

□ 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

<과정 설명>

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

Step 1. 페이지 동작 방식 확인

Step 1-1. 대출 과정 확인

- : 상품 페이지를 따라가며 대출 과정을 확인한다.
- : 상품 설명에서 연봉 5 천만원 이상은 5 천만원으로 계산된다는 내용을 확인할 수 있다.

The screenshot shows a web browser window for 'Shield Bank' with the URL 'elms2.skinfosec.co.kr:8111/practice/practice13'. The page displays a product detail for '문제 13번. 연봉의 200% 최대 1억까지 대출해주는 상품에서 10억을 대출 받으세요.' (Problem 13. Borrow 10 billion from a loan product where the annual income is 200% and the maximum is 1 billion). The product description states: '연봉의 200%(최대 1억원)을 대출 받을 수 있는 상품입니다.' (A product that can borrow up to 100 million won based on 200% of the annual income (maximum 1 billion)). Below this, there is a note: '연봉 5천만원 이상은 5천만원으로 계산됩니다.' (Annual income over 50 million won is calculated as 50 million won). The page also includes sections for application requirements, loan amount, and repayment plan.

- : 신청 버튼을 누르면, 연봉과 계좌번호를 입력하는 가입 정보 입력 페이지로 넘어간다.
- : 연봉을 99999999999999999999 원으로 입력해본다.

The screenshot shows the '가입 정보 입력' (Application Information Input) page. The form contains the following data:

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

At the bottom of the form is a button labeled '다음으로' (Next).

: 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.

완료

대출이 완료되었습니다!

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

: 연봉을 99999999999999999999 원으로 입력하였지만 5천만원으로 계산되어, 연봉의 200%인 1억이 대출되었음을 알 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/받는분	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	1년
연봉 정보	9999999999999999 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.

Burp Suite Community Edition v2025.4.4 - Temporary Project

Request

Pretty Raw Hex

1 POST /practice/practice13/applyproduct/loan/checkinfo HTTP/1.1
2 Host: elms2.skinfosec.co.kr:8111
3 Cookie: jwttoken=JWTTOKENLCJhbGciOiJIUzI1NiJ9eyJzdWIiOiJkYjYtMjE2MDA5OTc4ZDctZC1lLCJleHAiOEM0dgIkczN0psInVzZXJpZC18IvVic3QnIiI0In0.rHtDe7PhMyo6fHQ8tnIbuByZ-BedX8_CodPQJWnU;
4 SECSESSIONID=3C7CEB9A7C4B309C5aF0C7241
5 Sec-Ch-Ua-Platform: "Windows"
6 X-Requested-With: XMLHttpRequest
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.0.0 Safari/537.36
8 Accept: text/html, */*; q=0.01
9 Sec-Ch-Ua: "Chromium";v="138", "Google Chrome";v="138", "Not/A/Brand";v="99"
10 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
11 Sec-Ch-Ua-Mobile: ?0
12 Origin: https://elms2.skinfosec.co.kr:8111
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Dest: empty
16 Referer: https://elms2.skinfosec.co.kr:8111/practice/practice13/applyproduct/loan
17 Content-Encoding: gzip; deflate, br
18 Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
19 Priority: uvt, i
20 Connection: keep-alive
21
22 e2e-data=
UFsd0lXN12BqY5hWohJ3Kd#GPkjD1#TzCob1#40B#%2Fc07#1#1Y4q#%z#2gKFEVL8#azSTtqpnBhg#res18#40Lnesk#F9Pt#bsaQmQYqzI#FDKT0#ED#ITnu#7%2BD1#E#lgy5SYrEZ808W#28b#tz4Pd#syeOkqJQck3Asks#8fTC#h#h#-TMFIHjsprQwYGLX#PSD7yjL8xFtCoI#MJS#2B#W#b#7#h#45#D

Inspector

Request attributes: 2
Request query parameters: 0
Request body parameters: 1
Request cookies: 2
Request headers: 19

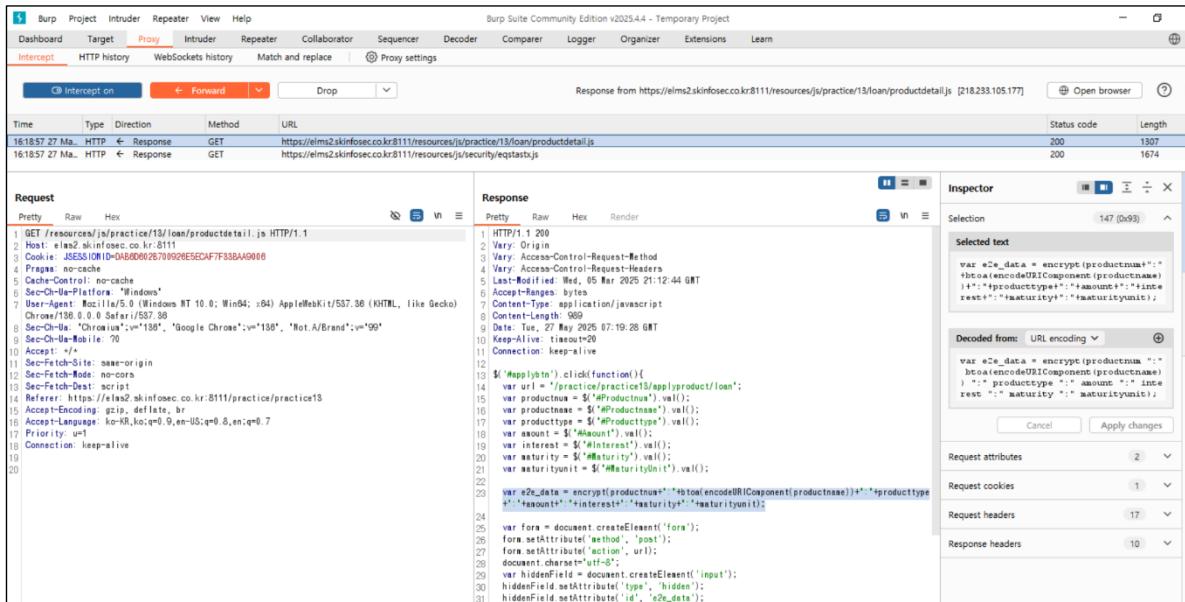
Step 2. prompt 를 이용한 파라미터 변조

Step 2-1. prompt 문 삽입

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

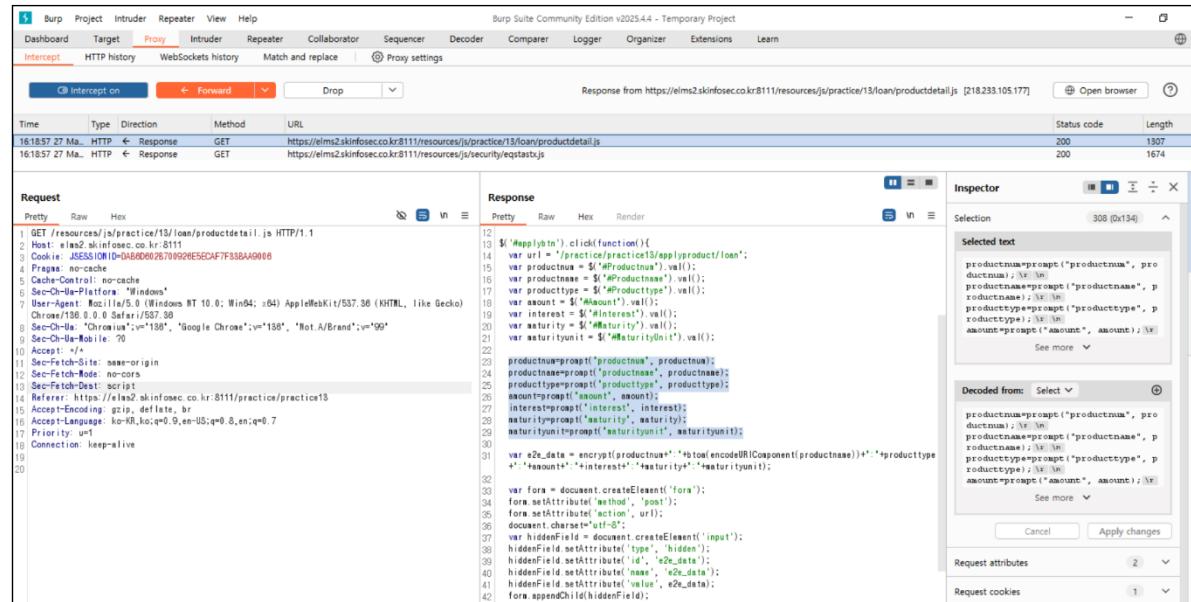
: 페이지의 소스코드에 prompt 문을 삽입하고, 띄워진 prompt 를 이용하여 파라미터 값을
변조한다.

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

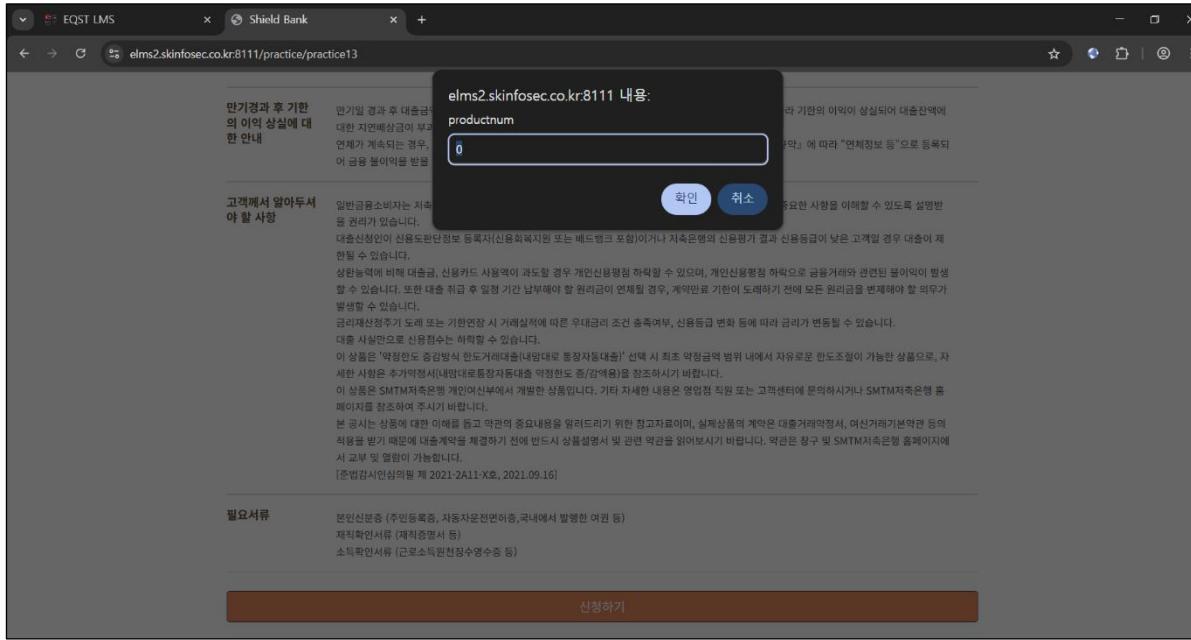


: E2E 암호화 코드 바로 앞에, 모든 파라미터에 대한 prompt 문을 작성하여 삽입한다.

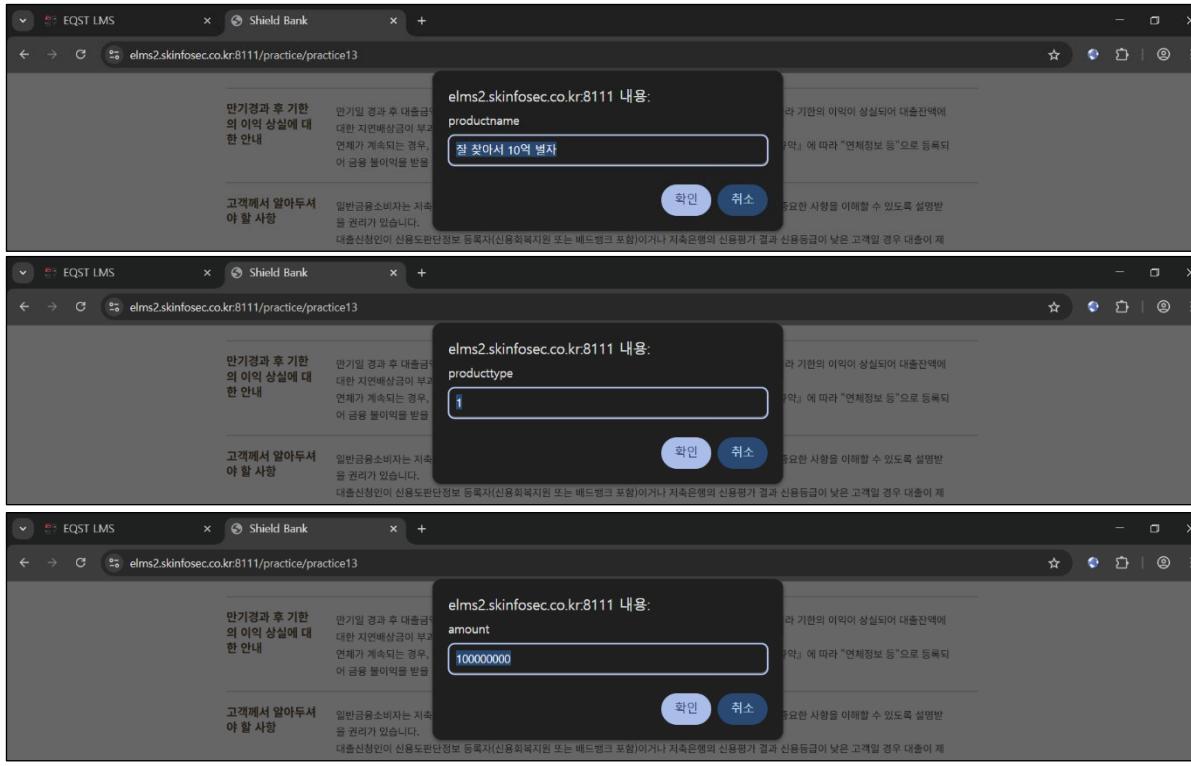
```
( {parameter}=prompt("parameter", {parameter}) )
```

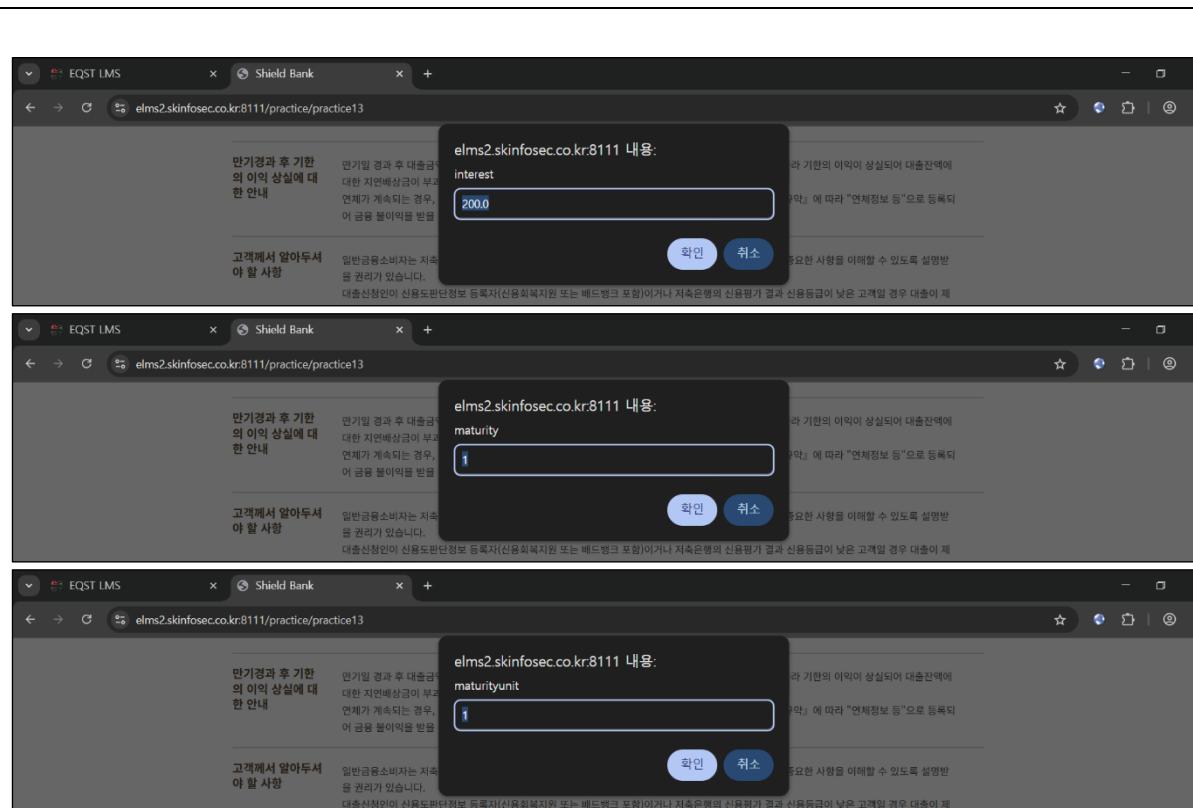


: intercept 를 해제한 후 페이지로 돌아가 신청하기 버튼을 누르면 prompt 창이 띄워지며, 파라미터와 파라미터 값에 접근할 수 있게 된다.



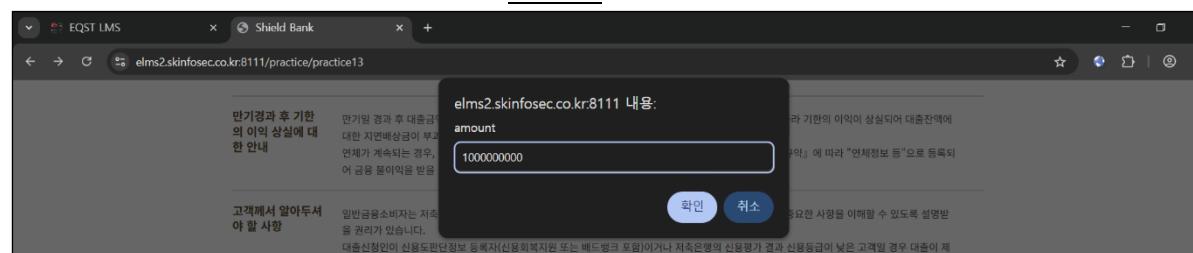
(모든 파라미터와 파라미터 값이 prompt 로 띄워진다.)





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 200.0에서 2000.0으로 변조해본다.

The screenshot shows a browser window with a modal dialog. The modal title is "elms2.skininfosec.co.kr:8111 내용:" and it contains the parameter "interest" with the value "2000.0". There are "확인" (Confirm) and "취소" (Cancel) buttons at the bottom of the modal. The background page has Korean text about loan application conditions and a note about interest rates.

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

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

The screenshot shows a form titled "가입 정보 입력". The form fields are: 신청 상품 (Loan Product), 신청 기간 (Application Period), 연봉 정보 (Salary Information), 대출 금액 (Loan Amount), and 입금 계좌번호 (Bank Account Number). The "연봉 정보" field is highlighted with a yellow background. Below the form is a large orange button labeled "다음으로" (Next Step).

: 기타 정보들을 입력한 후 '다음으로' 버튼을 누른다.

The screenshot shows the same form as the previous step, but with the "연봉 정보" field filled with "50000000 원". Below the form is a large orange button labeled "다음으로" (Next Step).

: "신용 등급이 낮습니다."라는 문구의 알림창이 띄워진다.

The screenshot shows a modal dialog with the message "Hint: 파라미터 변조" and "신용 등급이 낮습니다.". A blue "확인" (Confirm) button is visible at the bottom right of the modal.

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.

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. A GET request to <https://elms2.skinfosec.co.kr:8111/resources/js/practice/13/loan/applyproduct.js> is captured. The response body is as follows:

```
if(interest > 200){  
    interest = 200;  
}  
  
if(salary > 5000000){  
    salary = 5000000;  
}  
  
if(accountnum == ''){  
    alert('계좌를 선택해 주세요!');  
}  
  
else{  
    var e2e_data = encrypt(accountnum+'.'+salary+'.'+productnum+'.'+btos(encodeURI(Component.productname))+'.'+producttype+'.'+amount+'.'+interest+'.'+maturityType+'.'+maturityUnit);  
    $.ajax({  
        type: "POST",  
        url: "/practice/practice13/applyproduct/loan/checkinfo",  
        dataType: "text",  
        data: {  
            "e2e_data": e2e_data  
        },  
        error: function(){  
            alert('다시 시도해 주세요!');  
        },  
        success: function(data){  
            var obj = JSON.parse(data);  
            if(obj.result == ''){  
                window.location.href = obj.url;  
            }  
            else{  
                alert("신청 등록이 납득니다.");  
            }  
        }  
    })  
}
```

: interest 값을 200 이하로 제한하는 코드를 삭제해본다.

The screenshot shows the Burp Suite interface with the following details:

- Request:** A GET request to `https://elms2.skinfosec.co.kr:8111/resources/js/practice/13/loan/applyproduct.js`.
- Response:** The response body contains a JavaScript file with code related to loan application processing.
- Inspector:** The right panel displays the response headers, which include:
 - Request attributes: 2
 - Request cookies: 1
 - Request headers: 15
 - Response headers: 10

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

The screenshot shows a loan application form with the following details:

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

다음으로 (Next) button at the bottom.

: 사용자 인증 페이지로 무사히 이어진다.

The screenshot shows a user authentication page for Shield Bank. The top navigation bar includes links for 보안카드, 금융서비스, 고객센터, 은행소개, 카드, 증권, and 실드멤버스. The main content area displays a message: "문제 13번. 연봉의 200%, 최대 1억까지 대출해주는 상품에서 10억을 대출 받으세요." Below this is a hint: "Hint 파라미터 변조". The "사용자 인증" section contains fields for 계좌 인증 (Account Authentication) and 보안카드 인증 (Card Authentication). The account authentication section has fields for 계좌번호 (Account Number) and 비밀번호 (Password). The card authentication section has fields for 14번 앞 두자리 (First two digits of 14), 20번 뒤 두자리 (Last two digits of 20), and a placeholder for the card number itself.

Step 3-2. 정보 입력 및 10 억 대출

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

The screenshot shows the final step of the user authentication process. It consists of two main sections: "계좌 인증" (Account Authentication) and "보안카드 인증" (Card Authentication).

계좌 인증 section:

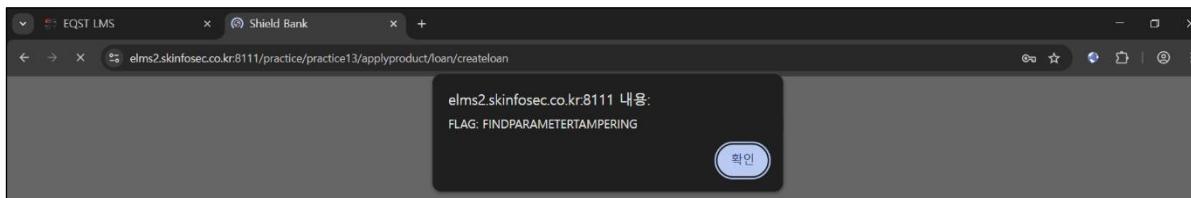
- 계좌번호: 21354 - 001 - 0479001
- 비밀번호: (Placeholder)
- 계좌 비밀번호 인증** button

보안카드 인증 section:

- 14번 앞 두자리: ..
- 20번 뒤 두자리: ..
- 카드 번호: .. (Placeholder)
- 보안카드 인증** button

At the bottom center is a large orange button labeled "고동인증서/고인인증서".

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



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

A screenshot of a transaction history search page. The title is '거래 내역 조회'. There are search filters for '계좌번호' (Account Number) set to '21354 - 001 - 0479001', '시작 날짜' (Start Date) set to '2025-05-27', '종료 날짜' (End Date) set to '2025-05-27', and '거래 유형' (Transaction Type) set to '전체' (All). Below the filters is a table showing transaction details. The table has columns: 거래일시 (Transaction Date), 보낸분/받는분 (Recipient/Beneficiary), 금액 (Amount), 잔액 (Balance), and 거래 유형 (Transaction Type). Two rows are shown: one for May 27, 2025, where the amount is 1,000,000,000원 and balance is 3,701,000,000원, and another for May 27, 2025, where the amount is 100,000,000원 and balance is 2,701,000,000원. Both entries are categorized as '입금' (Deposit).

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.

```

1  $('#applybtn').click(function(){
2    var url = '/practice/practice3/applyproduct/loan';
3    var productnum = $('##Productnum').val();
4    var productname = $('##ProductName').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   var e2e_data = encrypt(productnum+":"+btos(encodedURL));
11   var form = document.createElement('form');
12   form.setAttribute('method', 'post');
13   form.setAttribute('action', url);
14   document.charset='utf-8';
15   var hiddenField = document.createElement('input');
16   hiddenField.setAttribute('type', 'hidden');
17   hiddenField.setAttribute('id', 'e2e_data');
18   hiddenField.setAttribute('name', 'e2e_data');
19   hiddenField.setAttribute('value', e2e_data);
20   form.appendChild(hiddenField);
21   document.body.appendChild(form);
22   form.submit();
23 });
24 });
25 });

```



```

1  $('#applybtn').click(function(){
2    var url = '/practice/practice3/applyproduct/loan';
3    var productnum = $('##Productnum').val();
4    var productname = $('##ProductName').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   var e2e_data = encrypt(productnum+":"+btos(encodedURL));
11   var form = document.createElement('form');
12   form.setAttribute('method', 'post');
13   form.setAttribute('action', url);
14   document.charset='utf-8';
15   var hiddenField = document.createElement('input');
16   hiddenField.setAttribute('type', 'hidden');
17   hiddenField.setAttribute('id', 'e2e_data');
18   hiddenField.setAttribute('name', 'e2e_data');
19   hiddenField.setAttribute('value', e2e_data);
20   amount=encrypt('amount', amount);
21   interest=encrypt('interest', interest);
22   maturity=encrypt('maturity', maturity);
23   maturityunit=encrypt('maturityunit', maturityunit);
24   var e2e_data = encrypt(productnum+":"+btos(encodedURL));
25 });
26 });
27 });
28 });
29 });
30 });
31 });
32 });
33 });

```

: interest 파라미터의 값을 2000.0으로 변조한다.

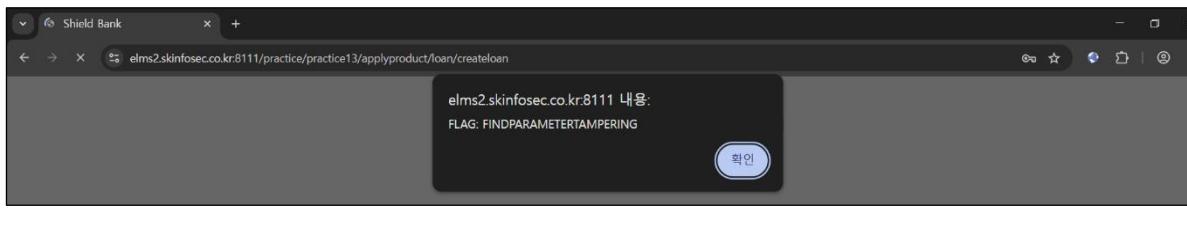
The screenshot shows a web browser window with a form titled "elms2.skinfosec.co.kr:8111/practice/practice13". The form has an input field for "interest" set to "2000.0". The developer tools are open, showing the "Sources" tab with the file "productdetails.js". A specific line of code is highlighted: `var interest = \$('#interest').val();`

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

The screenshot shows a web browser window with a form titled "elms2.skinfosec.co.kr:8111/practice/practice13/applyproduct/loan". The form has an input field for "interest" set to "2000.0". The developer tools are open, showing the "Sources" tab with the file "applyproduct.js". The line of code `if(interest > 200){ interest = 200; }` is commented out with a double slash `//` at the start of the line.

The screenshot shows a web browser window with a form titled "elms2.skinfosec.co.kr:8111/practice/practice13/applyproduct/loan". The form has an input field for "interest" set to "2000.0". The developer tools are open, showing the "Sources" tab with the file "applyproduct.js". The line of code `if(interest > 200){ interest = 200; }` is no longer commented out.

: 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.