Knockin’ on Trackers’ Door:

Large-Scale Automatic Analysis of Web Tracking

Iskander Sanchez-Rola, Igor Santos

Web Tracking

It is a common practice to gather **user browsing data**.

www.cdrokc.com
CDR
ELECTRONICS
LAPTOPS & CELL PHONES SUPER STORE
VISIT OUR LIKE US PAGE BELOW TO FIND OUT HOW YOU CAN ENTER AND WIN!
$100 CASH!
ENTER BELOW FOR A CHANCE TO WIN!
Share the love and tell a friend!
Thank you for your continued support and patronage!

No need to meet in a gas station parking lot.
We've got a store!
TWISTER WIRELESS
405-429-7914
Great prices and friendly service.
You'll be glad you stopped by!

Web Tracking

Recent studies provided a better understanding of a **particular subset** of web tracking techniques but they were not devoted to fully understand and to generically discover web tracking script.

Web Tracking

Recent studies provided a better understanding of a **particular subset** of web tracking techniques but they were not devoted to fully understand and to generically discover web tracking script.

Existing solutions are based on:

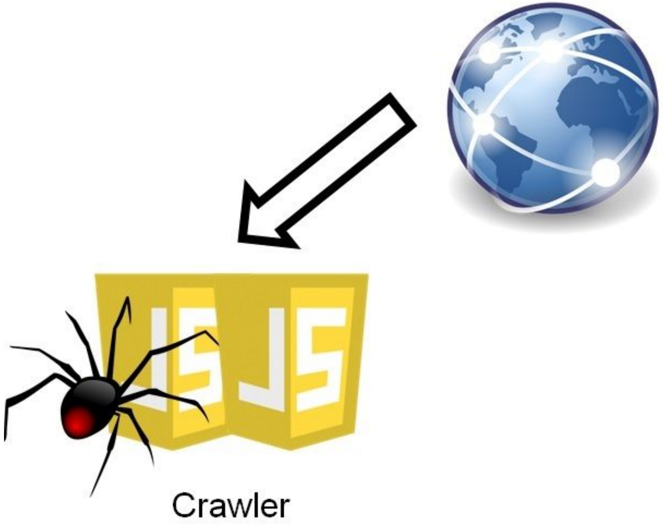
Blacklists
Static rules

Web Tracking

Due to the limitations of current solutions, we build our own tracking analysis tool called **TrackingInspector**, and we present the **first large-scale** analysis of generic web tracking scripts.

We can automatically detect known tracking script **variations** and also identify likely **unknown** tracking script candidates.

TrackingInspector



Crawler

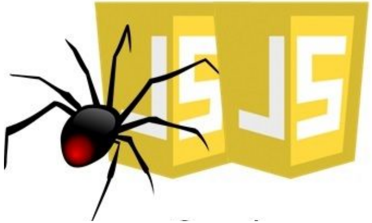
Crawler

Implementation based on PhantomJS

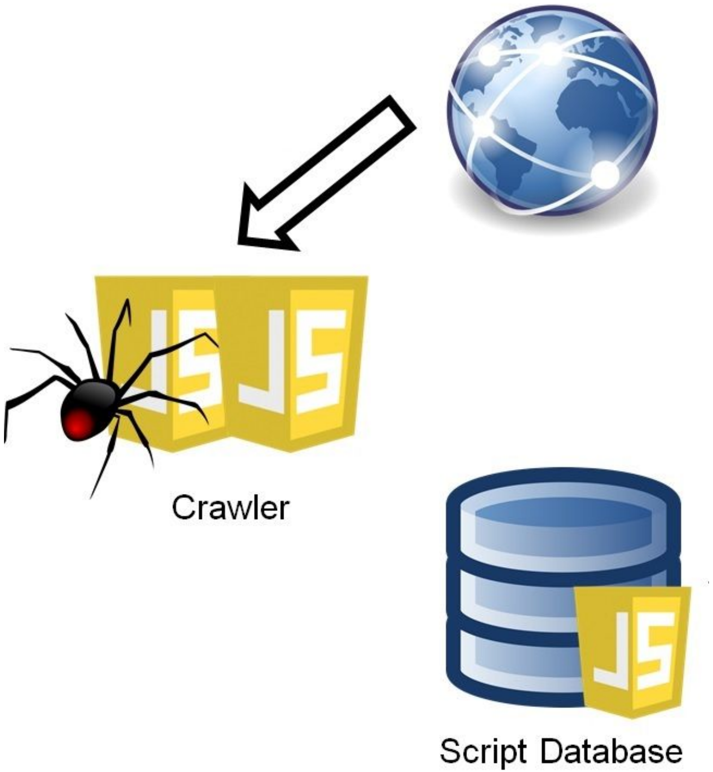
- Modified to **hide** its automatic nature from sites
- Can deal with script **obfuscation** (based on JSBeautifier)

Data Retrieved

- JavaScript files loaded
- HTML-embedded scripts



TrackingInspector



Script Database

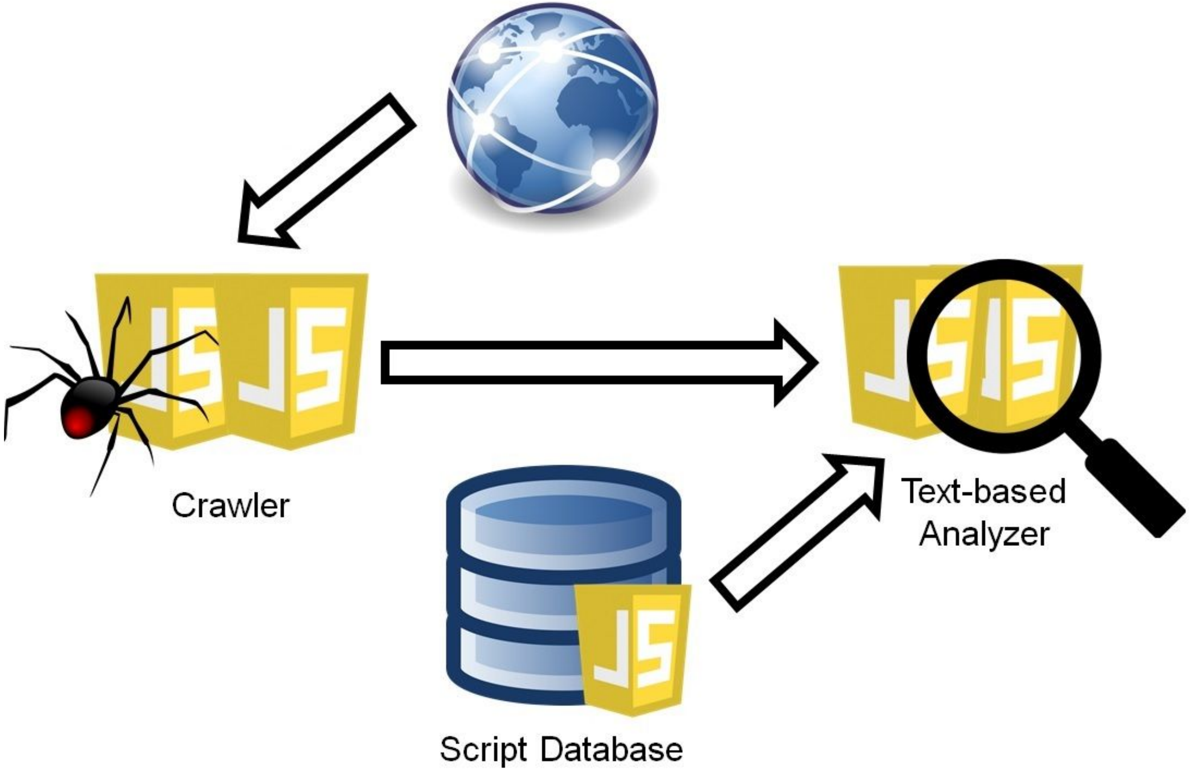
Script Representation

Using the Bag of Words approach
Modeled through Vector Space Model
Term Frequency – Inverse Document Frequency schema

Data Sources

Blacklists (that include scripts)
Open-source Projects
Academic Papers

TrackingInspector



Text-based Analyzer

Known Tracking Analysis

Detects versions or modifications

Computes the cosine similarity

Empirically computed threshold of 85%

Unknown Tracking Analysis

Finds new tracking script

Based on supervised machine learning

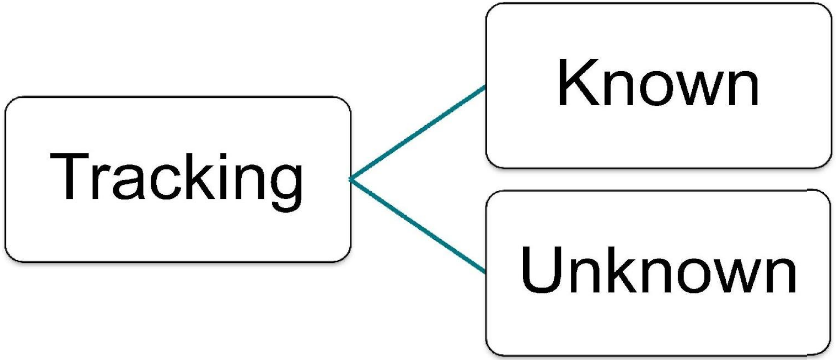
Data labeled as tracking/non-tracking

Large-Scale Analysis

The Crawler retrieved the scripts within the **Alexa top 1M**. Nearly **21M** script samples were downloaded, and just around 5% of the websites had no scripts at all.

We gathered data about the website and the top-level domains where the scripts were hosted (e.g., **reputation** and **category**).

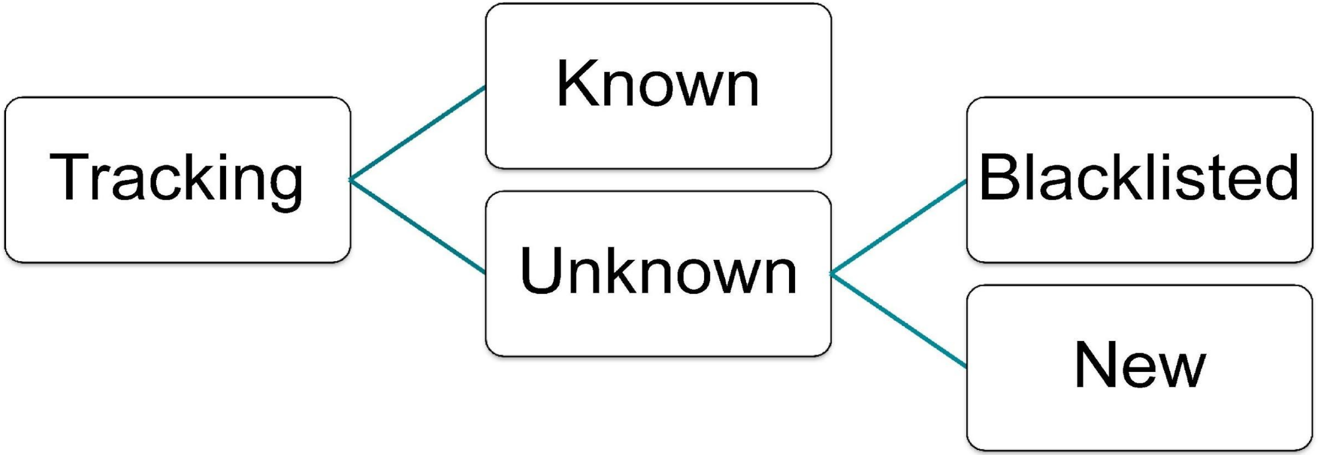
Tracking Script Classification


Tracking

Known

Unknown

Tracking Script Classification



Tracking Prevalence

The percentage of every type of tracking script in analyzed websites, can show **how distributed** are trackers in every case.

Known and new unknown scripts were in 83% of websites
Blacklisted unknown scripts were in 67% of the websites

Tracking Prevalence

The percentage of every type of tracking script in analyzed websites, can show **how distributed** are trackers in every case.

Known and new unknown scripts were in 83% of websites
Blacklisted unknown scripts were in 67% of the websites

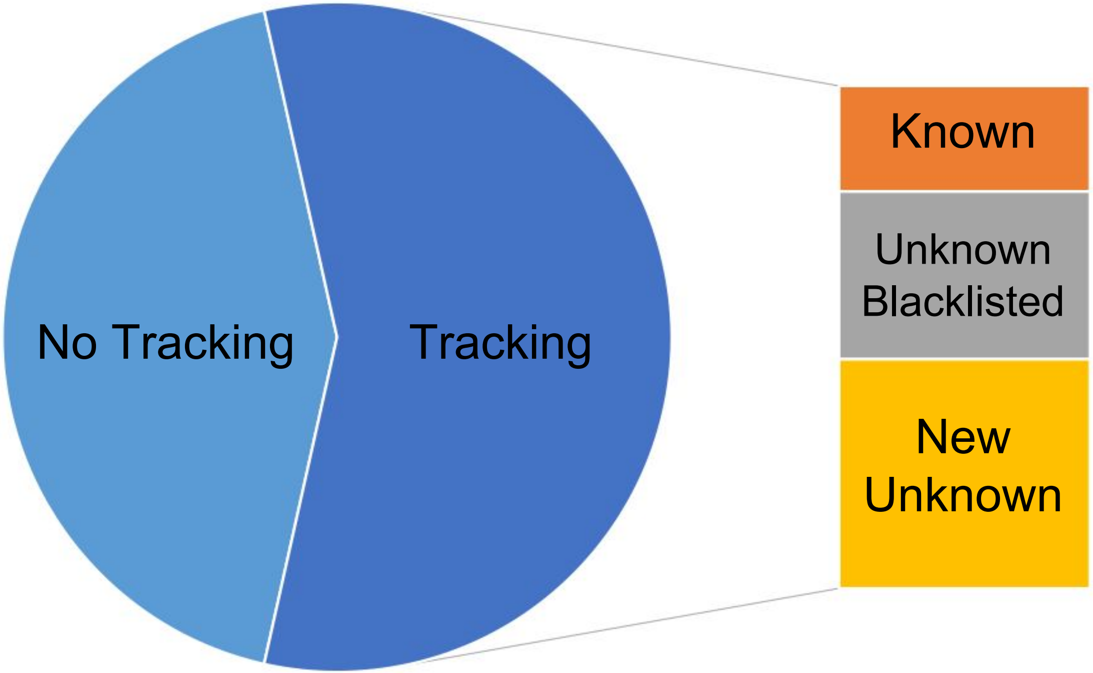
In total around **93%** of the websites have at least one of the above mentioned types of tracking scripts.

Tracking Demographics

The relation between domains with tracking scripts and their reputation (based on *webutation*) hinted that the presence of only **tracking affects the reputation**.

The top categories with **only tracking** scripts were *malicious*, *questionable*, *unknown*, and *websites with adult content*.

Tracking Script Distribution



Current Solutions

We measured the percentage of known script that **blacklisting** solutions would have blocked. **Combined** blacklisting solutions only blocked the 64.65% of the known scripts.

These results show that current anti-tracking solutions are **clearly not enough**, not only to fight against unknown tracking scripts, but also against modified known tracking scripts.

Script Renaming

Functionality script renaming

Modifies the name describing their goal

➔ *fingerprint.js and tracking.js*

Related script renaming

Changes the name to one directly or indirectly related to service or website using the script

➔ *chrysler.js* and *dodge.js*

Script Renaming

Random/neutral script renaming

Replaces the name randomly

➔ *penguin2.js* and *welcome.js*

Misleading script renaming

Changes their names to well-known non-tracking scripts (thinking in possible whitelists)

➔ *jquery.alt.min.js and j.min.js*

Conclusion

The results show that web tracking is **very extended**, and the presence of only tracking scripts is related to the **reputation**.

Current solutions cannot detect unknown tracking script, but they cannot even detect modifications of know ones.

Different script renaming **hiding techniques** are used nowadays to avoid existing blacklists.

Bob Dylan was Knockin' on Heaven's Door…

but we are…

Knockin' on Trackers' Door

iskander.sanchez@deusto.es iskander-sanchez-rola.github.io