



# 웹해킹이란 무엇인가?

---

## Web Hacking Tutorial

*세션을 이용한 개인 정보 해킹과 세션 보안의 중요성*

**[BOSS] 손 우 규**

<https://github.com/swk3169/web-hacking>

---

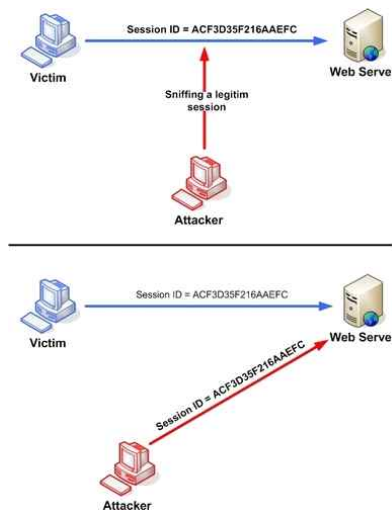
## 목차

|                                  |           |
|----------------------------------|-----------|
| <b>1. 세션 하이재킹</b>                | <b>3</b>  |
| 1.1 정의                           |           |
| 1.2 위험성                          |           |
| 1.3 공격법                          |           |
| 1.4 방어법                          |           |
| <b>2. 실습</b>                     | <b>7</b>  |
| 2.1 세션 관리와 개인 정보 출력              |           |
| 2.2 세션을 이용한 개인 정보 해킹과 세션 보안의 중요성 |           |
| <b>3. 참조</b>                     | <b>14</b> |

---

## 1. 세션 하이재킹

### 1.1 정의



세션 하이재킹 공격은 두 시스템 간 연결이 활성화된 상태, 즉 로그인된 상태를 가로채는 것을 말한다. 서버와 클라이언트가 TCP를 이용해서 통신하고 있을 때, RST 패킷을 보내 일시적으로 TCP 세션을 끊고 시퀀스 넘버를 새로 생성해 세션을 빼앗아 인증을 회피하는 공격이다.

세션 하이재킹은 세션 가로채기라고 한다. 세션이 사용자와 컴퓨터, 또는 두 대의 컴퓨터 간의 활성화된 상태를 나타낸다. 가장 쉬운 세션 하이재킹은 누군가 작업을 하다가 잠시 자리를 비운 PC를 몰래 사용해 원하는 작업을 하는 것이다.

### 1.2 위험성

- 시스템을 침투하는 방법 중에서 가장 어려운 분야 중에 하나가 정당한 접근 권한을 획득하기 위한 패스워드를 얻는 일이다. 정당한 접근 권한을 획득한 사용자가 패스워드를 성공적으로 입력하여 로그인한 후에 공격자는 현재의 연결된 세션을 넘겨받아 그 사용자처럼 위장할 수 있다. 이렇게 함으로써 공격자는 사용자의 패스워드를 얻기 위한 노력은 전혀 필요치 않게 된다.
- 세션 하이재킹은 현재 연결중인 세션을 하이재킹하기 위한 공격 기법이다. 하이재킹을 함으로써 얻어지는 가장 큰 이점은 서버로의 접근 권한을 얻기 위한 ID와 패스워드를 사용하는 인증 절차를 건너뛸 수 있다는 것이다.
- 대부분의 컴퓨터는 취약하다. 또한 하이재킹을 막기 위한 특별한 대응책이 없다. 특정 소프트웨어를 사용하면 하이재킹은 무척 간단하다.

### 1.3 공격법

- **Session ID 공격**

Session ID에 대한 공격은 크게 2 단계로 이루어진다. 1단계는 Session ID 취득 단계이고 2단계는 취득한 Session ID 공략 단계이다.

공격자는 Session ID를 취득하기 위하여 웹 서버와 웹 클라이언트의 트래픽을 직접적으로 스니핑하거나, 웹 서버 상에 공격 코드를 삽입하여 두고 사용자가 클릭할 때 null, Session ID 값을 전송받을 수 있도록 한다. 웹 서버와 웹 클라이언트 사이의 트래픽을 직접적으로 스니핑하는 방법은 일반적인 네트워크 트래픽 스니핑 기법을 통해 가능하다.

- **네트워크 트래픽 스니핑을 통한 HTTP 요청 헤더 안의 쿠키값**

```
..comAccept-Language: koContent-Type:
application/x-www-form-urlencodedAccept-Encoding: gzip, deflateUser-Agent:
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)Host:
ibn.kbstar.comContent-Length: 297Connection: Keep-AliveCache-Control:
no-cachenull: Session=QJ48621878865
```

웹 서버 상의 HTML 코드 삽입이 가능한 페이지는 주로 사용자가 글을 게시할 수 있는 게시판이나 자료실 등에 존재한다. 정상적인 글을 게재하는 대신 공격자는 HTML 코드 및 스크립트를 심어 넣는다. 일반 사용자는 해당 게시물을 열람하게 될 때 자신도 모르는 사이 null, Session ID 정보가 제 3의 공격자 서버나 이메일로 전송되게 된다. (이러한 공격 기법을 Cross-Site Scripting이라고 부른다.)

- **쿠키값 탈취를 위한 HTML 삽입**

```
<form name=f method=POST action="http://host/hello.php"><input type=hidden
name="name" value="<script>alert(doc-ument.cook-ie)</script>"></form><script>f
.submit()</script>
```

결과적으로 공격자는 취득한 타인의 Session ID 값을 웹 서버에 요청함으로써 HTTP Session Hijacking 공격을 시도할 수 있다. 물론 이 공격이 성공하려면 Session ID 값이 유효해야 하므로, 사용자가 로그인 한 상태에서 공격이 이루어져야 하거나 Session ID 값의 유지 시간이 긴 경우라는 제한 사항이 필요하다.

그러나, 기본적으로 잘못 설계된 세션 관리 기법을 사용하고 있는 웹 서버는 이러한 Hijacking 공격에 취약할 수 밖에 없다. 굳이 타인의 Session ID 값을 직, 간접적으로 취득하지 않고도 무작위 추측 대입(Brute-force Guessing)함으로써

공격이 가능하다. 공격자는 정상적인 로그인 과정시 분석한 자신의 null 값, 웹 브라우저의 주소창의 URL, 웹 페이지 폼 소스 hidden field 내에 노출된 Session ID 값 자체를 분석한다.

Session ID 생성 방식의 취약점을 파악한 후, 공격자의 컴퓨터에서 로컬 프록시 툴이나 웹 브라우저 창 URL 주소창에서 직접 Session ID 값 단일 대입을 시도한다. 더 나아가 자동적인 Session ID 대입 스크립트를 작성하여 공격할 수도 있다.

### **네트워크 트래픽 스니핑을 통한 HTTP 요청 헤더 안의 쿠키값**

```
..comAccept-Language: koContent-Type:
application/x-www-form-urlencodedAccept-Encoding: gzip, deflateUser-Agent:
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)Host:
ibn.kbstar.comContent-Length: 297Connection: Keep-AliveCache-Control:
no-cachenull: Session=QJ48621878865
```

## **1.4 방어법**

### **• 생성범위 지정**

Session ID 생성 범위값을 사용자 수에 대비하여 충분히 큰 값으로 설정한다.

### **• 랜덤 생성**

Session ID는 가능한 한 추측 불가능(random)하게 생성한다. 무작위 추측 대입 공격을 할 때 공격자는 그만큼 더 많은 시간을 투입하여야 하며 현재 연결되어 활성화 되어 있는 유효한 Session ID 값을 찾는 확률은 낮아진다.

### **• 주기적 생성**

Session Timeout 기능과 Session ID 재생성 기능을 사용한다. 일정 시간 동안 활동이 없는 사용자는 새로운 Session ID로 재로그인 하도록 하고, 사용자 로그아웃 시에는 Session ID 값을 폐기한다. 장시간 접속이 필요한 어플리케이션의 경우에는 일정 주기마다 Session ID값을 자동으로 재생성하여 세션을 유지하도록 한다.

### **• 사용자 잠금 기능**

무작위 추측 대입(Brute-force Guessing)에 대비하여 일정 회수 이상의 인증 실패 시에는 사용자 잠금 기능을 구현한다.

- **사용자 인증 제한**

로그인 이후에도 중요한 서비스 이용 시에는 사용자 인증을 통하여 인가된 사용자만이 해당 서비스를 이용할 수 있도록 통제한다.

- **변수 암호화**

null 내용 안의 Session ID와 기타 변수 값 자체를 암호화한다.

- **SSL 구현**

웹 서비스 자체의 중요성에 따라 null이 전달되는 방식을 SSL로 구현함으로써 스니핑 공격에 대응할 수도 있다.

- **주기적 필터링**

웹 서비스 상에 공격자가 HTML 공격 코드 삽입이 가능한 페이지가 있는지 점검한다. 직접적인 공격 코드 삽입을 차단할 수 있도록 특수 문자 및 스크립트 코드를 필터링하여야 한다.

## 2. 실습

### 2.1 세션 관리와 개인 정보 출력

```
Main.java  main.java x
10 import java.io.BufferedReader;
11
12 public class main
13 {
14     public static void main(String[] args) throws Exception {
15
16         String memberID = "swk";
17         String memberPassword = "test";
18         String target = "http://dowellcomputer.com/member/memberLoginAction.jsp?memberID=" + memberID + "&memberPassword=" + memberPassword;
19         HttpURLConnection con = (HttpURLConnection) new URL(target).openConnection();
20         String cookie = "";
21         String temp = con.getHeaderField("Set-Cookie");
22         if(temp != null)
23         {
24             cookie = temp;
25         }
26         System.out.println("현재 당신의 세션은 : " + cookie + "입니다.");
27     }
28 }
29
30 Problems  Javadoc  Declaration  Console x  Coverage  Debug
terminated> Main [Java Application] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (2018. 4. 16. 오후 5:55:49)
현재 당신의 세션은 : JSESSIONID=7DD18B033D3DA35DB053B38DDE4321CA; Path=/; HttpOnly입니다.
```

```
Main.java  main.java x
6 public class main
7 {
8     public static void main(String[] args) throws Exception {
9
10         String memberID = "swk";
11         String memberPassword = "test";
12         String target = "http://dowellcomputer.com/member/memberLoginAction.jsp?memberID=" + memberID + "&memberPassword=" + memberPassword;
13         HttpURLConnection con = (HttpURLConnection) new URL(target).openConnection();
14         String cookie = "";
15         String temp = con.getHeaderField("Set-Cookie");
16         if(temp != null)
17         {
18             cookie = temp;
19         }
20         System.out.println("현재 당신의 세션은 : " + cookie + "입니다.");
21         con = (HttpURLConnection) new URL("http://dowellcomputer.com/member/memberUpdateForm.jsp?ID=" + memberID).openConnection();
22         con.setRequestProperty("Cookie", cookie);
23         BufferedReader br = new BufferedReader(new InputStreamReader(con.getInputStream(), "UTF-8"));
24         while((temp = br.readLine()) != null)
25         {
26             System.out.println(temp);
27         }
28         con.disconnect();
29         br.close();
30     }
31 }
32
33 Problems  Javadoc  Declaration  Console x  Coverage  Debug
terminated> Main [Java Application] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (2018. 4. 16. 오후 6:00:22)
<td><b>swk</b></td>
</tr>
<tr>
<td class="middle">비밀번호</td>
<td><input type="password" name="memberPassword" value="test"></td>
</tr>
<tr>
<td class="middle">닉네임</td>
<td><input type="text" name="memberNickname" value="swk"></td>
</tr>
<tr>
<td class="middle">이메일</td>
<td><input type="text" name="memberEmail" value="swk3169@gmail.com"></td>
</tr>
```

## 회원 정보 수정

|      |                   |
|------|-------------------|
| 아이디  | swk               |
| 비밀번호 | ....              |
| 닉네임  | swk               |
| 이메일  | swk3169@gmail.com |

수정

메인 화면

회원탈퇴

```

12 String target = "http://dowellcomputer.com/member/memberLoginAction.jsp?memberID=" + memberID + "&memberPassword=" + memberPassword;
13 HttpURLConnection con = (HttpURLConnection) new URL(target).openConnection();
14 String cookie = "";
15 String temp = con.getHeaderField("Set-Cookie");
16 if(temp != null)
17 {
18     cookie = temp;
19 }
20 System.out.println("현재 당신의 세션은 : " + cookie + "입니다.");
21 con = (HttpURLConnection) new URL("http://dowellcomputer.com/member/memberUpdateForm.jsp?ID=" + memberID).openConnection();
22 con.setRequestProperty("Cookie", cookie);
23 BufferedReader br = new BufferedReader(new InputStreamReader(con.getInputStream(), "UTF-8"));
24 while((temp = br.readLine()) != null)
25 {
26     if(temp.contains("value=\"") && temp.contains("memberNickname"))
27     {
28         System.out.println("해당 사용자의 아이디 : " + temp.split("value=\"")[1].split("\">")[0]);
29     }
30     if(temp.contains("value=\"") && temp.contains("memberPassword"))
31     {
32         System.out.println("해당 사용자의 비밀번호 : " + temp.split("value=\"")[1].split("\">")[0]);
33     }
34     if(temp.contains("value=\"") && temp.contains("memberEmail"))
35     {
36         System.out.println("해당 사용자의 이메일 : " + temp.split("value=\"")[1].split("\">")[0]);
37     }
38 }

```

Problems Javadoc Declaration Console Coverage Debug

<terminated> Main [Java Application] C:\Program Files\Java\jdk1.8.0\_144\bin\javaw.exe (2018. 4. 16. 오후 7:04:15)

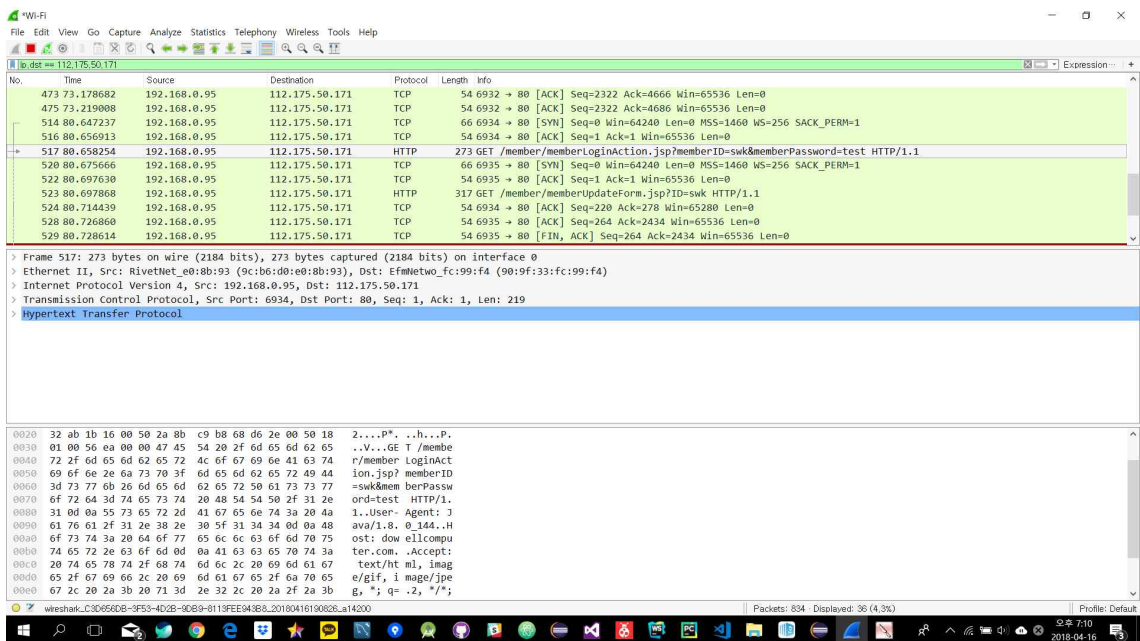
현재 당신의 세션은 : JSESSIONID=EF67907471146D1B23E268FD65EACE62; Path=/; HttpOnly입니다.

해당 사용자의 비밀번호 : test

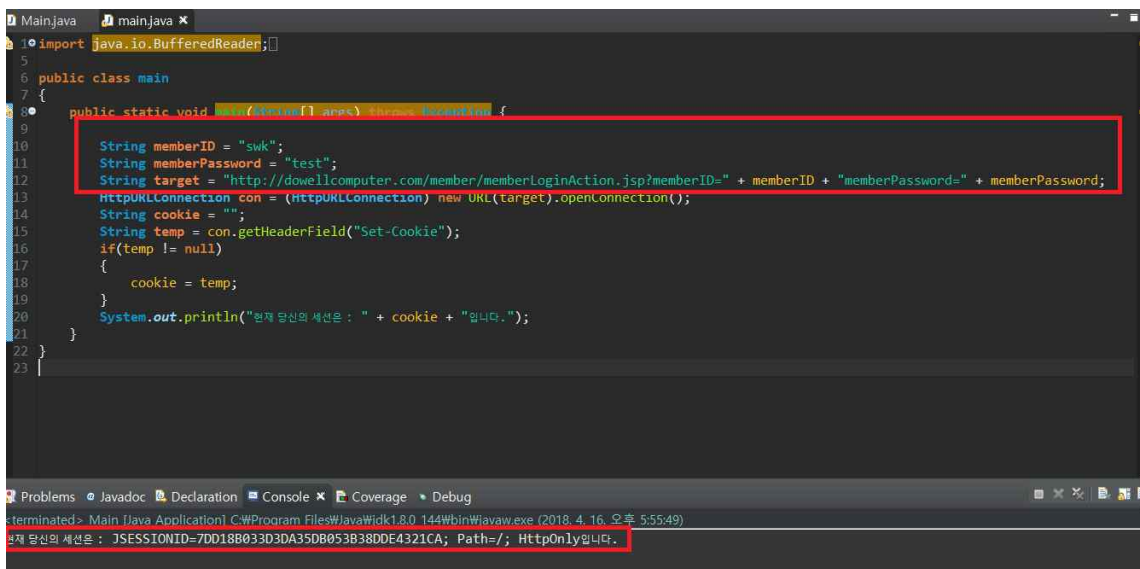
해당 사용자의 아이디 : swk

해당 사용자의 이메일 : swk3169@gmail.com





## 2.2 세션을 이용한 개인 정보 해킹과 세션 보안의 중요성



```

MainJava  mainjava
6 public class main
7 {
8     public static void main(String[] args) throws Exception {
9
10         String memberID = "swk";
11         String memberPassword = "test";
12         String target = "http://dowellcomputer.com/member/memberLoginAction.jsp?memberID=" + memberID + "&memberPassword=" + memberPassword;
13         HttpURLConnection con = (HttpURLConnection) new URL(target).openConnection();
14         String cookie = "";
15         String temp = con.getHeaderField("Set-Cookie");
16         if(temp != null)
17         {
18             cookie = temp;
19         }
20         System.out.println("현재 당신의 세션은 : " + cookie + "입니다.");
21         con = (HttpURLConnection) new URL("http://dowellcomputer.com/member/memberUpdateForm.jsp?ID=" + memberID).openConnection();
22         con.setRequestProperty("Cookie", cookie);
23         BufferedReader br = new BufferedReader(new InputStreamReader(con.getInputStream(), "UTF-8"));
24         while((temp = br.readLine()) != null)
25         {
26             System.out.println(temp);
27         }
28         con.disconnect();
29         br.close();
30     }
31 }

```

```

Problems  Javadoc  Declaration  Console  Coverage  Debug
terminated> Main [Java Application] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (2018. 4. 16. 오후 6:00:22)
<td><b>swk</b></td>
</tr>
<tr>
<td class="middle">비밀번호</td>
<td><input type="password" name="memberPassword" value="test"></td>
</tr>
<tr>
<td class="middle">닉네임</td>
<td><input type="text" name="memberNickname" value="swk"></td>
</tr>
<tr>
<td class="middle">이메일</td>
<td><input type="text" name="memberEmail" value="swk3169@gmail.com"></td>
</tr>

```

```

12     String target = "http://dowellcomputer.com/member/memberLoginAction.jsp?memberID=" + memberID + "&memberPassword=" + memberPassword;
13     HttpURLConnection con = (HttpURLConnection) new URL(target).openConnection();
14     String cookie = "";
15     String temp = con.getHeaderField("Set-Cookie");
16     if(temp != null)
17     {
18         cookie = temp;
19     }
20     System.out.println("현재 당신의 세션은 : " + cookie + "입니다.");
21     con = (HttpURLConnection) new URL("http://dowellcomputer.com/member/memberUpdateForm.jsp?ID=" + memberID).openConnection();
22     con.setRequestProperty("Cookie", cookie);
23     BufferedReader br = new BufferedReader(new InputStreamReader(con.getInputStream(), "UTF-8"));
24     while((temp = br.readLine()) != null)
25     {
26         if(temp.contains("value=\"") && temp.contains("memberNickname"))
27         {
28             System.out.println("해당 사용자의 아이디 : " + temp.split("value=\"")[1].split("\"")[0]);
29         }
30         if(temp.contains("value=\"") && temp.contains("memberPassword"))
31         {
32             System.out.println("해당 사용자의 비밀번호 : " + temp.split("value=\"")[1].split("\"")[0]);
33         }
34         if(temp.contains("value=\"") && temp.contains("memberEmail"))
35         {
36             System.out.println("해당 사용자의 이메일 : " + temp.split("value=\"")[1].split("\"")[0]);
37         }
38     }

```

```

Problems  Javadoc  Declaration  Console  Coverage  Debug
terminated> Main [Java Application] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (2018. 4. 16. 오후 7:04:15)
현재 당신의 세션은 : JSESSIONID=EF67907471146D1B23E268FD65EACE62; Path=/; HttpOnly입니다.
해당 사용자의 비밀번호 : test
해당 사용자의 아이디 : swk
해당 사용자의 이메일 : swk3169@gmail.com

```

Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Filter: [No filter] [112.175.50.171]

| No. | Time      | Source       | Destination    | Protocol | Length | Info  |
|-----|-----------|--------------|----------------|----------|--------|---|
| 473 | 73.178682 | 192.168.0.95 | 112.175.50.171 | TCP      | 54     | 6932 → 80 [ACK] Seq=2322 Ack=4666 Win=65536 Len=0                           |
| 475 | 73.219808 | 192.168.0.95 | 112.175.50.171 | TCP      | 54     | 6932 → 80 [ACK] Seq=2322 Ack=4686 Win=65536 Len=0                           |
| 514 | 80.647237 | 192.168.0.95 | 112.175.50.171 | TCP      | 66     | 6934 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1           |
| 516 | 80.656913 | 192.168.0.95 | 112.175.50.171 | TCP      | 66     | 6934 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0                                 |
| 517 | 80.658254 | 192.168.0.95 | 112.175.50.171 | HTTP     | 273    | GET /member/memberLoginAction.jsp?memberID=swk&memberPassword=test HTTP/1.1 |
| 520 | 80.675666 | 192.168.0.95 | 112.175.50.171 | TCP      | 66     | 6935 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1           |
| 522 | 80.697630 | 192.168.0.95 | 112.175.50.171 | TCP      | 54     | 6935 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0                                 |
| 523 | 80.697868 | 192.168.0.95 | 112.175.50.171 | HTTP     | 317    | GET /member/memberUpdateForm.jsp?ID=swk HTTP/1.1                            |
| 524 | 80.714439 | 192.168.0.95 | 112.175.50.171 | TCP      | 54     | 6934 → 80 [ACK] Seq=220 Ack=278 Win=65280 Len=0                             |
| 528 | 80.726860 | 192.168.0.95 | 112.175.50.171 | TCP      | 54     | 6935 → 80 [ACK] Seq=264 Ack=2434 Win=65536 Len=0                            |
| 529 | 80.728614 | 192.168.0.95 | 112.175.50.171 | TCP      | 54     | 6935 → 80 [FIN, ACK] Seq=264 Ack=2434 Win=65536 Len=0                       |

Frame 517: 273 bytes on wire (2184 bits), 273 bytes captured (2184 bits) on interface 0  
Ethernet II, Src: RixNetNet\_00:8b:93 (9c:b6:d0:00:8b:93), Dst: EfmNetwo\_fc:99:fa (90:9f:33:fc:99:fa)  
Internet Protocol Version 4, Src: 192.168.0.95, Dst: 112.175.50.171  
Transmission Control Protocol, Src Port: 6934, Dst Port: 80, Seq: 1, Ack: 1, Len: 219  
Hypertext Transfer Protocol

0020 32 ab 1b 16 00 50 2a 80 c9 b8 68 d0 2e 00 50 18 2...P...h...P.  
0030 01 00 56 6a 00 00 47 45 54 20 2f 6d 65 6d 62 65 ..V...GE T /membe  
0040 72 2f 6d 65 6d 62 65 72 4c 6f 67 69 6e 41 63 74 r/member LoginAct  
0050 69 6f 6e 2e 6a 73 70 3f 6d 65 6d 62 65 72 49 44 ion.jsp? memberID  
0060 3d 73 77 6b 26 6d 65 6d 62 65 72 50 61 73 73 77 =swk&mem berPassw  
0070 6f 72 64 3d 74 65 73 74 20 48 54 54 50 2f 31 2e ord=test HTTP/1.  
0080 31 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4a 1..User- Agent: J  
0090 61 76 61 2f 31 2e 38 2e 30 5f 31 34 3a 0d 0a 48 ava/1.8. 0\_144..H  
00a0 6f 73 74 3a 20 64 6f 77 65 6c 6c 63 6f 6d 70 75 ost: dow ellcompu  
00b0 74 65 72 2e 63 6f 6d 0d 0a 41 63 63 65 70 74 3a ter.com. .Accept:  
00c0 20 74 65 78 74 2f 68 74 6d 6c 2c 20 69 6d 61 67 text/ht ml, imag  
00d0 65 2f 67 69 66 2c 20 69 6d 61 67 65 2f 6a 70 65 e/gif, i mage/jpe  
00e0 67 2c 20 2a 3b 20 71 3d 2e 32 2c 20 2a 2f 2a 3b g, \*; q= .2, \*/\*;  
00f0 6e 3a 20 6b 65 65 70 2d 61 6c 69 76 65 0d 0a 0d n: keep- alive...

Wireshark-C3C65D8-3F53-4D28-8D89-813FEE94588-20180416160006a14200

Packets: 138023 Displayed: 98 (0.0%) Profile: Default

2018-04-16

Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Filter: [No filter] [112.175.50.171]

| No.     | Time        | Source       | Destination    | Protocol | Length | Info  |
|---------|-------------|--------------|----------------|----------|--------|---|
| 1397... | 2792.770281 | 192.168.0.95 | 112.175.50.171 | TCP      | 66     | 7355 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1           |
| 1397... | 2792.781643 | 192.168.0.95 | 112.175.50.171 | TCP      | 54     | 7355 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0                                 |
| 1397... | 2792.782894 | 192.168.0.95 | 112.175.50.171 | HTTP     | 273    | GET /member/memberLoginAction.jsp?memberID=swk&memberPassword=test HTTP/1.1 |
| 1397... | 2792.818998 | 192.168.0.95 | 112.175.50.171 | TCP      | 66     | 7356 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1           |
| 1397... | 2792.830213 | 192.168.0.95 | 112.175.50.171 | TCP      | 54     | 7356 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0                                 |
| 1397... | 2792.839942 | 192.168.0.95 | 112.175.50.171 | HTTP     | 317    | GET /member/memberUpdateForm.jsp?ID=swk HTTP/1.1                            |
| 1397... | 2792.842943 | 192.168.0.95 | 112.175.50.171 | TCP      | 54     | 7356 → 80 [ACK] Seq=264 Ack=2434 Win=65536 Len=0                            |
| 1397... | 2792.849380 | 192.168.0.95 | 112.175.50.171 | TCP      | 54     | 7356 → 80 [FIN, ACK] Seq=264 Ack=2434 Win=65536 Len=0                       |
| 1397... | 2792.855026 | 192.168.0.95 | 112.175.50.171 | TCP      | 54     | 7355 → 80 [ACK] Seq=220 Ack=278 Win=65280 Len=0                             |
| 1397... | 2792.859229 | 192.168.0.95 | 112.175.50.171 | TCP      | 54     | 7356 → 80 [ACK] Seq=265 Ack=2435 Win=65536 Len=0                            |
| 1397... | 2792.872712 | 192.168.0.95 | 112.175.50.171 | TCP      | 54     | 7355 → 80 [RST, ACK] Seq=220 Ack=278 Win=0 Len=0                            |

Hypertext Transfer Protocol

GET /member/memberLoginAction.jsp?memberID=swk&memberPassword=test HTTP/1.1\r\n

[Expert Info (Chat/Sequence): GET /member/memberLoginAction.jsp?memberID=swk&memberPassword=test HTTP/1.1\r\n]

Request Method: GET

Request URI: /member/memberLoginAction.jsp?memberID=swk&memberPassword=test

Request URI Path: /member/memberLoginAction.jsp

Request URI Query: memberID=swk&memberPassword=test

Request URI Query Parameter: memberID=swk

Request URI Query Parameter: memberPassword=test

Request Version: HTTP/1.1

User-Agent: Java/1.8.0\_144\r\n

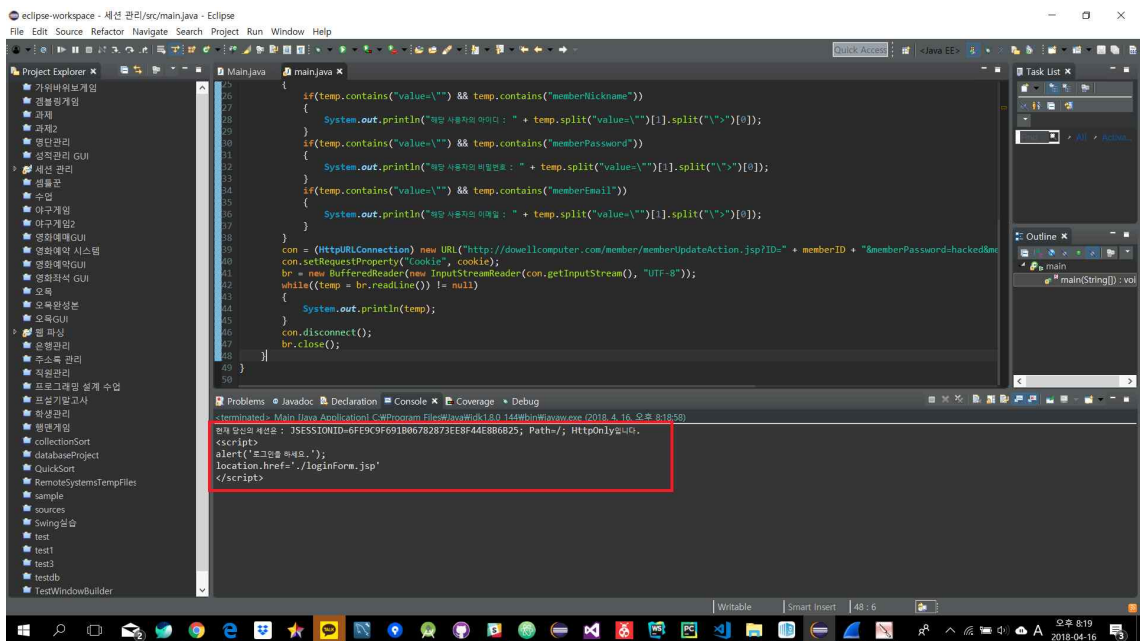
Host: dowellcomputer.com\r\n

0050 69 6f 6e 2e 6a 73 70 3f 6d 65 6d 62 65 72 49 44 ion.jsp? memberID  
0060 3d 73 77 6b 26 6d 65 6d 62 65 72 50 61 73 73 77 =swk&mem berPassw  
0070 6f 72 64 3d 74 65 73 74 20 48 54 54 50 2f 31 2e ord=test HTTP/1.  
0080 31 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4a 1..User- Agent: J  
0090 61 76 61 2f 31 2e 38 2e 30 5f 31 34 3a 0d 0a 48 ava/1.8. 0\_144..H  
00a0 6f 73 74 3a 20 64 6f 77 65 6c 6c 63 6f 6d 70 75 ost: dow ellcompu  
00b0 74 65 72 2e 63 6f 6d 0d 0a 41 63 63 65 70 74 3a ter.com. .Accept:  
00c0 20 74 65 78 74 2f 68 74 6d 6c 2c 20 69 6d 61 67 text/ht ml, imag  
00d0 65 2f 67 69 66 2c 20 69 6d 61 67 65 2f 6a 70 65 e/gif, i mage/jpe  
00e0 67 2c 20 2a 3b 20 71 3d 2e 32 2c 20 2a 2f 2a 3b g, \*; q= .2, \*/\*;  
00f0 20 71 3d 2e 32 0d 0a 43 6f 6e 6e 65 63 74 69 6f n: keep- alive...

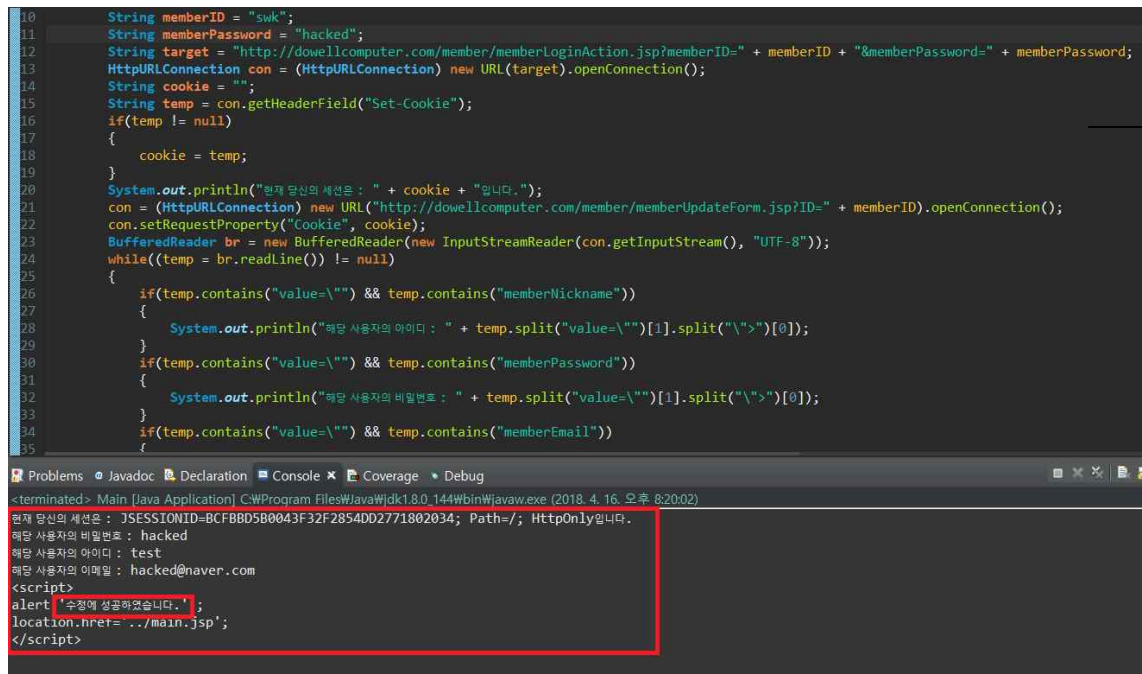
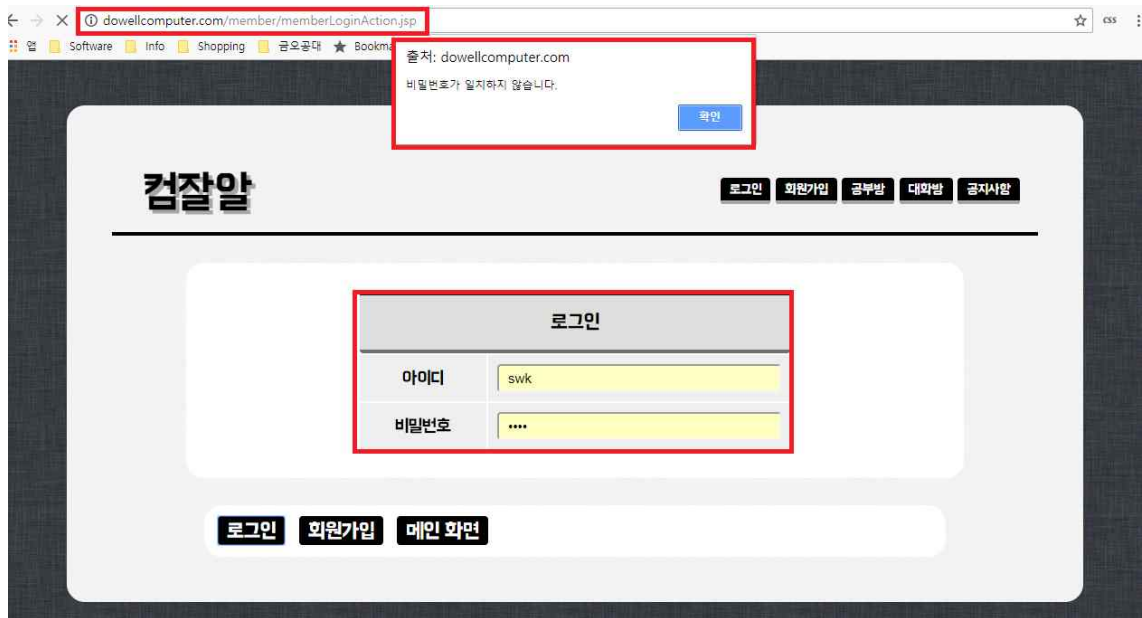
HTTP Request+URI Query (http.request.uri.query): 32 bytes

Packets: 141519 Displayed: 47 (0.0%) Profile: Default

2018-04-16







← → ↻ ① 안전하지 않음 dowellcomputer.com/member/memberUpdateForm.jsp?ID=swk

앱 Software Info Shopping 금융공대 ★ Bookmarks 새 탭 네이버 : 나의 경제력 Facebook GitHub

**컴잘알** **로그아웃** **회원 정보 수정** **공부방** **대화방** **공지사항**

**회원 정보 수정**

|      |                  |
|------|------------------|
| 아이디  | swk              |
| 비밀번호 | *****            |
| 닉네임  | test             |
| 이메일  | hacked@naver.com |

**수정** **메인 화면** **회원탈퇴**

### 3. 참조

<http://copycode.tistory.com/81>