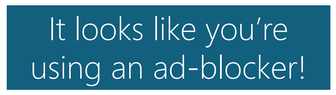
ShadowBlock: A Lightweight and Stealthy Adblocking Browser

Shitong Zhu*, Umar Iqbal†, Zhongjie Wang*, Zhiyun Qian*, Zubair Shafiq†, Weiteng Chen*
University of California, Riverside*, The University of Iowa†

Overview

- More than 600 million devices globally use adblockers as of December 2016
- The rise of adblocking has jeopardized the ad-powered business model and publishers have deployed anti-adblocking paywalls



- We propose ShadowBlock, a Chromium-based adblocking browser that bypasses anti-adblocking paywalls
- ShadowBlock bypasses anti-adblocking paywalls with **100%** success rate and performs as well as state-of-the-art adblockers in terms of ads coverage and page loading speed

Shadow Copy

- How do anti-adblockers detect the use of adblockers?
  - Blocking ads introduces **different states** that are **observable** to JavaScript runtime

```
// Example anti-adblocking code
var adblock_state =
document.getElementById('some_ad');
window.setTimeout(function() {
   if (adblock_state === undefined)
       show_paywall();
}, some_timeout);
```

- The key of hiding adblockers is **masking the difference**

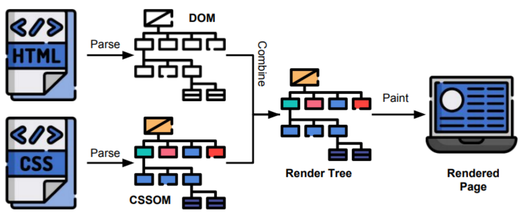
```
// What difference to mask?
var adblock_state =
document.getElementById('some_ad');
```

JavaScript API — Ad DOM element

- We must mask the state returned to `getElementById()` for DOM element "`some_ad`" as if it is still intact, even though it has been hidden by us

Hiding Mechanism

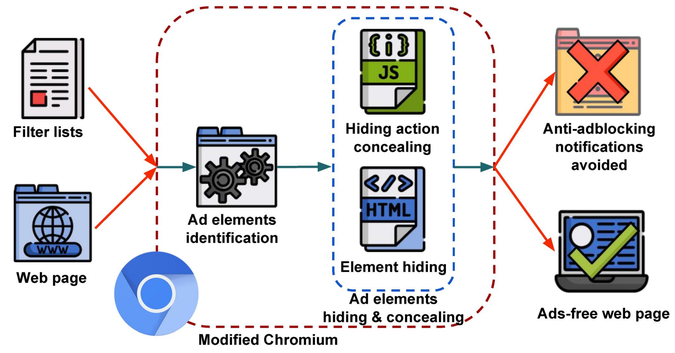
- **DOM/CSS Layer**: parse flat HTML and CSS in plain-text
- **Render Tree Layer**: combined from DOM and CSSOM
- **Paint Layer**: generating rendered pixels to user's viewpoint according to Render Tree



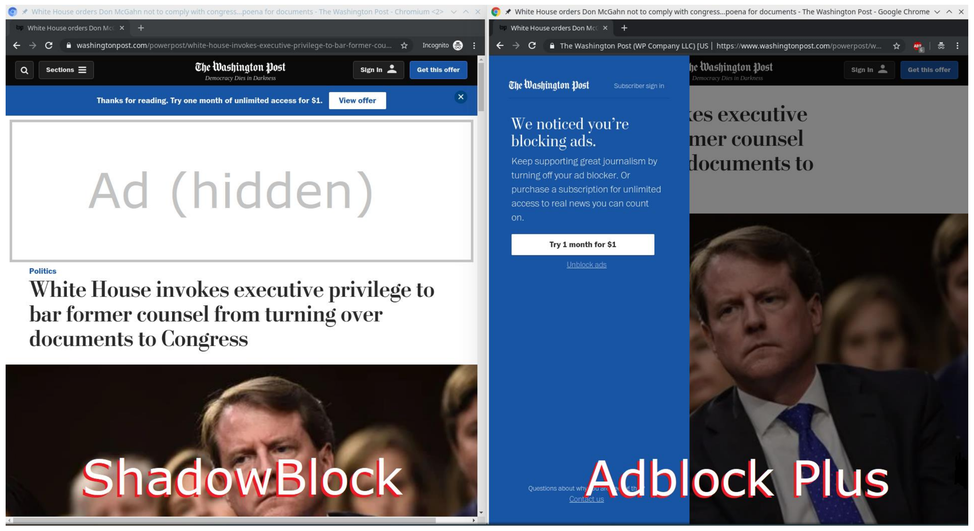
- We choose to toggle CSS property **`visibility: visible`** as our ad element hiding mechanism
  - Low-level enough so there are minimum number of channels leaking the action to hook
  - High-level enough to avoid complex object translation

ShadowBlock

- Ads Identification
  - Statically created *ads* are detected by monitoring attribute change events
  - Dynamically (JavaScript) created *ads* are detected by monitoring elements created with ad scripts
- Ads Hiding
  - ShadowBlock hides the traces of adblocking in a stealthy manner by masking different states caused by toggling **`visibility`** property
  - All JavaScript APIs that can be used by anti-adblockers to probe the actual states of ad elements are hooked to present a fake state as if ads are still intact



Demo

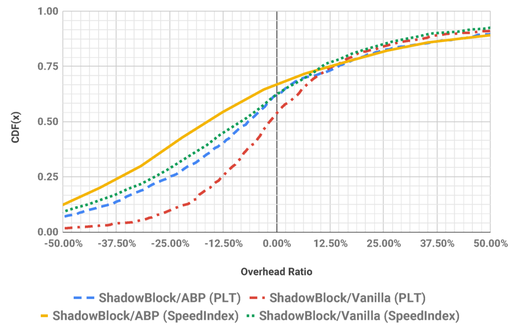


Results & Evaluation

- **100%** success rate against anti-adblockers whereas dedicated filter lists have only **29%** success rate
- 97.7% accuracy, with 98.2% recall and 99.5% precision in blocking ads on Alex top-1K websites
- Speeds up page loads by 5.96% in terms of median Page Load Time (PLT) and 6.37% in terms of median SpeedIndex on Alexa top-1K websites

Tool	Notification	Ad Switching	Crypto-mining
Total	201	5	1
ShadowBlock	201 (100%)	5 (100%)	1 (100%)
Filter lists	59 (29%)	1 (20%)	0 (0%)

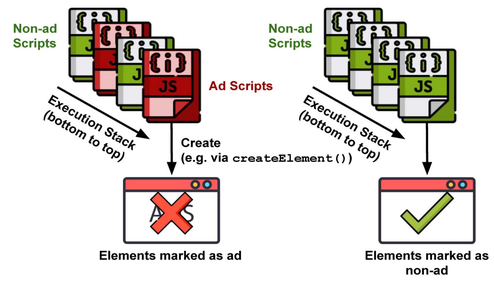
Event	TP	FN	TN	FP
Count	926 (98.2%)	17 (2.8%)	938 (99.5%)	5 (0.5%)



Execution Projection

- Dynamically created ad elements can be identified by tracking execution stack
  - Determining the ad-ness by asserting whether there is any ad script on stack at DOM events
  - Feasible due to single-threaded JavaScript execution

```
// Typical dynamically created ad
var ad_img = document.createElement("img");
ad_img.src = "https://advertiser.com/ad.jpg";
document.body.appendChild(ad_img);
```



Chromium Instrumentation

- Low level instrumentation makes ShadowBlock stealthy and efficient
- We instrument two major components in Chromium: Blink and V8
  - Blink is responsible for constructing the rendering tree
  - Bindings module handles interaction between V8 and Blink
- Hooking for ad identification
  - Capture element creation and modification
  - Capture JavaScript execution stack
- Hooking for concealing actions
  - CSS/Style related – `getComputedStyle()`
  - Event Related – `onfocus`
  - Hit testing related – `elementFromPoint()`
- Keep track of ad related scripts in execution stack and their activity (*execution projection*) and element modifications for identification of ad elements

Key Contributions

- Design and implement a stealthy adblocking browser
- Evade **100%** of anti-adblockers and replicate EasyList with 98.3% accuracy with less than 0.6% breakage
- We find that ShadowBlock loads pages as fast as stock Chromium running Adblock Plus
- We open source our implementation to allow reproducibility as well as help future extensions by the research community (**`https://github.com/seclab-ucr/ShadowBlock`**)

More details in our WWW'19 paper:
ShadowBlock: A Lightweight and Stealthy Adblocking Browser
Shitong Zhu, Umar Iqbal, Zhongjie Wang, Zhiyun Qian, Zubair Shafiq, and Weiteng Chen
The Web Conference (WWW) 2019

www.shitong.me @zst_rising88
www.umariqbal.com @umaarr6