WebGraph: Capturing Advertising and Tracking Information Flows for Robust Blocking

Sandra Siby, Umar Iqbal, Steven Englehardt, Zubair Shafiq, Carmela Troncoso

USENIX Security, 11 Aug 2022
Ad- and tracker-blocking solutions
Many blocking solutions rely on manually-curated filter lists

https://easylist.to/easylist/easylist.txt
Many blocking solutions rely on manually-curated filter lists

Issues?

- Scalability of updates
- Robustness
Many blocking solutions rely on manually-curated filter lists

Issues?

Scalability of updates

Robustness

Solution? Build automated detection systems?
Towards automated detection

Network layer
- URL features
- HTTP header/payload features

JavaScript layer
- Static and dynamic code analysis
Towards automated detection

**Network layer**
- URL features
- HTTP header/payload features

**JavaScript layer**
- Static and dynamic code analysis

**Cross-layer**
Combine interactions across network, JavaScript, HTML layers
Cross-layer approach: AdGraph

Iqbal et al. AdGraph: A Machine Learning Approach to Automatic and Effective Adblocking
Graph Representation — normal events
Graph Representation – tracker events

https://cm.g.doubleclick.net/pixel?google_nid=openx&google_cm=&google_sc=&google_tc=

https://us-u.openx.net/w/1.0?sd?id=537072991&val=CAESELem3bq0sLYNy11&google_cver=1

https://cm.g.doubleclick.net/pixel?google_nid=openx&google_cm&google_sc

https://www.nytimes.com/
Are current detectors the panacea?

AdGraph and related detectors improve accuracy.

What about robustness to adversarial evasion?
Adversarial evasion: Threat model
Adversarial evasion: Threat model

- Visited site (first party): example.com
- Image from img.example.com (first party)
- Ad from tracker.com (third party)
- Benign
- Ad/tracker
Adversarial evasion: Threat model

Goal:
Get ads(trackers) classified as benign content
Cross-layer approach: AdGraph

Features

Content
- Related to a node in the graph

Structure
- Related to the interactions between nodes
Cross-layer approach: AdGraph

- **Features**
  - Content
    - Related to a node in the graph
  - Structure
    - Related to the interactions between nodes

**Easy to manipulate!**
Question 1:

How much do graph-based detection systems rely on content features?
Content mutation

Check classifier predictions on unchanged and content mutated data.

https://ai.blockchain.com?iot=big_data

Domain name randomization

Query string randomization

Collusion models

No collusion

Limited collusion with first party
Results (collusion with first party)

Average success rate per webpage: ~96%
~32% of successful mutations happen to ad exchange trackers
Why?

<table>
<thead>
<tr>
<th>Length of URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL is a subdomain of the first party</td>
</tr>
<tr>
<td>URL is first or third party</td>
</tr>
</tbody>
</table>

Top-3 features are content based.

Third-partiness of the URL is important.
Why?

<table>
<thead>
<tr>
<th>Length of URL</th>
<th>Top-3 features are content based.</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL is a subdomain of the first party</td>
<td>Third-partiness of the URL is important.</td>
</tr>
<tr>
<td>URL is first or third party</td>
<td></td>
</tr>
</tbody>
</table>

AdGraph and related ML-based approaches rely on brittle content features.
Performance (Accuracy)

Original AdGraph (Content + Structure): ~92%
Performance (Accuracy)

Original AdGraph (Content + Structure): ~92%

Removal of content features drops accuracy to ~80%
Question 2:

Can we improve performance of graph-based detection systems without using content features?
Current detection systems lack ad/tracker behaviour information
Current detection systems lack ad/tracker behaviour information
Current detection systems lack ad/tracker behaviour information.

We create **WebGraph**.

**Idea:** Capture fundamental tracking behaviour.
Current detection systems lack ad/tracker behaviour information.

We create **WebGraph**.

**Idea:** Capture fundamental tracking behaviour.

**How?**
- Augment graph structure
- Introduce *flow* features
AdGraph Representation

10
Iframe request

10
HTML iframe

11
Script node

15
XMLHTTP request

2
Script request

2
Script node

5
Image request

6
HTML image
WebGraph Enhancements

- Iframe request (10)
- HTML iframe (10)
- Script node (11)
- Read Cookie storage (5,13)
- Shared cookie in request (5)
- Redirect initiation to load the image (5)
- Network request setting a cookie (5)
- XMLHTTP request (15)
- HTML image (6)
- Cookie storage (5)

- Script request (2)
- Script node (2)
- Image request (5)
- Image request (5)
WebGraph: Features

Features

Content
Structure
Flow
Performance (Accuracy)

Original AdGraph (Content + Structure): ~92%

Removal of content features drops accuracy to ~80%

WebGraph (Structure + Flow) raises it to ~87%
Performance (Accuracy)

Original AdGraph (Content + Structure): ~92%

Removal of content features drops accuracy to ~80%

WebGraph (Structure + Flow) raises it to ~87%

Content mutation success drops from 96% to 8% against WebGraph
Question 3:

Is WebGraph robust to evasion by an adversary targeting WebGraph’s features?
Targeting structural and flow features

example.com

First party
Adversary
Another third party

Classified as benign
Classified as ad/tracker
Graph mutation by adversary

Ex. Add child node by sending an HTTP request
Local and global structural changes cause classification switches to adversarial and non-adversarial nodes.

**Graph mutation by adversary**

![Diagram showing graph mutation by adversary]
Misclassifications caused by mutation
Misclassifications caused by mutation
Misclassifications caused by mutation
Misclassifications caused by mutation

Such attacks work only if the adversary can:

1. Co-ordinate with different entities
2. Tolerate side-effects
Current graph-based tracker detection systems rely heavily on content features that can be susceptible to adversarial evasion.

WebGraph is a robust detection system based on capturing fundamental patterns of ad/tracker behaviour.

We raise the bar for ads/trackers attempting evasion.

Paper: [https://www.usenix.org/conference/usenixsecurity22/presentation/siby](https://www.usenix.org/conference/usenixsecurity22/presentation/siby)

Code: [https://github.com/spring-epfl/WebGraph](https://github.com/spring-epfl/WebGraph)

Contact: sandra.siby@epfl.ch  @sansib