AdCube: WebVR Ad Fraud and Practical Confinement of Third-Party Ads

Hyunjoo Lee, Jiyeon Lee, Daejun Kim, Suman Jana, Insik Shin, Sooel Son

KAIST

Columbia University
In the City of New York
Virtual Reality (VR)

- VR is the next computing revolution, it changes how we play, work, learn and live.

Image Source:
https://etech.iec.ch/issue/2019-03/virtual-reality-makes-real-life-even-better
https://www.wareable.com/vr/vr-fitness-holodia-one-big-problem
Limitations of the VR ecosystem

Takes time to make it work on multiple headsets

Developer

Implement

App_A1
AppStore for Headset A

Headset A

App_B1
App_B2
AppStore for Headset B

Headset B

App_C1
AppStore for Headset C

Headset C

Needs to have a right headset

User
Enable VR on the Web

Easy to share

Developer

Implement

https://webvr.com

Access

User

Any device through any browser

Good!

Headset A

Headset B

Headset C
WebVR

• Enables VR on the Web, Supported by: Firefox 77+  Chrome 79+  Edge 79+

• Several frameworks (e.g., A-Frame) exist to help build a 3D world.

A-Frame Example

```
<head>
<script src="aframe.js"></script>
...
<a-scene>
<a-box position="-1 0.5 -3" color="#4CC3D9"></a-box>
<a-sphere position="0 1.25 -5" color="#EF2D5E"></a-sphere>
<a-cylinder position="1 0.75 -3" color="#FFC65D"></a-cylinder>
<a-plane position="0 0 -4" color="#F7BC8A4"></a-plane>
<a-sky color="#ECECEC"></a-sky>
</a-scene>
...
```

• Use cases: News, e-commerce, VR films, education, Custom business solutions
Motivation

• Online advertising is essential for the benefit of Web hosts.
• Recently, advertising has been applied to 3D world.

500% increase in the click-through rate due to the VR ad campaigns
Problem

• There is no iframe-like primitive to isolate third-party ads in WebVR

⇒ Abusive third-party ads share the canvas with the first-party webpage
We introduce **four ad fraud techniques** specific to the WebVR environment
WebVR ad Frauds

Blind Spot Tracking Attack

*Hide an ad entity in the opposite direction*

Abuse Auxiliary Display Attack

*Display an Ad entity on the auxiliary display*

Gaze Cursor Jacking Attack

*Create a fake gaze cursor and hide the original cursor*

Controller Jacking Attack

*Introduce an additional fake controller cursor*
WebVR ad Frauds

Blind Spot Tracking Attack
Hide an Ad entity in the opposite direction

Abuse Auxiliary Display Attack
*Display an ad entity on the auxiliary display*

Gaze Cursor Jacking Attack
Create a fake gaze cursor and hide the original cursor

Controller Jacking Attack
Introduce an additional fake controller cursor
### WebVR ad Frauds

<table>
<thead>
<tr>
<th>Blind Spot Tracking Attack</th>
<th>Abuse Auxiliary Display Attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hide an Ad entity in the opposite direction</td>
<td>Display an Ad entity on the auxiliary display</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Gaze Cursor Jacking Attack</th>
<th>Controller Jacking Attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a fake gaze cursor and hide the original cursor</td>
<td>Introduce an additional fake controller cursor</td>
</tr>
</tbody>
</table>
User Study on 82 Participants

Blind Spot Tracking Attack
Hide an ad entity in the opposite direction

Current line of sight

Abuse Auxiliary Display Attack
Display an ad entity on the auxiliary display

Gaze Cursor Jacking Attack
Create a fake gaze cursor and hide the original cursor

I am pointing...

Controller Jacking Attack
Introduce an additional fake controller cursor
User Study Results

Blind Spot Tracking Attack
# of Participants: 32
Success Rate: 94.12%

Abuse Auxiliary Display Attack
# of Participants: 32
Success Rate: 100%

Gaze Cursor Jacking Attack
# of Participants: 17
Success Rate: 88.23%

Controller Jacking Attack
# of Participants: 18
Success Rate: 93.75%
User Study Results

<table>
<thead>
<tr>
<th>Attack Type</th>
<th># of Participants</th>
<th>Success Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blind Spot Tracking Attack</td>
<td>32</td>
<td>88.23%</td>
</tr>
<tr>
<td>Abuse Auxiliary Display Attack</td>
<td>32</td>
<td>100%</td>
</tr>
<tr>
<td>Gaze Cursor Jacking Attack</td>
<td>17</td>
<td>88.23%</td>
</tr>
<tr>
<td>Controller Jacking Attack</td>
<td>18</td>
<td>93.75%</td>
</tr>
</tbody>
</table>

The four techniques are effective in conducting click and impression fraud in WebVR.
Defense Requirements

1. Third-party JavaScript code should place ad entities only within the confined area.

2. Third-party JavaScript code should not be able to alter DOM elements and sensitive entities (e.g., camera and controller).
AdCube Overview

- AdCube is a JavaScript library, designed to confine the execution of third-party scripts rendering WebVR ads.

Implement advertisements using AdCube APIs

Include an AdCube code with a security policy
AdCube in Detail

1. The publisher defines ad in scene and writes a security policy.

```html
1: <body>
2:  <script src='adcube.js'></script>
3:  <a-scene>
4:    <!-- part of the host app -->
5:    <a-box can-read></a-box>
6:    <a-cylinder can-write></a-cylinder>
7:    ...
8:    <!-- a new definition for ad -->
9:    <a-adcube position='0 0 -2' width='2' height='2' depth='2'></a-adcube>
10:  </a-scene>
11:  </script>
12: </body>
```

An example of A-Frame host page with AdCube
2. The Ad provider implements advertising with AdCube APIs.

An example of ad-serving JS script

```javascript
1: let e = createElement('a-gltf-model');
2: e.setAttribute('src', 'product.gltf');
3: e.addEventListener('click', onClick);
4: addElement('adcube-id', e);
5: function onClick(event){
6:   /** click event handler **/
7: }
```
AdCube in Detail

3. AdCube confines an ad in the space provided by the publisher.

Resizing the ad entity to fit within the space allowed by the publisher
3. AdCube confines an ad in the space provided by the publisher.

Calculating the maximum size of a bounding box including animation actions
Security Evaluation

• AdCube blocks all four of the attacks by:

✓ The default policy of AdCube specifies no read and write access

- Blind Spot Tracking Attack
- Abuse Auxiliary Display Attack
- Gaze Cursor Jacking Attack
- Controller Jacking Attack
Security Evaluation

**AdCube effectively defends all attack scenarios**

- The default policy of AdCube specifies no read and write access
- AdCube prohibits advertising behind the camera
- All fake cursors are visually distinguishable with the [AD] label

**Blind Spot Tracking Attack**

**Abuse Auxiliary Display Attack**

**Gaze Cursor Jacking Attack**

**Controller Jacking Attack**
Performance Evaluation

• Two other methods
  - **Baseline**: run third-party scripts without any underlying security defense
  - **Mirroring**: run the third-party script in a separate origin different from its host

• Experiment
  - Measured 1) the average page loading times ran on the nine WebVR sites and 2) FPS change for 12 events

<table>
<thead>
<tr>
<th>Performance Metric</th>
<th>Baseline</th>
<th>Mirroring</th>
<th>AdCube</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Loading Time (s)</td>
<td>0.55</td>
<td>0.95</td>
<td>0.78</td>
</tr>
<tr>
<td>FPS (drop rate)</td>
<td>56.70 (-)</td>
<td>53.12 (6.32%)</td>
<td>55.79 (1.60%)</td>
</tr>
</tbody>
</table>
Conclusion

• We have devised four new attack variants to conduct WebVR ad fraud.
• We propose AdCube that allows publishers to confine third-party ad entities.
• AdCube is able to block ad fraud threats with negligible overheads.

A Showcase of WebVR ads with AdCube
Thank You

Jiyeon Lee (jy.lee@kaist.ac.kr)