

Random Dynamic ID를 이용한 광고제거 방지 기법

한소정, 옹환승

이화여자대학교 컴퓨터학과

sjhan@ewhain.net, hsyong@ewha.ac.kr

Prevention Method from Removing Web Advertisement Using Random Dynamic ID

Sojeong Han, Hwan-Seung Yong

Dept. of Computer Science & Engineering, Ewha Womans University

요 약

오늘날 사람들은 점차 인터넷 사용에 익숙해져 가고 있으며, 소비자들로 하여금 공유와 참여를 적극적으로 유도하는 웹 2.0 시대가 본격화 되면서, 사이트에 묶여있던 콘텐츠를 브라우저상에서 마음대로 변형시킬 수 있는 그리스몽키가 등장하였다. 사용자들은 그리스몽키의 유저스크립트를 통해, 보기 싫은 광고를 제거하거나 보고자 하는 대로 변형시킴으로써 광고를 기반으로 사이트를 운영하는 기업이나, 사이트 운영자들은 사용자들에게 꼭 전달해야 하는 광고를 전달하지 못하는 경우가 발생하여 사이트를 운영 하는데 큰 손실을 줄 수가 있기 때문에, 광고제거 방지를 위한 연구가 필요하다. 본 논문에서는 이러한 그리스몽키의 광고제거 방지방법에 관하여 제안한다.

1. 서 론

오늘날 사람들은 점차 인터넷 사용에 익숙해져 가고 있으며, 인터넷을 통해 더 많은 정보를 생산하고 교환하고자 한다. 이를 통해 소비자들은 점차 적극적이고, 전문 지식을 갖춘 프로슈머(Prosumer, 생산자적 소비자)로 변화하고 있다. 정보 유통 방식 또한 사용자 편의성 극대화를 위해 획기적으로 개선되고, 다양화하는 소비자의 충족을 위해 상거래방식은 점차 정보 획득과 전달 중심으로 변화하고 있다. 소비자들로 하여금 공유와 참여를 적극적으로 유도하는 웹 2.0 시대가 본격화되면서 이러한 변화는 더욱 가속화 될 것이다.

여기서 '웹 2.0'[1]이란 기존의 클라이언트-서버 모델에 기반을 둔 정적인 웹, 그리고 주류미디어에 의해 장식되며 매스커뮤니케이션과 매스광고에 의해 지배되던 웹이 근본적으로 변화하고 진화한 차세대 웹을 뜻한다. 즉, 최종사용자에게 웹 애플리케이션을 제공하는 컴퓨팅 플랫폼이다.

그리스몽키(Greasemonkey)[2, 3]는 파이어폭의 확장기능으로, 사용자들은 CSS나 자바스크립트와 같은 단순한 도구를 이용하여 사이트를 사용자들이 재정의한

다양한 스크립트를 서로 공유함으로써 사이트를 자신의 마음에 드는 형태로 재정의해서 볼 수 있다.

사용자들은 그리스몽키의 유저 스크립트(user script)[3]를 통해, 보기 싫은 광고를 제거하거나 보고 싶은 대로 변형시킴으로써, 광고를 기반으로 사이트를 운영하는 기업이나, 사이트 운영자들은 사용자들에게 꼭 전달해야 하는 광고를 전달하지 못하는 경우가 발생하여 사이트를 운영 하는데 큰 손실을 줄 수가 있기 때문에, 광고제거 방지를 위한 연구가 필요하다.

본 논문에서는 이러한 그리스몽키의 광고제거 방지방법에 관하여 제안하고, 응용프로그램을 구현한다. 논문은 다음과 같이 구성되어 있다. 2장에서는 그리스몽키의 광고제거 유저스크립트들을 분석하고, 3장에서는 Random Dynamic ID라는 광고제거 방지 방법을 제안하고, 4장에서는 Random Dynamic ID로 변환해주는 응용 프로그램 구현, 5장에서는 실험 및 평가, 6장에서는 결론 및 추후 연구 발전 방향에 대하여 설명한다.

2. 기존 광고제거 유저스크립트의 분석

유저스크립트의 광고제거를 막기 위해, 우선 유저스크립트들을 제일 많이 등록되어있는 유저스크립트사이트(<http://userscripts.org/>)에 Ads라는 태그(Tag)로 등록된 스크립트 173개중 제거가 되지 않는 스크립트 35개를

*본 연구는 한국과학재단 목적기초연구(R01-2006-000-10609-0) 지원으로 수행되었음

제외하고 138개 스크립트들의 주요제거 방법을 분석 해 보았다.

분석한 결과 식별 자를 이용한 제거 방법은 109개, 특정 값을 이용한 제거 방법은 23개, 태그의 순서를 이용한 방법은 16개로 자세한 방법과 예는 다음과 같다.

2.1 식별자를 이용한 제거방법

광고제거 유저스크립트 중, 제일 많이 쓰인 방법으로, 유저스크립트를 작성하는 유저들이 웹 페이지코드를 분석하여, 제거하고 싶은 식별자인 ID나 Name을 찾아 스크립트에서 값을 제거하거나 CSS의 값을 스크립트에서 None 처리함으로써, 디스플레이 하지 않는 방법이다. [그림 1] ID를 이용한 광고제거 유저 스크립트 예를 보면, removeAds라는 함수에서 div id값이 'adbrand'인 것과, span id값이 'lc'인 것을 addGlobalStyle 함수에서 display:none 처리함으로써, 특정 ID들의 태그를 화면에서 보이지 않게 한다.

```
var Customize = {
  addGlobalStyle: function(css) {
    style = document.createElement("style");
    style.type = "text/css";
    style.innerHTML = css;

document.getElementsByTagName('head')[0].appendChild(style);

    removeAds: function() {
      this.addGlobalStyle('div[id="adbrand"
      { display: none ! important }');
      this.addGlobalStyle('span[id="lc"]
      { display: none ! important }');
    }
};
```

그림 1. ID를 이용한 광고제거 유저 스크립트의 예

2.2 특정 값을 이용한 제거방법

유저스크립트를 작성하는 유저들이 웹 페이지코드를 분석한 후, 어떠한 특정 주소나 특정 내용이 들어간 값을 찾아, 스크립트에서 None 처리함으로써, 디스플레이 하지 않는 방법이다

[그림 2] 내용을 이용한 유저스크립트 예를 보면, 페이지 코드 중 'Ads from Yahoo!'라는 내용이 있을 경우 그 내용의 부모노드를 display:none 처리함으로써, 특정 내용이 있는 태그를 화면에서 보이지 않게 한다.

```
var paragraphs
= document.getElementsByTagName("p");
for (var i = 0; i < paragraphs.length; i++) {
if(paragraphs[i].innerHTML=="Ads from Yahoo!")
{
paragraphs[i].parentNode.style.display = "none";
}
}
```

그림 2. 특정내용을 이용한 광고제거 유저 스크립트의 예

2.3 태그 순서를 이용한 제거방법

유저스크립트를 작성하는 유저들이 웹 페이지코드를 분석한 후, 테이블이나 아이프레임 등등의 순서를 파악한 후, 제거하고 싶은 태그의 순서를 이용하는 방법이다.

[그림 3] 태그 순서를 이용한 유저스크립트 예를 보면, 함수에서 첫 번째 IFRAME을 advert에 저장하고, advert의 노드를 제거함으로써, 화면에서 보이지 않게 한다.

```
function() {
  var advert =
  document.getElementsByTagName("IFRAME")[0];
  advert.parentNode.removeChild(advert);
};
```

그림 3. 태그의 순서를 이용한 광고제거 유저 스크립트 예

2.4 기존 광고제거 유저 스크립트 작성의 한계

대부분의 유저스크립트 작성자들은 웹 페이지코드를 분석하여 스크립트를 작성한다. 앞에서 보듯이 광고제거 유저 스크립트 역시 유저들이 웹 페이지 코드에서 제거하고 싶은 ID나 태그등을 분석한 후, 광고를 제거하는 방법을 가장 많이 사용하고 있다. 따라서 웹 페이지 코드를 분석하기가 어려워 지고, 기존의 광고제거 스크립트들에서 제거하는 ID를 예측할 수 없게 된다면, 코드 분석 후 ID나 태그를 제거하는 유저 스크립트를 작성할 수 없을 것이다.

이러한 유저 스크립트의 작성 한계를 이용한 Random Dynamic ID 기법을 3장에서 제안한다.

3. Random Dynamic ID

Random Dynamic ID란? 웹 페이지가 로드 될 때마다, 함수를 통해 생성된 랜덤 한 ID로써, 페이지가 로드 될 때마다 값이 바뀌고, 한 페이지 안의 동일한 ID값이 없다는 특성을 가지고 있다.

이 기법은 ASP[4]로 구현하였으며, [그림 4]처럼

MakeRndId라는 함수는 먼저 의미 없는 문자열 str을 생성하고, 그 문자열의 Length 범위 안에서 Random함수를 통해 난 수를 생성해준다. 다음과정으로 문자열의 난 수 번째 값을 저장하고, 그 값을 넘겨받은 길이만큼 반복하면서 축적 시킨다. 이렇게 만들어진 Random Dynamic ID값을 ASP 코드 안에서 출력하여 기존의 정적ID와 바꾸어 사용한다.

```
<% Private Function MakeRndId(IdLen)
    Dim str, rndld
    str="abcdefghijklmnopqrstuvwxyz012345
6789"
    Randomize
    Dim r, i
    For i = 1 to IdLen
        r = Int((36 * Rnd) + 1)
        rndld = rndld + Mid(str,r,1)
    Next
    MakeRndId = rndld
end function %>

<*** id="<%=MakeRndId(10)%>">
```

그림 4. Random Dynamic ID 함수와 사용 예

4. RDI(Random Dynamic ID) Converter 구현

3장에서 제안하는 Random Dynamic ID기법을 적용하려면, 각 페이지파일마다 Random Dynamic ID함수를 넣어주고, ID값을 Random Dynamic ID로 바꿔야 한다. 하지만 웹사이트들의 웹 페이지파일에 일일이 함수를 넣고, ID값을 변경하는 것은 무리일 것이다. 그러한 문제점을 해결하기 위해 RDI Converter를 C#[5]으로 구현하였다.

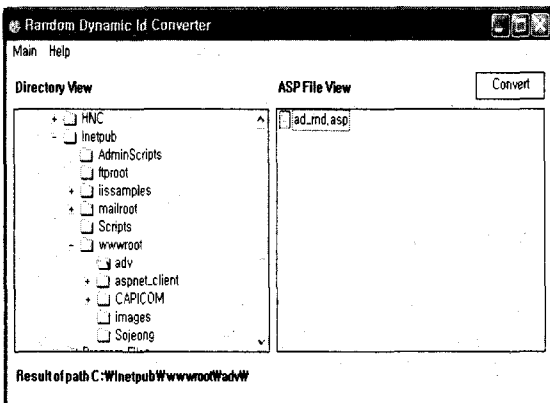


그림 5. RDI Converter 실행화면

RDI Converter는 사용자가 선택한 폴더 안의 페이지파일들을 읽어 문자열형태로 변환한 후, 사용자의 편의를 위해 동일한 백업파일을 만들고, ID값을 찾아 Random Dynamic ID로 변환하고, Random Dynamic ID 함수를 추가한 후, 그 문자열을 기존 파일의 이름으로 저장한다.

[그림 5]와 같이 Windows에서 기본으로 제공하는 윈도우 탐색기 형태로 구현하여, 사용자들이 보다 쉽게 프로그램을 사용할 수 있도록 하였다. 사용방법은 Directory View로 변환하고자 하는 폴더를 찾고, ASP 파일의 유무를 확인한 후, Convert 버튼을 누른다.

5. 실험 및 평가

5.1 Random Dynamic ID을 이용한 광고 제거 방지

3장 Random Dynamic ID에서 제안한 방법으로 정확한 실험을 위해 Naver의 Code로 부분 적용해 보았다.

[그림 6]을 보면, 일반 정적 ID인 span id값이 변하지 않아 추측이 가능하지만, Random Dynamic ID를 적용한 div id 값을 보면 추측하기 어려운 랜덤한 문자열이 ID값으로 적용된 것을 볼 수 있고, [그림 7]의 결과처럼 정적 ID는 광고 제거 스크립트 적용 후 제거 되었지만, Random Dynamic ID는 제거되지 않은 것을 볼 수 있다.

```
<span id="ad1" style=position: absolute;
left:75%:>
<div id="z5dfctb5r0">
<iframe src="http://203.255.177.212/ad2.html"
width="204" height="100" border="0"
frameborder=0 scrolling=no marginheight=0
marginwidth=0 align=center>
</iframe>
```

그림 6. Random Dynamic ID 적용 후 코드

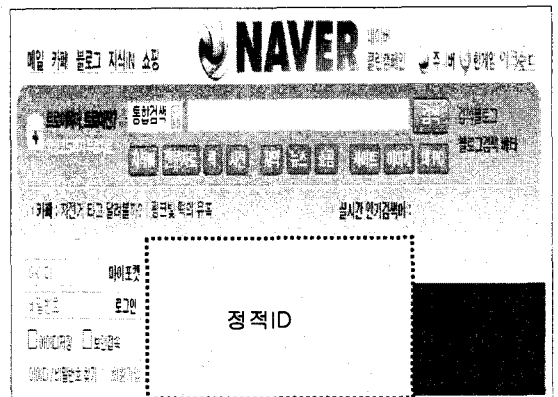


그림 7. 광고제거 유저스크립트 적용 후

5.2 RDI Converter를 사용하여 자동 RDI 변환

RDI Converter를 사용하여 코드변환이 완료되면, 기존 웹 페이지들의 동일한 백업파일이 생성되고, [그림 9]의 결과처럼 기존의 [그림 8]과 같은 정적ID를 가진 웹 페이지의 코드가 Random Dynamic ID로 변환된 것을 볼 수 있고, [그림 10]처럼 정적ID를 가진 광고는 제거되었지만, [그림 11] 결과처럼 RDI Converter를 이용하여도 Random Dynamic ID를 적용한 것처럼 광고가 제거되지 않는 것을 확인 할 수 있다.

```

.....생략
<span id="ad1" style=position:
absolute: left:75%:>
<div id="ad2">
<iframe src="http://203.255.177.212/ad2.html"
width="204" height="100" border="0"
frameborder=0 scrolling=no marginheight=0
marginwidth=0 align=center>
</iframe>
</div>
</span>
    
```

그림 8. RDI Converter를 사용하여 코드변환 전 소스코드

```

<% Private Function MakeRndId(IdxLen)
Dim str, rndId
str="abcdefghijklmnopqrstuvwxyz012345
6789"
Randomize
Dim r, i
For i = 1 to IdxLen
r = Int((36 * Rnd) + 1)
rndId = rndId + Mid(str,r,1)
Next
MakeRndId = rndId
end function %>
.....생략
<span id="<%=MakeRndId(10)%>"
style=position:
absolute: left:75%:>
<div id="<%=MakeRndId(10)%>">
<iframe src="http://203.255.177.212/ad2.html"
width="204" height="100" border="0"
frameborder=0 scrolling=no marginheight=0
marginwidth=0 align=center>
</iframe>
</div>
</span>
    
```

그림 9. RDI Converter를 사용하여 코드 변환된 후 소스코드

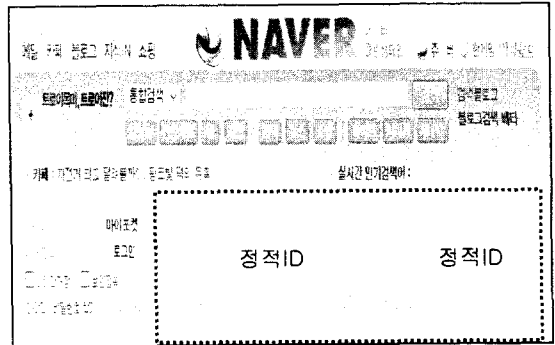


그림 10. RDI Converter를 사용하여 코드변환 전 광고 제거 유저스크립트 적용

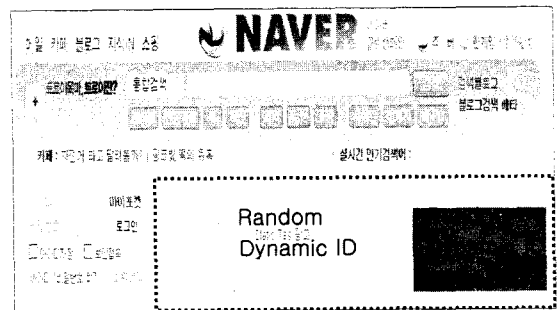


그림 11. RDI Converter를 사용하여 코드 변환된 후 광고 제거 유저스크립트 적용

6. 결론 및 추후 연구

그리스용키의 광고제거 스크립트를 막기 위해 본 논문에서는 방법과 응용 프로그램을 구현하였다. 기존의 광고 제거 유저 스크립트 중 약 79%의 방법을, 본 논문에서 광고 제거 방식을 구현하였기 때문에, 모든 스크립트들의 광고 제거 방지 할 수는 없지만, 많은 광고제거 유저스크립트를 방지 할 수 있을 것이다. 그리고 본 연구에서 구현한 RDI Converter프로그램의 사용으로 쉽고 빠르게, 스크립트의 광고를 제거를 방지할 수 있다. 하지만 태그자체를 제거하는 스크립트를 방지할 수는 없다는 것이 문제점이다. 앞으로 그러한 점을 보완하는 태그자체의 제거방지를 위한 연구가 필요할 것이다.

7. 참고 문헌

- [1] IT전략지원단 IT신기술사업팀, 플랫폼으로서의 웹, 웹2.0이란 무엇인가?, 한국통신전산원, 2005
- [2] 김태우, 태우's Completely Incomplete Guide to Web 2.0, 크리에이티브 커먼즈라이선스, 2005
- [3] Mark pilgrim, Greasemonkey Hack, O'Reilly, 2005
- [4] 김태영, Taeyo's ASP, 삼양출판사, 1999
- [5] Karli Watson, Beginning C#, 정보문화사, 2002