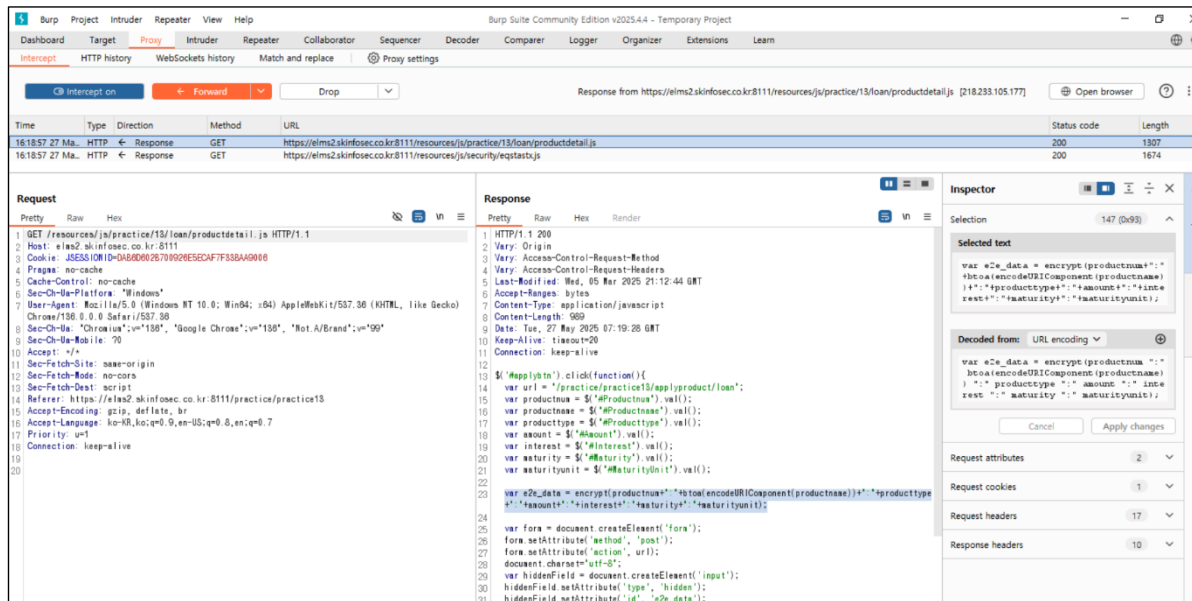
Step 2. prompt 를 이용한 파라미터 변조

Step 2-1. prompt 문 삽입

: 패킷의 파라미터 값들이 암호화되어 전달되고 있으므로, 패킷을 intercept 하여 실시간으로 변조하는 것은 불가능하다.

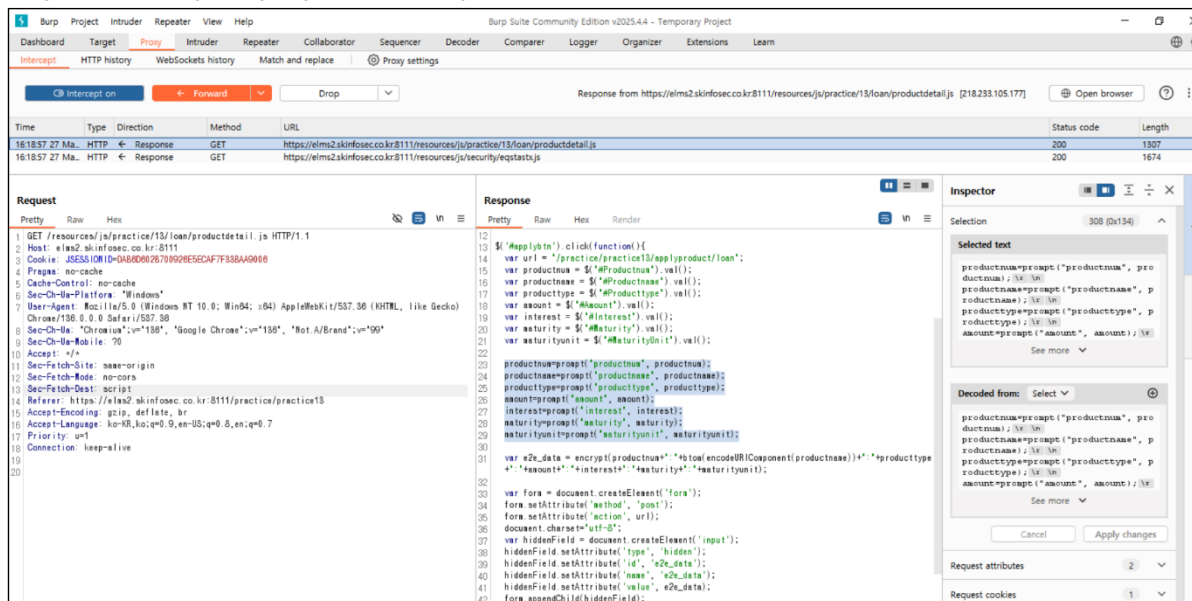
: 페이지의 소스코드에 prompt 문을 삽입하고, 띄워진 prompt 를 이용하여 파라미터 값을 변조한다.

: 페이지 최초 로드 시 전달되는 `productdetail.js` 파일을 intercept 하여 소스 코드를 확인해보면, E2E 암호화가 실행되는 부분을 확인할 수 있다.

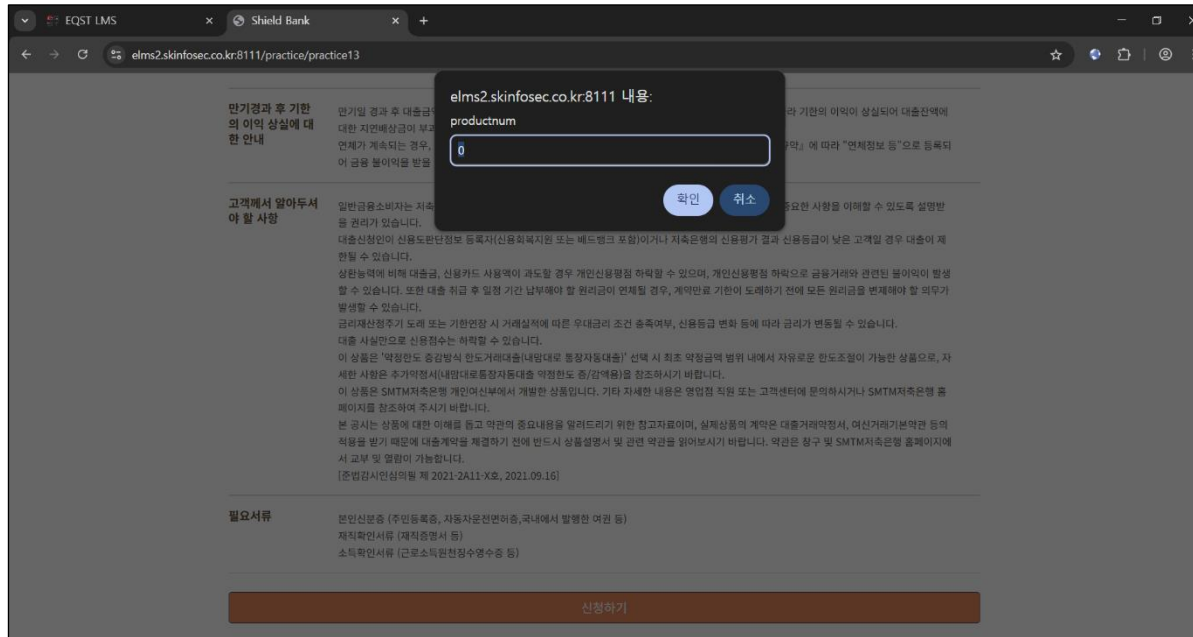


: E2E 암호화 코드 바로 앞에, 모든 파라미터에 대한 prompt 문을 작성하여 삽입한다.

({parameter}=prompt("parameter", {parameter}))



: intercept 를 해제한 후 페이지로 돌아가 신청하기 버튼을 누르면 prompt 창이 띄워지며, 파라미터와 파라미터 값에 접근할 수 있게 된다.



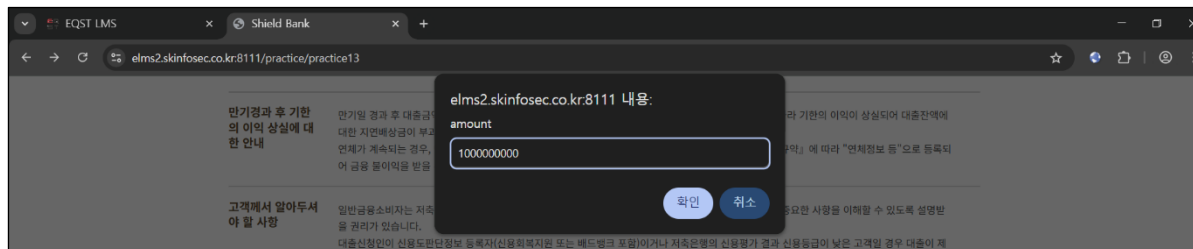
(모든 파라미터와 파라미터 값이 prompt 로 띄워진다.)



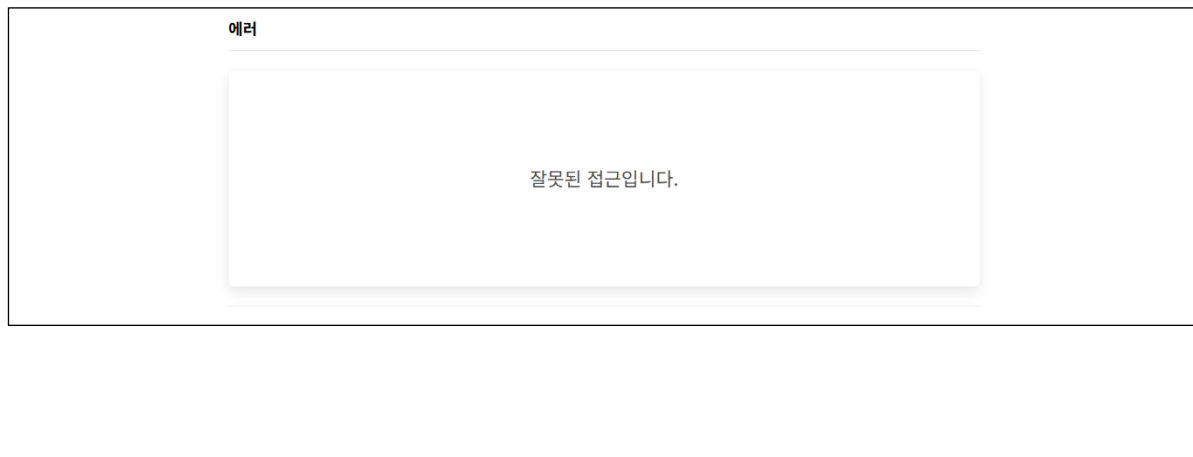


Step 2-2. prompt 를 이용한 파라미터 변조 시도 (amount)

: 최대 대출 금액 파라미터로 추정되는 amount 값을 1 억에서 10 억으로 변조해본다.



: 잘못된 접근이라는 문구가 있는 에러 페이지로 이어진다.



Step 2-3. prompt 를 이용한 파라미터 변조 시도 (interest)

: amount 대신, 최대 대출 금액의 연봉 대비 비율 파라미터로 추정되는 interest 값을 200.0 에서 2000.0 으로 변조해본다.



: 가입 정보 입력 페이지로 무사히 이어진다. (타 파라미터들의 값을 변조하면, amount 파라미터 변조 시와 마찬가지로 예러 페이지로 이어진다.)

: 최대 대출 금액의 연봉 대비 비율이 200.0%에서 2000.0%로 변조되었음을 확인할 수 있다.

가입 정보 입력

신청 상품	잘 찾아서 10억 별자
신청 기간	1 년
연봉 정보	<input type="text"/> 원
대출 금액	연봉의 2000.0% (최대 100,000,000원)
입금 계좌번호	계좌를 선택해 주세요. ▼

다음으로

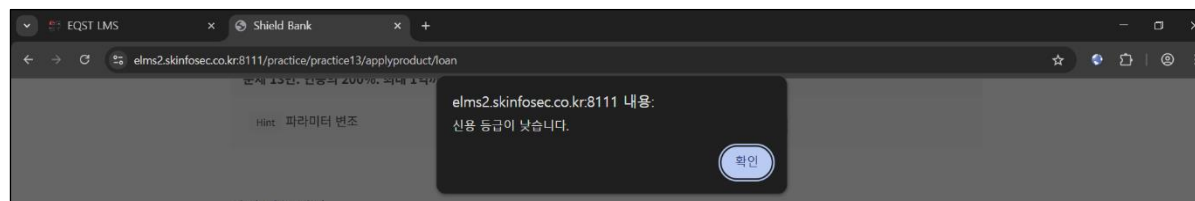
: 기타 정보들을 입력한 후 '다음으로' 버튼을 누른다.

가입 정보 입력

신청 상품	잘 찾아서 10억 별자
신청 기간	1 년
연봉 정보	50000000 원
대출 금액	연봉의 2000.0% (최대 100,000,000원)
입금 계좌번호	21354 - 001 - 0479001 ▼

다음으로

: "신용 등급이 낮습니다."라는 문구의 알림창이 띄워진다.

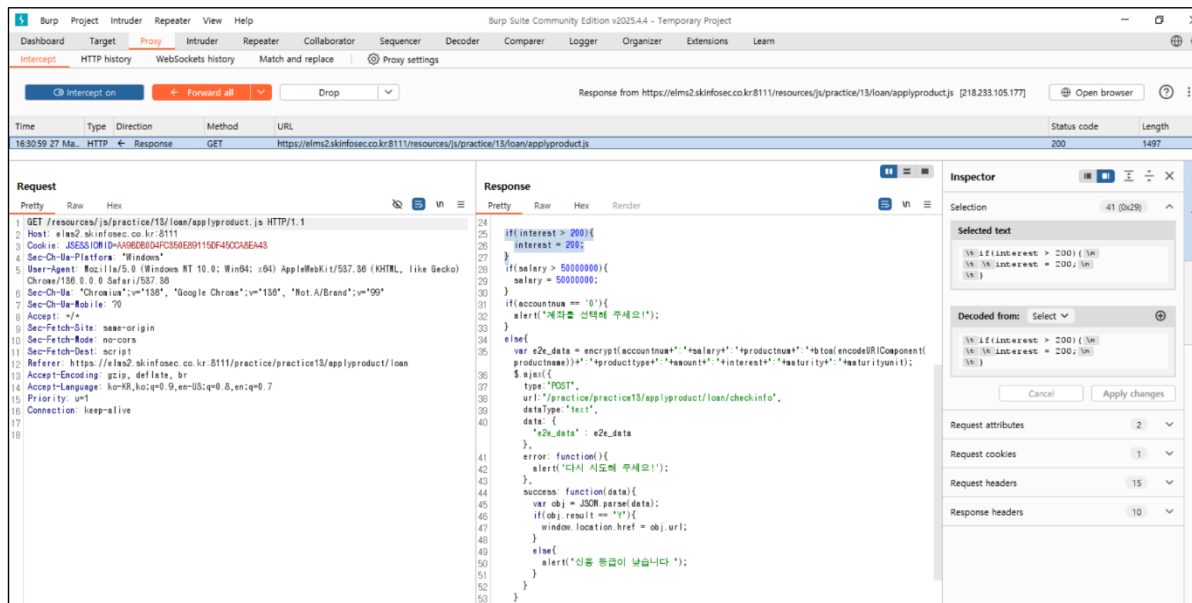


Step 3. 두 번째 페이지 소스코드 변조

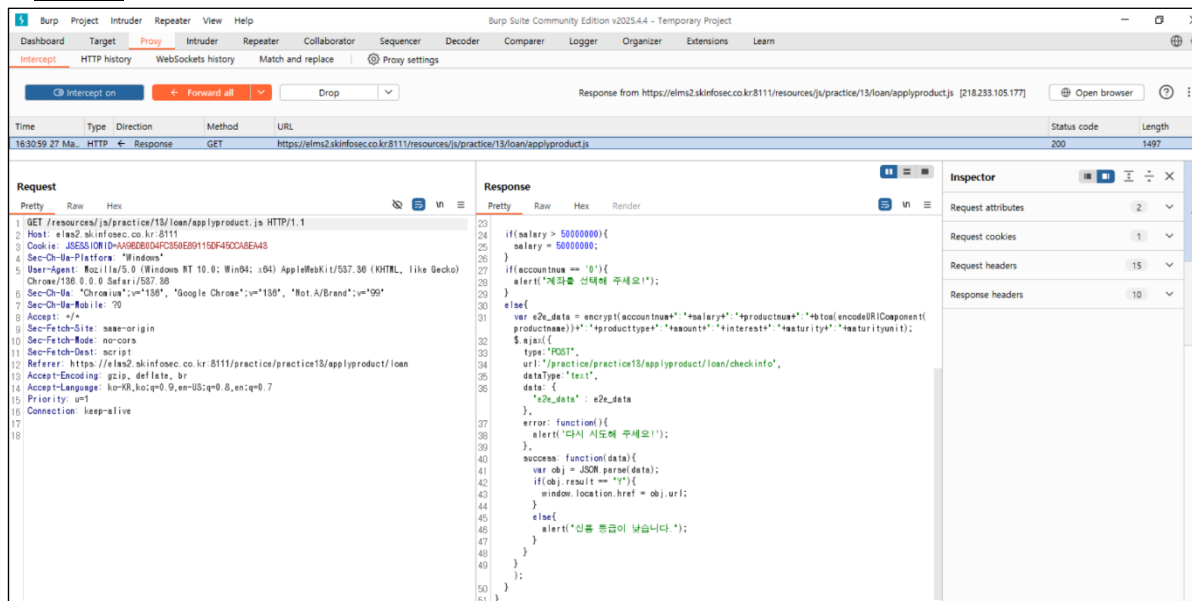
Step 3-1. 두 번째 페이지 소스코드 확인

: applyproduct.js 파일을 intercept 하여 소스 코드를 확인해보면, 모종의 이유로 obj.result 값이 y가 아닌 다른 값으로 설정되어 "신용 등급이 낮습니다" 문구가 띄워졌음을 알 수 있다.

: 이전 페이지에서, prompt 를 이용하여 유일하게 interest 파라미터 값만 변조하였으므로, interest 파라미터 관련 코드에서 오류가 발생했을 가능성이 가장 높다.



: interest 값을 200 이하로 제한하는 코드를 삭제해본다.



: 코드 변조 후 intercept 를 해제하고, 기타 정보들을 입력한 후 '다음으로' 버튼을 눌러본다.

가입 정보 입력

신청 상품	잘 찾아서 10억 별자
신청 기간	1 년
연봉 정보	50000000 원
대출 금액	연봉의 2000.0% (최대 100,000,000원)
입금 계좌번호	21354 - 001 - 0479001

다음으로

: 사용자 인증 페이지로 무사히 이어진다.

Shield Bank

이류스트님 로그인 인증센터

개인 기업 금융상품 실습문제

문제 13번. 연봉의 200%. 최대 1억까지 대출해주는 상품에서 10억을 대출 받으세요.

Hint 파라미터 변조

사용자 인증

계좌 인증

계좌번호 21354 - 001 - 0479001 비밀번호 계좌 비밀번호 4자리

Step 3-2. 정보 입력 및 10 억 대출

: 계좌번호와 비밀번호, 보안카드 번호를 입력한 후 공동인증서 인증을 마친다.

사용자 인증

계좌 인증

계좌번호 21354 - 001 - 0479001 비밀번호 ****

계좌 비밀번호 인증

보안카드 인증

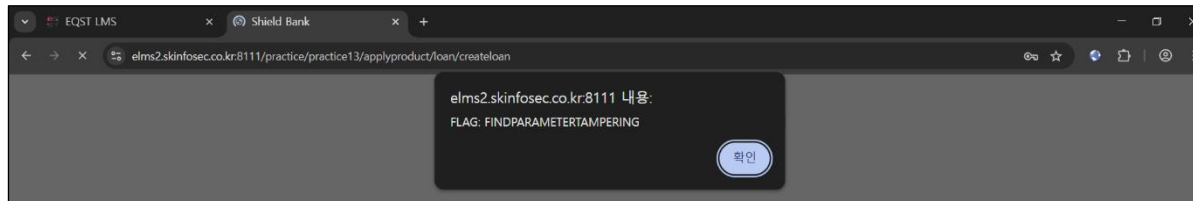
14번 앞 두자리 20번 뒤 두자리

.. ..

보안카드 인증

공동인증서(국공인인증서)

: 10 억 대출에 성공하여 FLAG 를 획득할 수 있다. (FLAG: FINDPARAMETERTAMPERING)



: 거래 내역 조회 페이지에서도 10 억 대출 내역을 확인할 수 있다.

거래 내역 조회

계좌번호
21354 - 001 - 0479001

시작 날짜: 2025-05-27 종료 날짜: 2025-05-27 거래 유형: 전체 검색

거래일시	보낸분/받은분	금액	잔액	거래 유형
2025-05-27	대출금 입금	1,000,000,000원	3,701,000,000원	입금
2025-05-27	대출금 입금	100,000,000원	2,701,000,000원	입금

Step X. 기타 방법 (개발자도구 이용)

금융권 실무에서는 개발자도구 사용이 막혀 있는 경우가 많아 본 실습에서도 개발자도구를 사용하지 않았지만, 개발자도구를 사용하여 보다 간편하게 실습을 진행할 수 있다.

: 개발자도구를 통해 productdetail.js 파일의 소스코드를 확인하고 prompt 문을 삽입한다.

The image displays two screenshots of a web browser (Chrome) with the Chrome DevTools interface open. The browser window shows a web page titled "Shield Bank" with a URL of "elms2skinfosec.co.kr:8111/practice/practice13". The page content includes a "필요서류" (Required Documents) section and a "신청하기" (Apply) button.

The DevTools interface is open to the "Sources" panel, showing the file "productdetail.js". The code in the file is as follows:

```
1  $( "#apolybta" ).click(function(){
2      var url = "/practice/practice3/apolyproduct/loan";
3      var productnum = $("#productnum").val();
4      var productnae = $("#productnae").val();
5      var producttype = $("#producttype").val();
6      var amount = $("#amount").val();
7      var interest = $("#interest").val();
8      var maturity = $("#maturity").val();
9      var maturityunit = $("#maturityunit").val();
10
11      var e2e_data = encrypt(productnum+"*"+btoa(encodeURIComponent(
12
13      var fora = document.createElement( 'form' );
14      fora.setAttribute( 'method', 'post' );
15      fora.setAttribute( 'action', url );
16      document.charset="utf-8";
17      var hiddenField = document.createElement( 'input' );
18      hiddenField.setAttribute( 'type', 'hidden' );
19      hiddenField.setAttribute( 'id', 'e2e_data' );
20      hiddenField.setAttribute( 'name', 'e2e_data' );
21      hiddenField.setAttribute( 'value', e2e_data );
22      fora.appendChild( hiddenField );
23      document.body.appendChild( fora );
24      fora.submit();
25  });
```

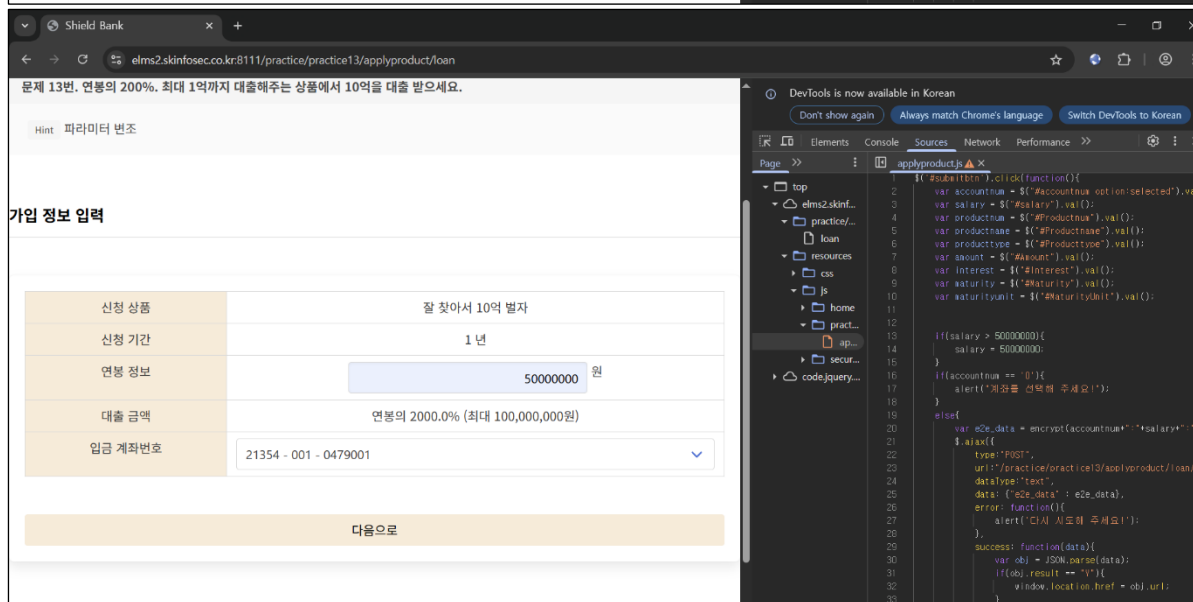
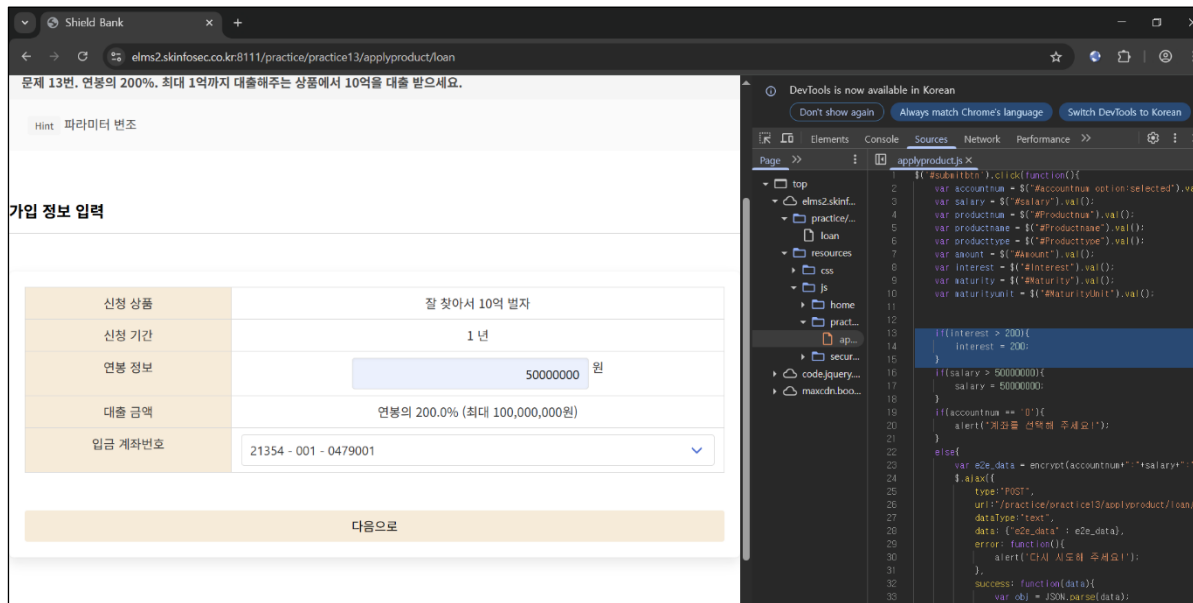
The second screenshot shows the same file with the following JavaScript code injected into the "productnum" variable:

```
productnum=prompt("productnum", productnum);
productnae=prompt("productnae", productnae);
producttype=prompt("producttype", producttype);
amount=prompt("amount", amount);
interest=prompt("interest", interest);
maturity=prompt("maturity", maturity);
maturityunit=prompt("maturityunit", maturityunit);
```

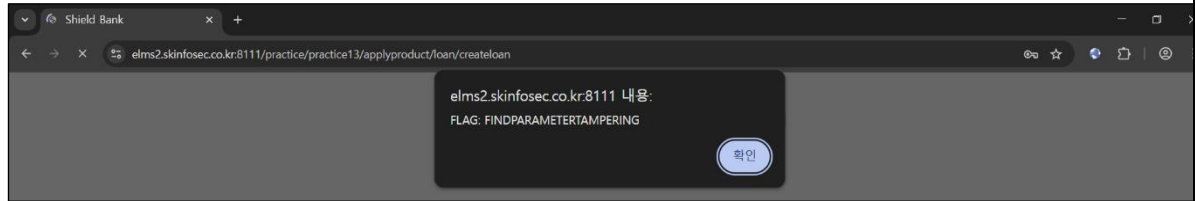
: interest 파라미터의 값을 2000.0으로 변조한다.



: applyprodunt.js 파일의 소스코드 중 interest 값을 제한하는 코드를 삭제한다.



: 마찬가지로 10 억 대출에 성공하여 플래그를 획득한다.



성명	프로젝트 후 소감
박기쁨	지금까지의 실습은 그날 배운 내용을 적용해볼 수 있는 단순한 페이지에서의 실습이었다면, 이번 실습은 실제 금융 사이트와 흡사한 사이트에서 진행되어 더욱 재밌고 특별한 실습이었던 것 같다. 파라미터를 변조하는 방법 중 프롬프트를 이용하는 방법을 처음 배웠는데, 패킷을 intercept 하여 실시간으로 변조하는 방법을 사용할 수 없을 때, 소스코드를 변조하는 방법도 시도해볼 수 있다는 사실을 알게 되었다.