Vulnerabilities and Countermeasures of Commercial Online Music Streaming Services in Korea

Sangsik Lee[1], Donghyun Choi[2], Dongho Won[2] and Seungjoo Kim[3*]
[1]Information Security Group, Sungkyunkwan University, Suwon-si, Korea
Email: Risk Management Division of LOTTE Capital, Seoul, Korea
crackerless@gmail.com
[2]Information Security Group, Sungkyunkwan University, Suwon-si, Korea
{dhchoi, dhwon}@security.re.kr
[3]CIST (Center for Information Security Technologies), Korea University, Seoul, Korea
skim71@korea.ac.kr

***Abstract*—The music industry is rapidly moving from analog to digital, and most of the big portal sites provide commercial online music streaming services. In this paper, we analyze the vulnerabilities of commercial online music streaming services provided by Korea's major portal sites (Dosirak, Cyworld, and Naver). We show attacks on commercial online music streaming services that lead to an infringement of copyright and propose appropriate countermeasures.**

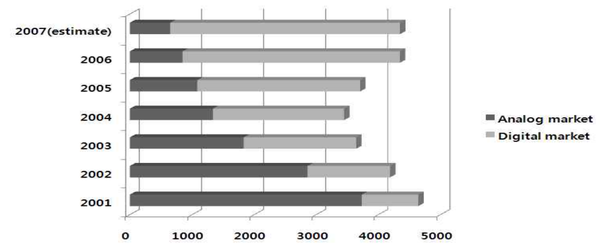
***Index Terms*—Security, Commercial Online Music Streaming Service,**

I. INTRODUCTION

The music industry is moving from analog to digital and the demand for paid online music is increasing under the influence of reinforcement of the copyright law. Therefore, online music service companies are expanding the music industry by introducing various services. For example, portal sites such as Dosirak, Cyworld [2] and Naver [3], provide background music services for microsites, blogs, and online communities, and opens up the paid online music market.

Figure 1 shows the size of the digital and analog music markets. In the past, the analog music market was bigger, but today the digital music market is much larger. While the digital music market has increased, digital music services have not been analyzed for security issues. If the services have vulnerabilities and adversaries use these vulnerabilities, the attack damages the content owner's intellectual property. This is a huge problem, which has been neglected. Therefore, this paper analyzes the processes of current online digital music streaming services and points out problems with regard to protection of copyright from a technological perspective. In addition, we suggest appropriate countermeasures.

The organization of the paper is as follows. Section 2 introduces typical music streaming service processes of Korea's major portal sites. In Section 3, we analyze vulnerabilities for each of three portal sites (Dosirak, Cyworld, and Naver) and present actual attack scenarios. We suggest music streaming service countermeasures for the analyzed vulnerabilities in Section 4. Section 5 concludes the paper.



(Unit : Hundred million won)

	2001	2002	2003	2004	2005	2006	2007 (estimate)
Analog market	3733	2861	1833	1338	1087	848	650
Digital market	911	1328	1811	2112	2621	3500	3700

Figure 1. Size of digital and analog music markets [1]

II. COMMERCIAL ONLINE MUSIC STREAMING SERVICE PROCESSES

In this section, we show the typical processes of commercial online music streaming services. The major portals in Korea provide streaming services by exchanging information between a server and a client, through a web-based music player. Although the flow of information varies by service provider, the execution result remains the same. Figure 2 shows the typical process followed by the major Korean portals:

① Step 1: User selects music on a web-based music player.

② Step 2: The music player sends information of the selected music and the user authentication information to the server. Client authentication information is in the form of cookies and parameters containing the user id and information on whether the user has logged in.

③ Step 3: Upon receiving the information, the server identifies if the user is a paid user.

*Corresponding author: Seungjoo Kim (skim71@korea.ac.kr).

doi:10.4304/jsw.8.3.520-527

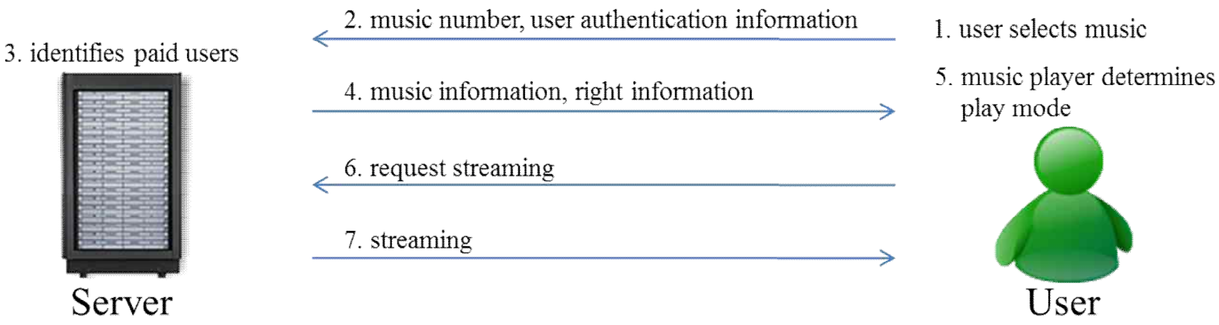

3. identifies paid users
Server
2. music number, user authentication information
4. music information, right information
6. request streaming
7. streaming
1. user selects music
5. music player determines play mode
User

Figure 2. The processes of a commercial music streaming service

④ Step 4: Depending on the result, the server transmits music and rights information. The music information consists of the title, artist, and location of the music source. The rights information consists of a variable (owner or free) that determines whether the current user has purchased the music.

⑤ Step 5: Upon receiving the information, the music player determines the play mode (preview or normal).

⑥ Step 6: The music player requests the streaming service.

⑦ Step 7: The server streams the music, and the music player renders the stream content.

III. Attack on Commercial Online Music Streaming Services

In this section, we demonstrate an attack on current music streaming services. We use Fiddler, Cooxie Toolbar, ands Burp Suite. Fiddler is a web debugging proxy, which logs all HTTP(s) traffic between a computer and the Internet [5]. The Cooxie Toolbar reveals what Internet Explorer has recorded about online activities [6]. Burp Suite is an integrated platform for attacking and testing web applications. It includes intercepting a web proxy, a hacker-oriented vulnerability scanner, and tools to decode and compare application data [7].

A. Attack by Altering Cookies (Dosirak)

Dosirak is one of the major online music websites in Korea. Dosirak stores user authentication information via cookies, which is a historical information file. When a user visits a website, a cookie is stored on the user’s computer by the server. A cookie consists of one or more name-value pairs containing bits of information. A cookie can be used for authentication, session tracking, storing site preferences, storing shopping cart contents, or anything else that can be accomplished through storing textual data [8].

An attacker can alter this cookie and impersonate a paid user. We will explain attacks on a music streaming service via cookie alteration.

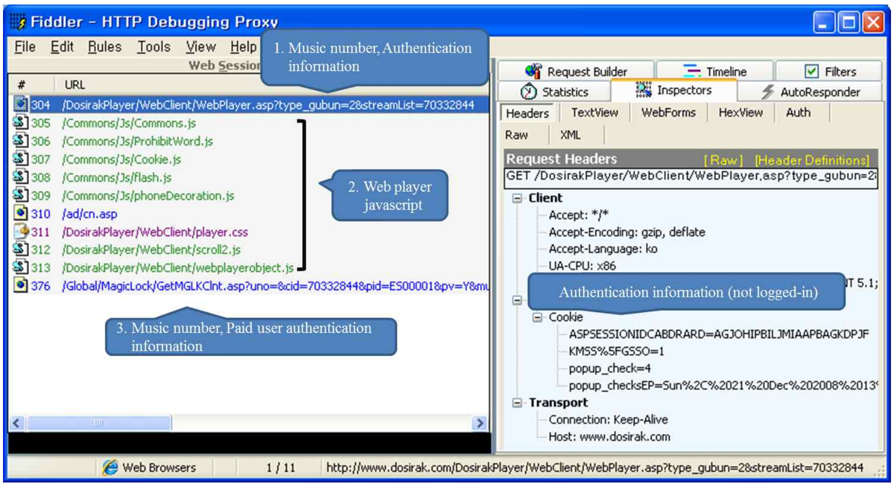
First, we explain the normal music streaming service of Dosirak when a user has not logged in. Figure 3 shows the Dosirak communications process.

① Step 1: When a user selects music, Internet Explorer transfers the song ID and authentication information to the server through a cookie.

② Step 2: The user downloads a JavaScript music player from the server.

③ Step 3: The player requests the music information from the server.

After Step 3, the music information is transferred from server to client. Figure 4 shows this information.



```
#    URL
304  /DosirakPlayer/WebClient/WebPlayer.asp?type_gubun=2&streamList=70332844
305  /Commons/Js/Commons.js
306  /Commons/Js/ProhibitWord.js
307  /Commons/Js/Cookie.js
308  /Commons/Js/flash.js
309  /Commons/Js/phoneDecoration.js
310  /ad/cn.asp
311  /DosirakPlayer/WebClient/player.css
312  /DosirakPlayer/WebClient/scroll2.js
313  /DosirakPlayer/WebClient/webplayerobject.js
376  /Global/MagicLock/GetMGLKClnt.asp?uno=&cid=70332844&pid=ES00001&pv=Y&mu
```

```
Request Headers                                [Raw]  [Header Definitions]
GET /DosirakPlayer/WebClient/WebPlayer.asp?type_gubun=2
⊟ Client
    Accept: */*
    Accept-Encoding: gzip, deflate
    Accept-Language: ko
    UA-CPU: x86
                                                                  T 5.1;
⊟
  ⊟ Cookie
      ASPSESSIONIDCABDRARD=AGJOHIPBILJMIAAPBAGKDPJF
      KMSS%5FGSSO=1
      popup_check=4
      popup_checksEP=Sun%2C%2021%20Dec%202008%2013
⊟ Transport
    Connection: Keep-Alive
    Host: www.dosirak.com
```

```
http://www.dosirak.com/DosirakPlayer/WebClient/WebPlayer.asp?type_gubun=2&streamList=70332844
```

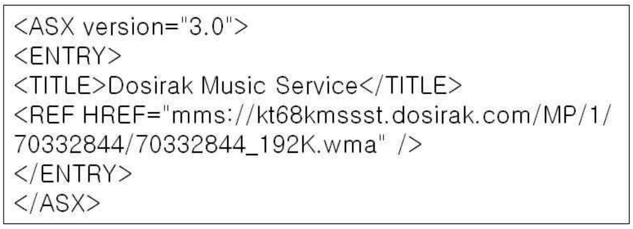
1. Music number, Authentication information

2. Web player javascript

3. Music number, Paid user authentication information

Authentication information (not logged-in)

Figure 3. Communication process of the Dosirak streaming service



```
<ASX version="3.0">
<ENTRY>
<TITLE>Dosirak Music Service</TITLE>
<REF HREF="mms://kt68kmssst.dosirak.com/MP/1/
70332844/70332844_192K.wma" />
</ENTRY>
</ASX>
```

Figure 4. Request result

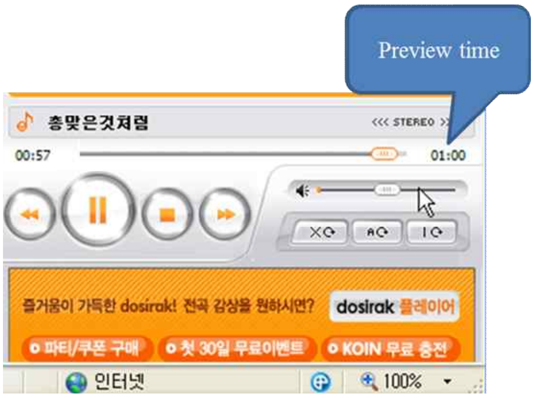
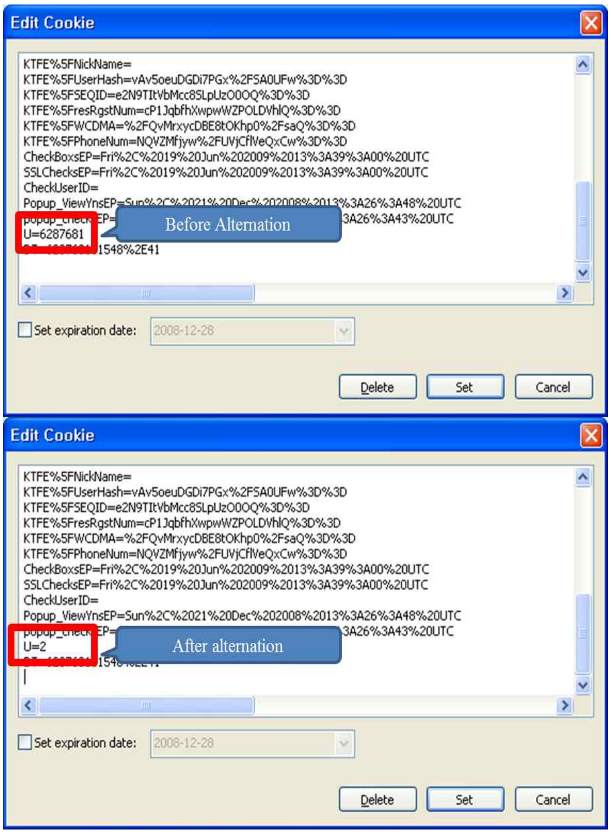


Figure 5. Music player

The information contains the address of the music service for the music player. When the music player connects to the address, it can render the content for a preview time (one minute).

Next, we show an attack of the music streaming service by altering cookies. Figure 6 shows the communication process when a user has logged in—there are many changes during log in. Among the differences between Figures 3 and 6, we concentrate on the variable U.

As a result of analysis, we found that the variable U represents a user identification number. Therefore, we try to alter variable U to obtain the right to stream music.



```
KTFE%5FNickName=
KTFE%5FUserHash=vAv5oeuDGDi7PGx%2FSA0UFw%3D%3D
KTFE%5FSEQID=e2N9TItVbMcc8SLpUzO0OQ%3D%3D
KTFE%5FresRgstNum=cP1JqbfhXwpwWZPOLDVhlQ%3D%3D
KTFE%5FWCDMA=%2FQvMrxycDBE8tOKhp0%2FsaQ%3D%3D
KTFE%5FPhoneNum=NQVZMfjyw%2FUVjCflVeQxCw%3D%3D
CheckBoxsEP=Fri%2C%2019%20Jun%202009%2013%3A39%3A00%20UTC
SSLChecksEP=Fri%2C%2019%20Jun%202009%2013%3A39%3A00%20UTC
CheckUserID=
Popup_ViewYnsEP=Sun%2C%2021%20Dec%202008%2013%3A26%3A48%20UTC
popup_check EP=                         3A26%3A43%20UTC
U=6287681
        1548%2E41
```

☐ Set expiration date: 2008-12-28

```
KTFE%5FNickName=
KTFE%5FUserHash=vAv5oeuDGDi7PGx%2FSA0UFw%3D%3D
KTFE%5FSEQID=e2N9TItVbMcc8SLpUzO0OQ%3D%3D
KTFE%5FresRgstNum=cP1JqbfhXwpwWZPOLDVhlQ%3D%3D
KTFE%5FWCDMA=%2FQvMrxycDBE8tOKhp0%2FsaQ%3D%3D
KTFE%5FPhoneNum=NQVZMfjyw%2FUVjCflVeQxCw%3D%3D
CheckBoxsEP=Fri%2C%2019%20Jun%202009%2013%3A39%3A00%20UTC
SSLChecksEP=Fri%2C%2019%20Jun%202009%2013%3A39%3A00%20UTC
CheckUserID=
Popup_ViewYnsEP=Sun%2C%2021%20Dec%202008%2013%3A26%3A48%20UTC
popup_check EP=                         3A26%3A43%20UTC
U=2
```

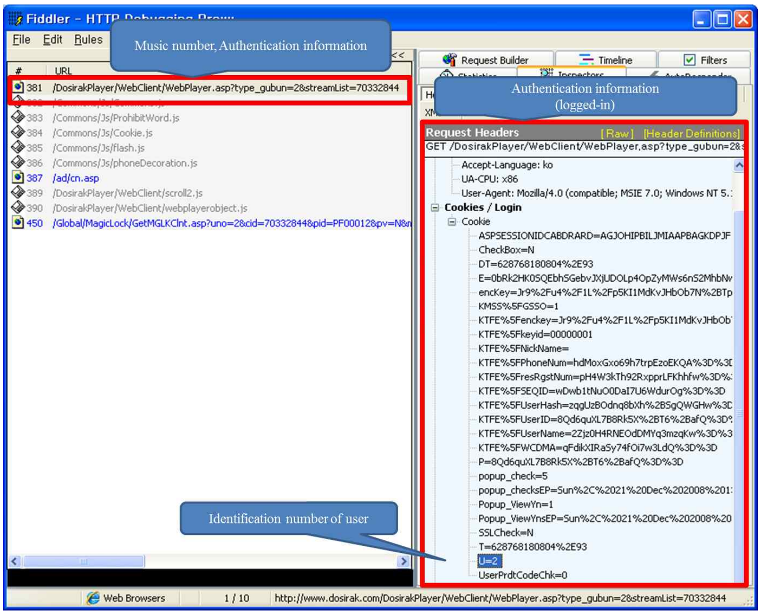
☐ Set expiration date: 2008-12-28

Figure 7. Alteration of the variable U

We alter variable U from 62876 to 2, as shown in Figure 6. After the alteration, we obtain the rights of a paid user. We also found that a user ID was changed and cyber money came in a different user (see Figure 7).

In this manner, the attacker can use another user's paid service by altering a cookie. If the altered user number represents a paid user, the attacker gets the result shown in Figure 9.

By requesting streaming with the address in Figure 9, we can listen to the entire song (see Figure 10), rather than the one minute preview. Thus, the server uses only one variable U to authenticate paid users.



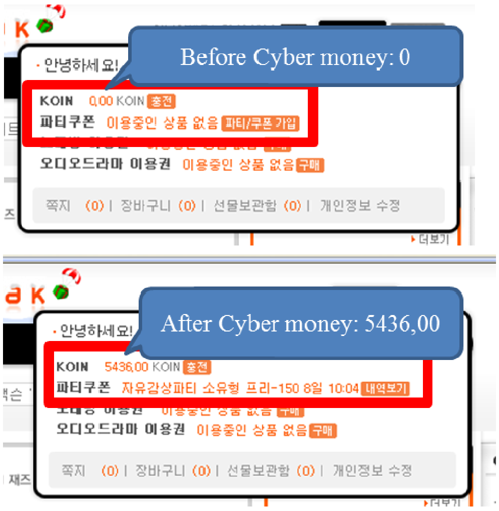
```
#    URL
381  /DosirakPlayer/WebClient/WebPlayer.asp?type_gubun=2&streamList=70332844
383  /Commons/Js/ProhibitWord.js
384  /Commons/Js/Cookie.js
385  /Commons/Js/flash.js
386  /Commons/Js/phoneDecoration.js
387  /ad/cn.asp
389  /DosirakPlayer/WebClient/scroll2.js
390  /DosirakPlayer/WebClient/webplayerobject.js
450  /Global/MagicLock/GetMGLKClnt.asp?uno=2&cid=70332844&pid=PF00012&pv=N&n
```

```
GET /DosirakPlayer/WebClient/WebPlayer.asp?type_gubun=2&s
    Accept-Language: ko
    UA-CPU: x86
    User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1
Cookies / Login
    Cookie
        ASPSESSIONIDCABDRARD=AGJOHIPBILJMIAAPBAGKDPJF
        CheckBox=N
        DT=628768180804%2E93
        E=0bRk2HK0SQEbhSGebvJXjUDOLp4OpZyMWs6nS2MhbNv
        encKey=Jr9%2Fu4%2F1L%2Fp5KI1MdKvJHbOb7N%2BTp
        KMSS%5FGSSO=1
        KTFE%5Fenckey=Jr9%2Fu4%2F1L%2Fp5KI1MdKvJHbOb
        KTFE%5Fkeyid=00000001
        KTFE%5FNickName=
        KTFE%5FPhoneNum=hdMoxGxo69h7trpEzoEKQA%3D%3D
        KTFE%5FresRgstNum=pH4W3kTh92RxpprLFKhhfw%3D%3
        KTFE%5FSEQID=wDwb1tNuO0DaI7U6WdurOg%3D%3D
        KTFE%5FUserHash=zqgUzBOdnq8bXh%2BSgQWGHw%3D
        KTFE%5FUserID=8Qd6quXL7B8Rk5X%2BT6%2BafQ%3D%
        KTFE%5FUserName=2Zjz0H4RNEOdDMYq3mzqKw%3D%3
        KTFE%5FWCDMA=qFdikXIRaSy74fOi7w3LdQ%3D%3D
        P=8Qd6quXL7B8Rk5X%2BT6%2BafQ%3D%3D
        popup_check=5
        popup_checksEP=Sun%2C%2021%20Dec%202008%201
        Popup_ViewYn=1
        Popup_ViewYnsEP=Sun%2C%2021%20Dec%202008%20
        SSLCheck=N
        T=628768180804%2E93
        U=2
        UserPrdtCodeChk=0
```

```
1 / 10    http://www.dosirak.com/DosirakPlayer/WebClient/WebPlayer.asp?type_gubun=2&streamList=70332844
```

Figure 6. Communication process when a user has logged in

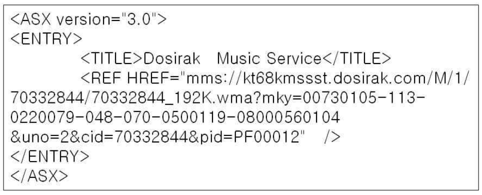
B. Attack by Altering the Message (Cyworld and Naver)



· 안녕하세요!
KOIN 0,00 KOIN 충전
파티쿠폰 이용중인 상품 없음 파티/쿠폰 가입
오디오드라마 이용권 이용중인 상품 없음 구매
쪽지 (0) | 장바구니 (0) | 선물보관함 (0) | 개인정보 수정

· 안녕하세요!
KOIN 5436,00 KOIN 충전
파티쿠폰 자유감상파티 소유형 프리-150 8일 10:04 내역보기
오디오드라마 이용권 이용중인 상품 없음 구매
쪽지 (0) | 장바구니 (0) | 선물보관함 (0) | 개인정보 수정

Figure 8. Result of cookie alteration



```
<ASX version="3.0">
<ENTRY>
          <TITLE>Dosirak   Music Service</TITLE>
          <REF HREF="mms://kt68kmssst.dosirak.com/M/1/
70332844/70332844_192K.wma?mky=00730105-113-
0220079-048-070-0500119-08000560104
&uno=2&cid=70332844&pid=PF00012"   />
</ENTRY>
</ASX>
```

Figure 9. Request result

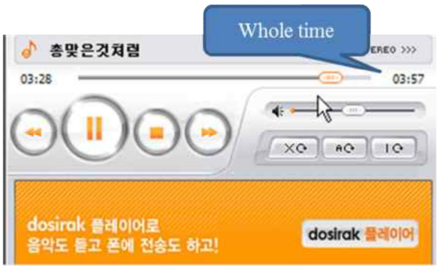


Figure 10. The music player after the alteration

TABLE I.
CYWORLD REQUEST/RESPONSE/ATTACK

Request	URL	POST /player/jukebox/43/ xml_song_list.asp HTTP/1.1	
	Information	product_seq=20864063&ndr_url=cymusic	
Response	<?xml version="1.0" encoding="EUC-KR"?> <songs> <song> <linkCode>2136421</linkCode> <title><![CDATA[title name]]></title> <artist><![CDATA[artist name]]></artist> <productSeq>20864063</productSeq> <owner>false</owner> <gradeYn>0</gradeYn> </song> </songs>		
Attack	Before		After
	<owner>false</owner>		<owner>true</owner>

In this section, we explain the message altering attack. This attack is applied to Naver and Cyworld. Naver is the most popular search portal in Korea. Cyworld is a social network service operated by SK Communications in Korea. Naver and Cyworld provide online streaming music service.

We used Fiddler for analyzing the process and Burp Suite for altering the message.

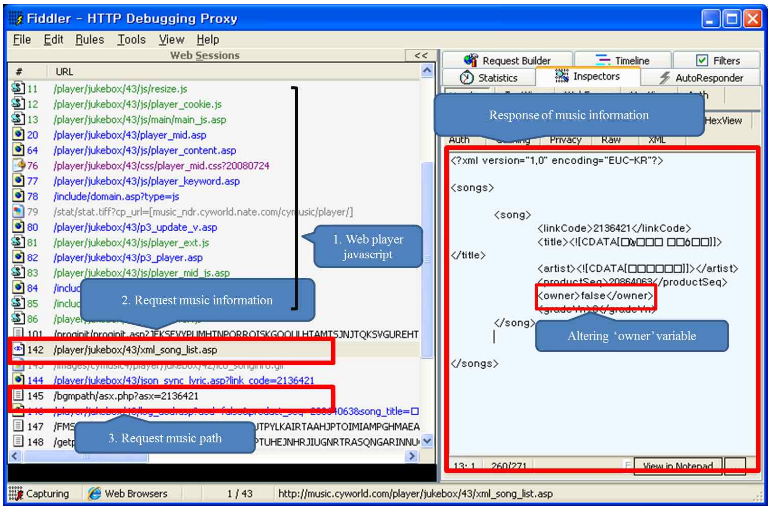
First, we explain the attack on Cyworld. Cyworld transfers music and authentication information in XML format without encryption. In such cases, the attacker can accomplish his goal by altering the message containing the authentication information.

Figure 11 shows that the user receives the music information through the */player/jukebox/43/xml_song_list.asp* page in Cyworld. This information is sent in XML format with the paid user's authentication information.

Table 1 shows some parts of this packet. The packet includes "linkCode," "title," "artist," "productSeq," "owner," and "gradeYn." Among these, the important point is the variable "owner." This variable represents the rights of the service. Therefore, we try to change the variable "owner" from false to true.

Figure 12 shows the result of the alteration—after the attack, the total play time is changed from the preview to the complete play time.

Next, we explain the attack on Naver, which is similar



```
#    URL
11   /player/jukebox/43/js/resize.js
12   /player/jukebox/43/js/player_cookie.js
13   /player/jukebox/43/js/main/main_js.asp
20   /player/jukebox/43/player_mid.asp
64   /player/jukebox/43/js/player_content.asp
76   /player/jukebox/43/css/player_mid.css?20080724
77   /player/jukebox/43/js/player_keyword.asp
78   /include/domain.asp?type=js
79   /stat/stat.tiff?cp_url=[music_ndr.cyworld.nate.com/cymusic/player/]
80   /player/jukebox/43/p3_update_v.asp
81   /player/jukebox/43/js/player_ext.js
82   /player/jukebox/43/p3_player.asp
83   /player/jukebox/43/js/player_mid_js.asp
84   /inclu
85   /inclu
86   /playe
101  /proginit/proginit.asp?JEKSFVYPUMHTNPORROISKGOOUHTAMTSJNJTQKSVGUREHT
142  /player/jukebox/43/xml_song_list.asp
144  /player/jukebox/43/json_sync_lyric.asp?link_code=2136421
145  /bgmpath/asx.php?asx=2136421
147  /FMS                                   JTPYLKAIRTAAHJPTOIMIAMPGHMAEA
148  /getp                                  PTUHEJNHRJIUGNRTRASQNGARINNU
```

```
<?xml version="1.0" encoding="EUC-KR"?>

<songs>

        <song>
                <linkCode>2136421</linkCode>
                <title><![CDATA[□□□□ □□□□□]]>
</title>
                <artist><![CDATA[□□□□□□□]]></artist>
                <productSeq>20864063</productSeq>
                <owner>false</owner>
        </song>

</songs>
```

13: 1   260/271

Capturing   Web Browsers   1 / 43   http://music.cyworld.com/player/jukebox/43/xml_song_list.asp

Figure 11. Communication process of Cyworld

to the attack on Cyworld.

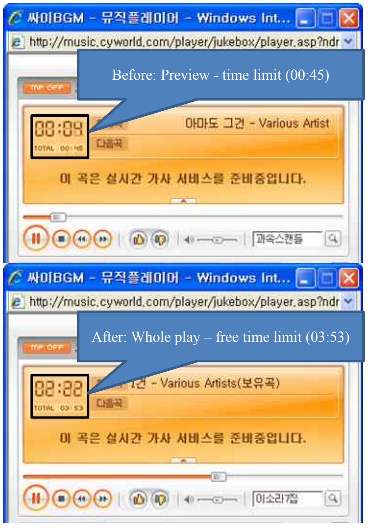

싸이BGM - 뮤직플레이어 - Windows Int...
http://music.cyworld.com/player/jukebox/player.asp?ndr
Before: Preview - time limit (00:45)
00:09
TOTAL 00:45
아마도 그건 - Various Artist
다음곡
이 곡은 실시간 가사 서비스를 준비중입니다.
과속스캔들
싸이BGM - 뮤직플레이어 - Windows Int...
http://music.cyworld.com/player/jukebox/player.asp?ndr
After: Whole play – free time limit (03:53)
02:22
TOTAL 03:53
그건 - Various Artists(보유곡)
다음곡
이 곡은 실시간 가사 서비스를 준비중입니다.
이소라7집

Figure 12. Cyworld music player

When the server receives the request for a song, it transfers the music information with paid-user authentication information to the client's music player in XML format.

Figure 13 shows that the user receives the music information with the authentication information through the */ISZone/PlayBackInfo_List.asp* page. The message is sent in XML format.

Table 2 shows some parts of this packet. This packet includes "song id," "title," "artist id," "free" and "adult," etc. The important point is the variable "free." This variable represents the rights of the service. Therefore, we try to change the variable "free" from 0 to 1.

Figure 14 shows the result of altering the variable "free." After the attack, the total song time is changed from the preview to the entire playtime.

TABLE II.
NAVER REQUEST/RESPONSE/ATTACK

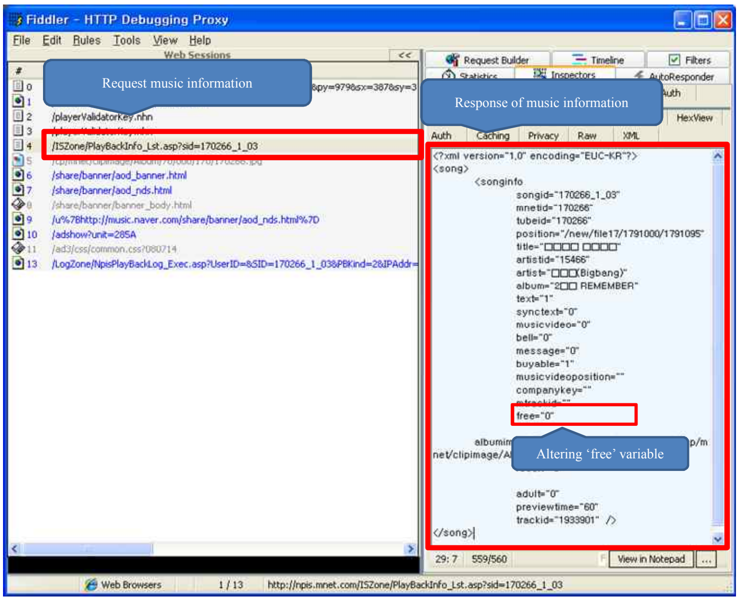
Request	URL	GET /ISZone/PlayBackInfo_Lst.asp HTTP/1.1
	Information	sid=170266_1_03

Response	<?xml version="1.0" encoding="EUC-KR"?>
	<song>
	<songinfo
	songid="170266_1_03"
	mnetid="170266"
	tubeid="170266"
	position="/new/file17/1791000/1791095"
	title="title name "
	artistid="15466"
	artist="Bigbang"
	album="2 REMEMBER"
	text="1"
	synctext="0"
	musicvideo="0"
	bell="0"
	message="0"
	buyable="1"
	musicvideoposition=""
	companykey=""
	mtrackid=""
	free="0"
	albumimgurl="http://images.hangame.co.kr/ cp/mnet/clipimage/Album/70/000/170/170266.jpg"
	result="0"
	adult="0"
	previewtime="60"
	trackid="1933901" />
	</song>	

Attack	Before	After
	<owner>false</owner>	<owner>true</owner>

C. Local JavaScript Altering Attack (Naver and Cyworld)

The music players of Naver and Cyworld use JavaScript source code. JavaScript source code is stored on the client's side and anyone can modify the source



```
#
0    &py=979&sx=387&sy=3
1
2    /playerValidatorKey.nhn
3
4    /ISZone/PlayBackInfo_Lst.asp?sid=170266_1_03
5
6    /share/banner/aod_banner.html
7    /share/banner/aod_nds.html
8    /share/banner/banner_body.html
9    /u%7Bhttp://music.naver.com/share/banner/aod_nds.html%7D
10   /adshow?unit=285A
11   /ad3/css/common.css?080714
13   /LogZone/NpisPlayBackLog_Exec.asp?UserID=&SID=170266_1_03&PBKind=2&IPAddr=
```

```
<?xml version="1.0" encoding="EUC-KR"?>
<song>
        <songinfo
                songid="170266_1_03"
                mnetid="170266"
                tubeid="170266"
                position="/new/file17/1791000/1791095"
                title="□□□□ □□□□"
                artistid="15466"
                artist="□□□(Bigbang)"
                album="2□□ REMEMBER"
                text="1"
                synctext="0"
                musicvideo="0"
                bell="0"
                message="0"
                buyable="1"
                musicvideoposition=""
                companykey=""
                mtrackid=""
                free="0"
        albumim                                        p/m
net/clipimage/A

                adult="0"
                previewtime="60"
                trackid="1933901"  />
</song>
```

http://npis.mnet.com/ISZone/PlayBackInfo_Lst.asp?sid=170266_1_03

Figure 13. Communication process of Naver

code easily. Therefore, an attack can be made by modifying the player source code to listen to complete songs.

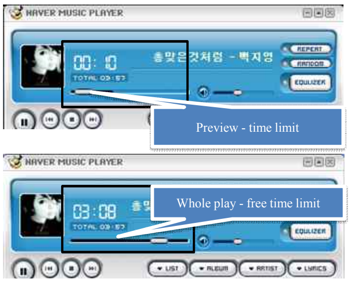
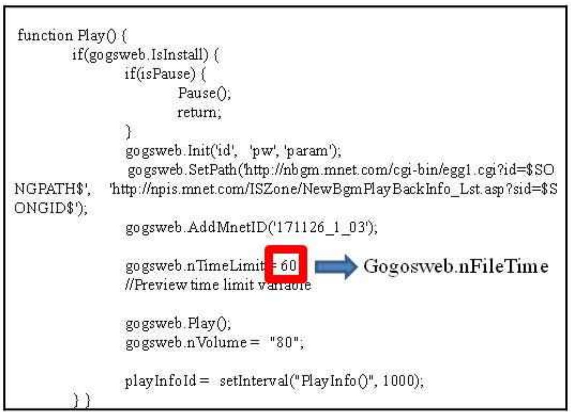


Figure 14. Naver music player



```
function Play() {
        if(gogsweb.IsInstall) {
                if(isPause) {
                        Pause();
                        return;
                }
                gogsweb.Init('id',   'pw', 'param');
                 gogsweb.SetPath('http://nbgm.mnet.com/cgi-bin/egg1.cgi?id=$SO
NGPATH$',    'http://npis.mnet.com/ISZone/NewBgmPlayBackInfo_Lst.asp?sid=$S
ONGID$');
                gogsweb.AddMnetID('171126_1_03');

                gogsweb.nTimeLimit = 60
                //Preview time limit variable

                gogsweb.Play();
                gogsweb.nVolume =  "80";

                playInfoId =  setInterval("PlayInfo()", 1000);
        } }
```

Figure 15. JavaScript of Naver music player

Figure 15 shows part of the source code of “Musicsem” of Naver. “Musicsem” allows the user to listen to background music on their forum or blog. This also provides the preview service. This service is made using simple JavaScript source code. The important point is the variable “nTimeLimit.” This variable represents the playtime. Therefore, we change the variable “gogsweb.nTimeLimit=60;” to “gogsweb.nTimeLimit = gogsweb.nFileTime;”. After the attack, the total time of the song is changed from the preview time that of the complete song.

Cyworld is also vulnerable to this attack

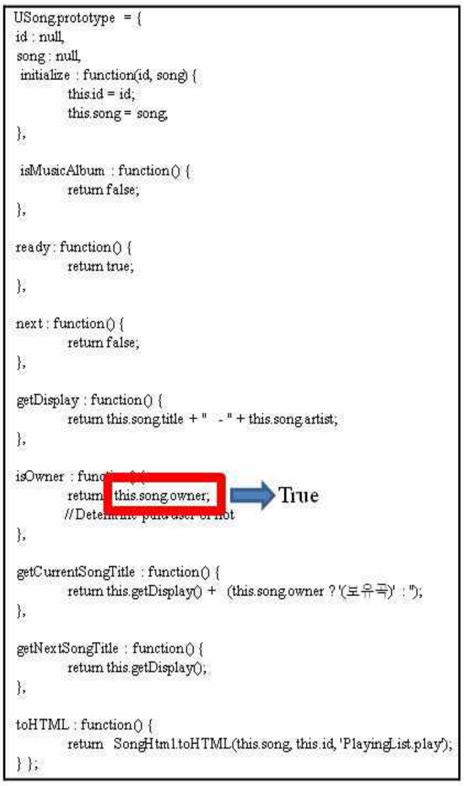
In Figure 16, by changing “return this.song.owner;” to “return true;”, the music player recognizes the attacker as a paid user. If that happens, he can listen to the entire song.

The vulnerability of a local JavaScript altering attack is due to inappropriate authentication and the developer’s trust in clients [9]. Authentication of paid users should be done in both the server and client side in a way that the attacker cannot understand.

D. Vulnerabilities and Their Effects

In this section, we summarize the vulnerabilities and their effects for each site, shown in Table 3. Although the cookie is encrypted, a vulnerability for abusing the paid service using variables has been found. However, Dosirak could prevent message and JavaScript alteration attacks by confirming additional user authentication information. Cyworld and Naver do not encrypt the authentication information, and they limit the preview in JavaScript, so the attacker can use the paid service. On the other hand, they prevent the cookie alteration attack by encrypting cookies and using several parameters for authentication.

We have shown attack scenarios and their effects using vulnerabilities that were created by applying security countermeasures to only a part of music streaming service system, rather than the whole system. These results suggest that the security level of the system is determined by the lowest security level in the system. Based on analysis made in this section, we suggest countermeasures for each vulnerability



```
USong.prototype  = {
id : null,
 song : null,
 initialize : function(id, song) {
          this.id = id;
          this.song = song,
},

 isMusicAlbum : function() {
          return false;
},

ready : function() {
          return true;
},

next : function() {
          return false;
},

getDisplay : function() {
          return this.song.title + "  - " + this.song.artist;
},

isOwner : function() {
          return this.song.owner;
          //Determine paid user or not
},

getCurrentSongTitle : function() {
          return this.getDisplay() +  (this.song.owner ? '(보유곡)' : '');
},

getNextSongTitle : function() {
          return this.getDisplay();
},

toHTML : function() {
          return  SongHtml.toHTML(this.song, this.id, 'PlayingList.play');
} };
```

Figure 16. JavaScript of Cyworld music player

IV. Countermeasures

In this section, we propose countermeasures to protect copyrighted material from three vulnerabilities: cookie, message, and JavaScript alteration.

Encryption and HMAC are countermeasures for cookie and message alteration. With encryption, the message and cookie is sent in encrypted format, therefore it can protect the user privacy and server information. For example, title of a song user heard and source address of the song need to encrypt. On the other hand, with HMAC, it sends message and cookie in clear text, but HMAC is faster than encryption [14]. If a leakage of cookie and message make no odds, HMAC is useful.

We introduce some basic notations used throughout the paper.

TABLE III.
COUNTERMEASURES

Vulnerabilities	Countermeasures
Cookie Alternation	- encryption - HMAC
Message Alternation	- encryption - HMAC
JavaScript Alternation	- compiled program - anti-reverse engineering.

① K_{SM} : A secret key, shared by the server and music player.

② T_n : Time-stamp

③ m: cookie or message.

④ $E_{K_{SM}}(\square)$: symmetric encryption with key K_{SM}

⑤ $||$: concatenation

⑥ $M_{K_{SM}}(\square)$: HMAC with key K_{SM}

⑦ K_{SM}: shared secret key

The shared secret key K_{SM} is securely distributed and updated. When the K_{SM} is compromised, the server encrypt new share secret key K'_{SM} by music player's public key and send it. Music player decrypt it by its private key.

A. Encryption

We use a symmetric encryption scheme to protect the cookie/message. We assume that the server and music player have a secret key K_{SM} and the music player securely stores this key.

Before sending a cookie or message, the client's music player concatenates time-stamp T_n and m (cookie/message) to prevent a replay attack, and then encrypts it with a common symmetric key K_{SP} (see (1)). The music player then sends it. Upon receiving it, the server decrypts it. If the cookie/message is valid, and the time difference of T_n and the server time is within a threshold value, the server performs the next step. Otherwise, the server stops the process.

$$E_{K_{SP}}(m\,||\,T_n)\ . \qquad (1)$$

B. HMAC

We use a hash-based message authentication code (HMAC) to prevent cookie/message alternation attacks.

Before sending a cookie or message, the client's music player concatenates time-stamp T_n and m (cookie/message) to prevent a replay attack, and then calculates a HMAC, such as (2). The music player sends HMAC, T_n and m. Upon receiving this packet, the server calculates the HMAC with these values. If the cookie/message is valid, and the time difference of T_n and the server time is within a threshold value, the server performs the next step; otherwise, the server stops the process.

$$M_{K_{SP}}(m\,||\,T_n)\ . \qquad (2)$$

C. Compiled Program and Anti-reverse Engineering

Music players that are written in JavaScript are very dangerous, since an attacker can easily modify the JavaScript code. The countermeasure to resolve this problem is to provide the music player as a compiled program—the attacker cannot directly see the source code of a compiled program. However, an attacker can recover the source code by reverse engineering. Therefore, anti-reverse engineering, such as code obfuscation and anti-debugging, is applied to the program. If anti-reverse engineering is applied, the attack success rate will decrease and the effort required to crack the program will increase [10].

Ⅴ. CONCLUSION

The digital music market is increasing under the influence of reinforcement of the copyright law. However, digital music services have not been analyzed for security issues. If the services have vulnerabilities and adversaries exploit them, the attack damages the content owner's intellectual property. Therefore, we analyzed commercial online music streaming services, such as Cyworld, Dosirak, and Naver.

In this paper, we discovered three vulnerabilities, and proposed appropriate countermeasures to combat them.

ACKNOWLEDGMENTS

This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program, supervised by the NIPA (National IT Industry Promotion Agency). (NIPA-2011-(C1090-1131-0005)). This work supported by the Defense Acquisition Program Administration and Agency for Defense Development under the contract UD070054AD.

Corresponding author: Seungjoo Kim (skim71@korea.ac.kr).

REFERENCES

[1] Jinho Jung, "musicians are crying…," I-News, 2007.

[2] Cyworld Music, "http://music.cyworld.com/"

[3] Naver Music, "http://music.naver.com/"

[4] Dosirak, "http://www.dosirak.com/"

[5] Fiddler, "http://www.fiddlertool.com/fiddler/".

[6] Cooxie, "http://www.diodia.com/cooxietoolbar.htm/".

[7] Burp suit, "http://portswigger.net/"

[8] Wikipedia, "http://www.wikipedia.org"

[9] Mike Andrews, James A. Whittaker, “How to break web software,” Addison-Wesley Professional (February 12, 2006).

[10] Hye-Young Chang, Seong-Je Cho, “Implementation of an Obfuscator for Visual C++ Source Code,” The Journal of the KICS (Korean Institute of Communication Sciences), Vol. 35/No.2, pp. 59 - 67, 2008. 2

[11] Jan Cappaert, Bart Preneel, Bertrand Anckaert, Matias Madou, Koen De Bosschere, “Toward Tamper Resistant Code Encryption : Practice and Experience,” LNCS vol. 4991,pp. 86-100, 2008.

[12] D.W Jung, H.S Kim, JK Park, “A Code Block Cipher Method to Protect Application Programs From Reverse Engineering,” Journal of the Korea Institute of Information Security and Cryptology, vol. 18, no.2, 2008.

[13] J. Cappaert, N. Kisserli, D. Schellekens, B. Preneel, “Self-encrypting code to protect against analysis and tampering,” First Benelux workshop on Information and System security, 2006.

[14] Crypto++ 5.6.0 Benchmarks, http://www.cryptopp.com/benchmarks.html

Sangsik Lee received his B.S. (2010) in computer engineering from Sungkyunkwan University (SKKU) in Korea. Now, he is currently working at Risk Management Division of LOTTE Capital,. His research interests include network security, DRM, and information security.

Donghyun Choi received his B.E. degree in Electrical and Computer Engineering from Sungkyunkwan University, Korea, in 2005, M.S. degree in Computer Science from Sungkyunkwan University, Korea, in 2007, and Ph.D. degree in Mobile Systems Engineering from Sungkyunkwan University, Korea in 2010. His current research interests are in the areas of cryptography, SCADA, mobile security, and DRM.

Dongho Won received his B.E., M.E., and Ph.D. degrees from Sungkyunkwan University in 1976, 1978, and 1988, respectively. After working at the Electronics & Telecommunications Research Institute (ETRI) from 1978 to 1980, he joined Sungkyunkwan University in 1982, where he is currently Professor of the School of Information and Communication Engineering. His interests lie in cryptology and information security. In 2002, he was the president of KIISC (Korea Institute of Information Security & Cryptology).

Seungjoo Kim received his B.S. (1994), M.S. (1996), and Ph.D. (1999) in information engineering from Sungkyunkwan University (SKKU) in Korea. Prior to joining the faculty at Korea University (KU) in 2011, He served as Associate Professor of School of Information and Communication Engineering at SKKU for 7 years. Before that, He served as Director of the Cryptographic Technology Team and the (CC-based) IT Security Evaluation Team of the Korea Information Security Agency (KISA) for 5 years. Now He is Associate Professor of Center for Information Security Technologies (CIST) at KU. Also, He have served as an executive committee member of Korean E-Government, and advisory committee members of several public and private organizations such as National Intelligence Service of Korea, Digital Investigation Advisory Committee of Supreme Prosecutors' Office, Ministry of Justice, The Bank of Korea, ETRI(Electronic and Telecommunication Research Institute), and KISA(Korea Information Security Agency), etc. His research interests include cryptography, information security and information assurance. He is the corresponding author.