

ShadowBlock: A Lightweight and Stealthy Adblocking Browser

Umar Iqbal
The University of Iowa

Overview

- More than 600 million devices globally use adblockers as of December 2016
 - Reasons contribute to the popularity of adblocking: (1) Flashy Ads, (2) Tracking ecosystem, and (3) Performance improvements
 - The rise of adblocking has jeopardized the ad-powered business model and publishers have deployed anti-adblocking paywalls
- It looks like you're using an ad-blocker!
- We propose ShadowBlock, a Chromium-based adblocking browser that bypasses anti-adblocking paywalls
 - ShadowBlock bypasses anti-adblocking paywalls with 100% success rate and outperforms state-of-the-art adblockers

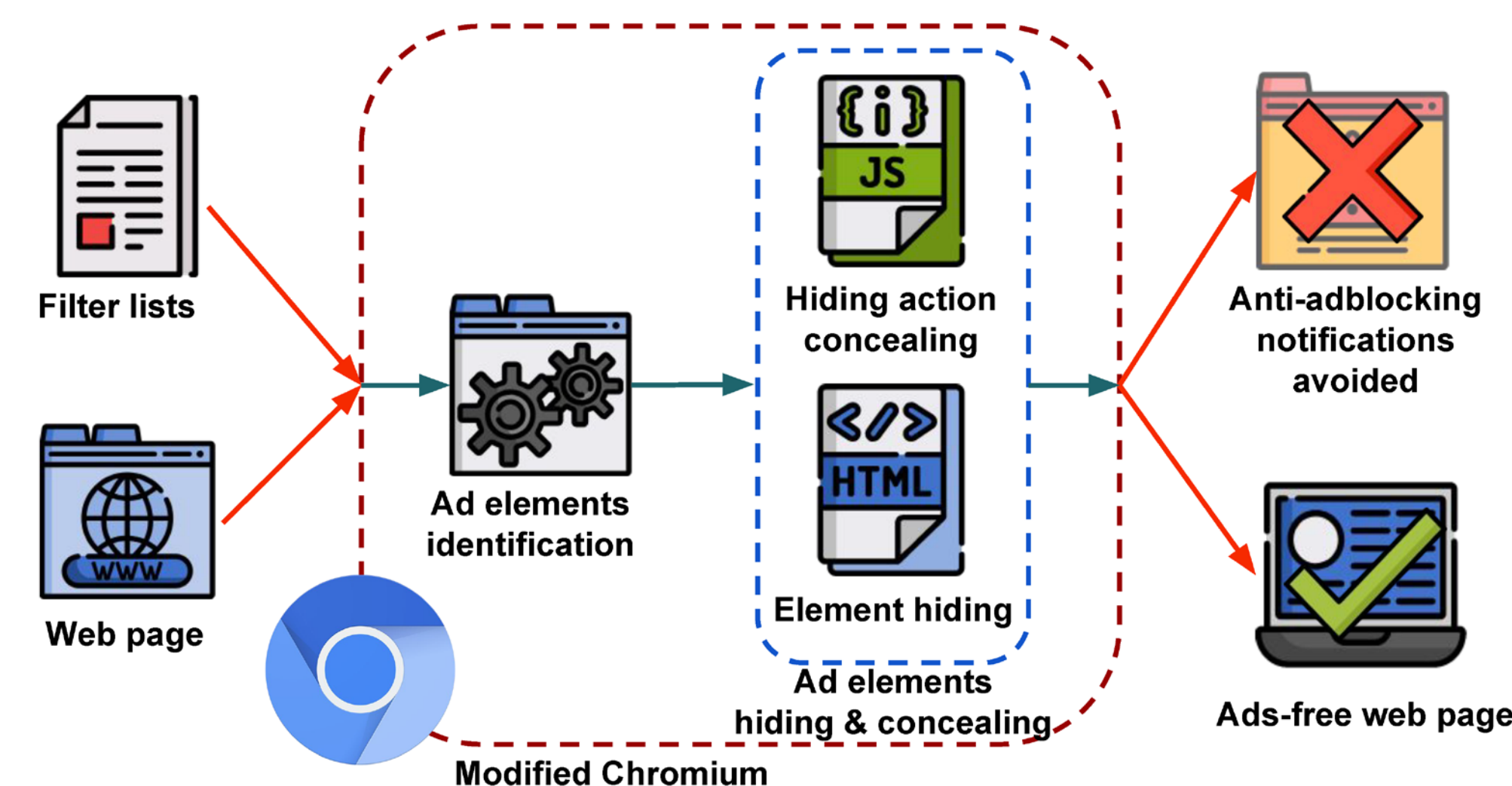
Chromium Instrumentation

- ShadowBlock is implemented as a modification to Chromium
- Low level instrumentation makes ShadowBlock stealthy and efficient
- Chromium has two major components: Blink and V8
- We instrument Blink's Rendering and Bindings module
 - Rendering module is responsible for constructing the rendering tree
 - Bindings module handles interaction between V8 and Blink
- Hooking for 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

ShadowBlock

- Adblockers leave traces that are detectable by anti-adblocking scripts
- ShadowBlock hides ads without leaving traces
- 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

```
// Typical anti-adblocking script
DetectAdblocker() {
  var ad = document.getElementById('ad');
  if (ad.offsetHeight < 33 || ad.clientHeight < 33)
    return true;
  else
    return false;
}
```

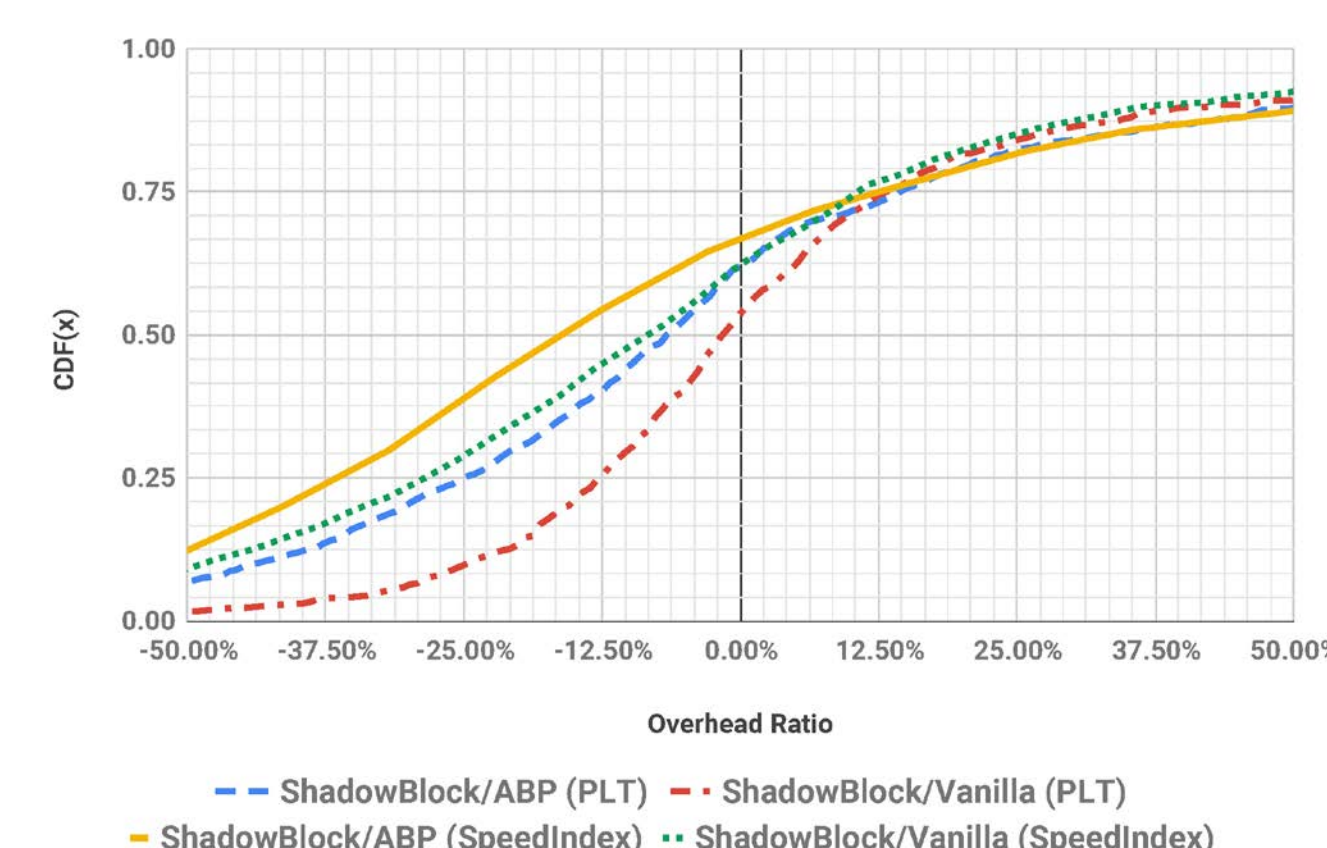


Results & Evaluation

- 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%)

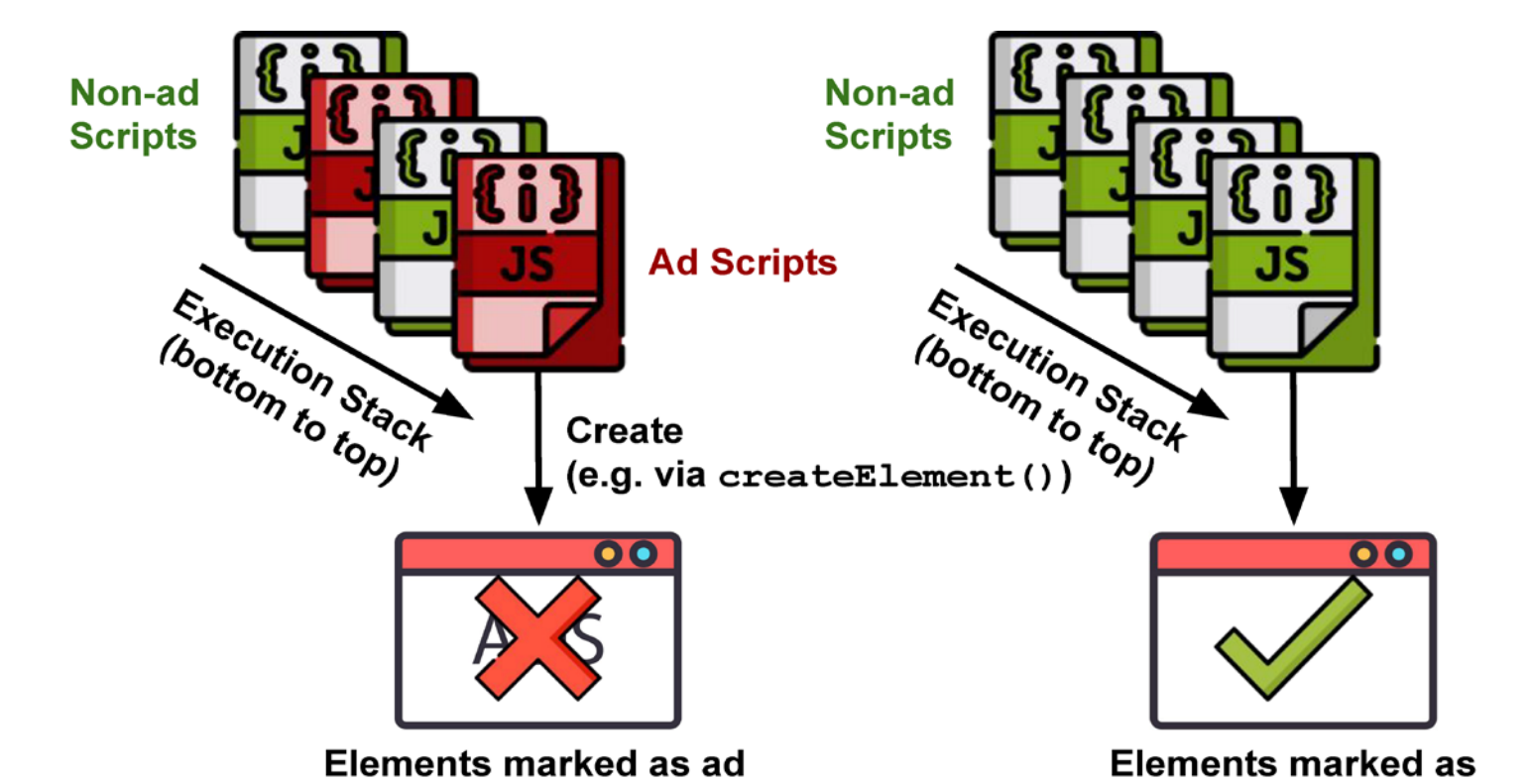
Event	TP	FN	TN	FP
Count	926 (98.2%)	17 (2.8%)	938 (99.5%)	5 (0.5%)



Execution Projection

- Dynamically (JavaScript) created ads identification
- Attribute parser's actions to scripts in the stack

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



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 faster than Adblock Plus and stock Chromium
- 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.umariqbal.com
[@umaarr6](https://twitter.com/umaarr6)