

Die Fingerabdrücke der Fingerabdrucknehmer

Browser-Fingerprinting-Verhalten lernen und erkennen

Vorgetragen von Antonio Sarcevic - Hackerpraktikum - WS 2021/22



FH MÜNSTER
University of Applied Sciences

Inhalt

- Browser-Fingerprinting
- FP-INSPECTOR
- Fingerprinting der Top-100k
- Analyse der für Fingerprinting missbrauchten APIs
- Fazit





Browser-Fingerprinting

Online Tracking durch Fingerprinting (FP)

- Form von stateless tracking
- Missbrauchen von JavaScript APIs und HTTP Header um Geräte zu identifizieren
 - z.B. Canvas und User-Agent
- Aufdringlicher als Cookies da:
 1. undurchsichtig
 2. fehlende Kontrolle
- Fingerprinting wird eingesetzt für:
 - Bot-Detection (Google reCAPTCHA)
 - Cookies regenerieren oder synchronisieren
 - Cross-Site Tracking

Browser-Fingerprinting

Beispiel: Canvas Font Fingerprinting

```
1  fonts = ["monospace", /* ... */, "sans-serif"];  
2  
3  canvasElement = document.createElement("canvas");  
4  canvasElement.width = "100";  
5  canvasElement.height = "100";  
6  canvasContext = canvasElement.getContext("2d");  
7  
8  fpDict = {};  
9  for (i = 0; i < fonts.length; i++) {  
10    canvasContext.font = "16px " + fonts[i];  
11    fpDict[fonts[i]] = canvasContext.measureText("example").width;  
12  }
```

```
1  > fpDict  
2  « {"monospace":61.578125,"Comic Sans MS":60.53125,"Arial":60.4765625,"sans-serif":60.4765625}
```



Browser-Fingerprinting

Bietet Fingerprinting eindeutige und dauerhafte Identifikatoren?

- Fingerprint als "statistischer" Identifikator
 - Identifizierbarkeit abhängig von Anzahl Geräten mit gleicher Konfiguration
 - bzw. "uniqueness" des Fingerprints
- Laperdrix et al. (AmlUnique) und Eckersley (Panopticlick): 83% - 90% der Geräte eindeutigen Fingerprint
(biased)
- Broix et al. auf Seite eines französischen Verlags: 33.6% Geräte mit eindeutigem Fingerprint (mehr Daten für mehr uniqueness benötigt)
- Eckersley (Panopticlick): 37% wiederkehrende Besucher mehr als einen Fingerprint
 - 65% der Geräte über einfache heuristische Verknüpfung re-identifiziert
 - Vastel et al. (AmlUnique) verbesserten Heuristik: mögliche Verfolgung von Besuchern für Ø 74 Tage



Browser-Fingerprinting

Verbreitung

- 2013 Paper "[Cookieless Monster: Exploring the Ecosystem of Web-based Device Fingerprinting](#)"
 - 3 fingerprinting Anbieter
 - 40 FP-Seiten aus Alexa Top-10k (0.4%)
- 2013 Paper "[FPDetective: Dusting the Web for Fingerprinters](#)"
 - 404 FP-Seiten aus Alexa Top-1 Millionen (0.04%)
- Weitere Studien [37], [47], [54], [78] finden vermehrt FP-Seiten über die Jahre
- 2019 schreibt Washington Post: "At least a third of the [top] 500 sites [...] use hidden [fingerprinting] code."

Q FP-INSPECTOR

- Machine Learning Ansatz zur Erkennung von FP
- statische + dynamische JavaScript Analyse für Feature-Extraktion
- Maßnahmen gegen Browser-Fingerprinting zur Mitigation

Q FP-INSPECTOR

Design: Architekturdiagramm

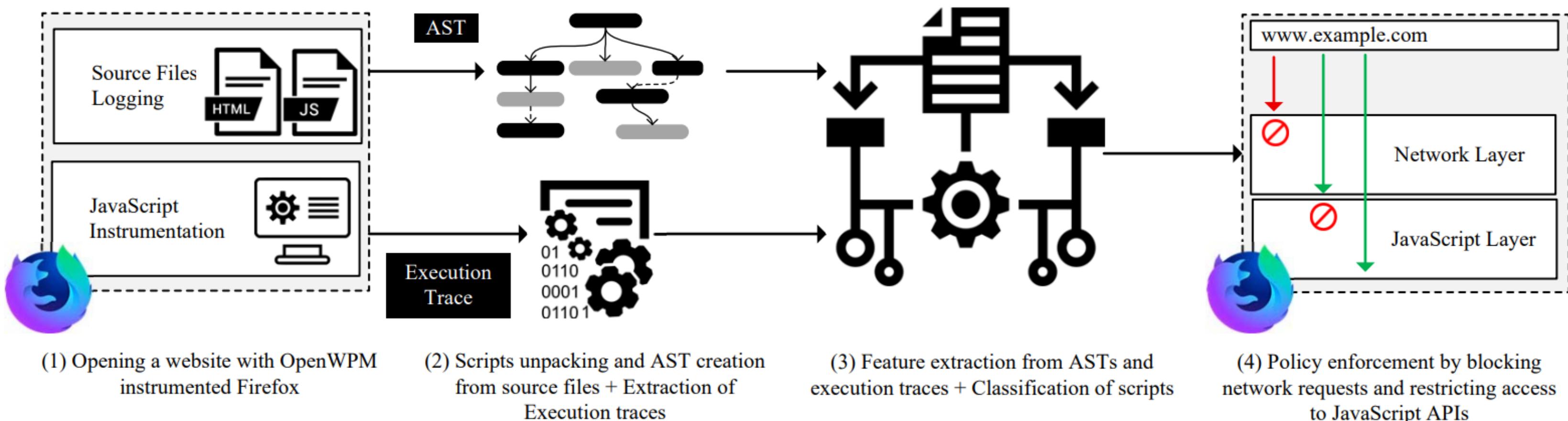


Fig. 1: FP-INSPECTOR: (1) We crawl the web with an extended version of OpenWPM that extracts JavaScript source files and their execution traces. (2) We extract Abstract Syntax Trees (ASTs) and execution traces for all scripts. (3) We use those representations to extract features and train a machine learning model to detect fingerprinting scripts. (4) We use a layered approach to counter fingerprinting scripts.

Q FP-INSPECTOR

Design: Erkennung - Script Monitoring

- Englehardt und Narayanan [54] entwickelten OpenWPM (Framework zur automatisierten Messung von Web Privacy)
- Crawlen mit OpenWPM: Sammeln von rohen Daten und execution traces für statische und dynamische Analyse
- Erweitern von OpenWPM
 - Speichern von HTML Dokumenten (zum sammeln von inline JavaScript)
 - Erweiterung der dynamisch aufgenommenen JavaScript API aufrufe

Q FP-INSPECTOR

Design: Erkennung - Statische Analyse

```
1 eval("Fonts =[\"m
2 CanvasElem = docu
3 ;CanvasElem.width
4 \\"100\\";context =
5 FPDict= {};for(i=
6 CanvasElem.font =
7 CanvasElem.measur
```

Script 1: A canvas font

```
1 // Canvas font fi
2 Fonts = ["monospa
3
4 CanvasElem = docu
5 CanvasElem.width
6 CanvasElem.height
7 context = CanvasE
8 FPDict= {};
9 for (i = 0; i < F
10 {
11   CanvasElem.font
12   FPDict[Fonts[i]
13     ").width;
```

Script 2: An unpack

■ Überwachtes Element

Static Features

ArrayExpression:monospace
MemberExpression:font
ForStatement:var
MemberExpression:measureText
MemberExpression:width
MemberExpression:length
MemberExpression:getContext
CallExpression:canvas

TABLE VII: A sample of features extracted from AST in Figure 2b.

(b) AST for unpacked script 2

Fig. 2: A truncated AST representation of Scripts 1 and 2. The edges represent the syntactic relationship between nodes. Dotted lines indicate an indirect connection through truncated nodes.

Q FP-INSPECTOR

Design: Erkennung - Dynamische Analyse

- dynamische
- Verwendet
- abgeleit

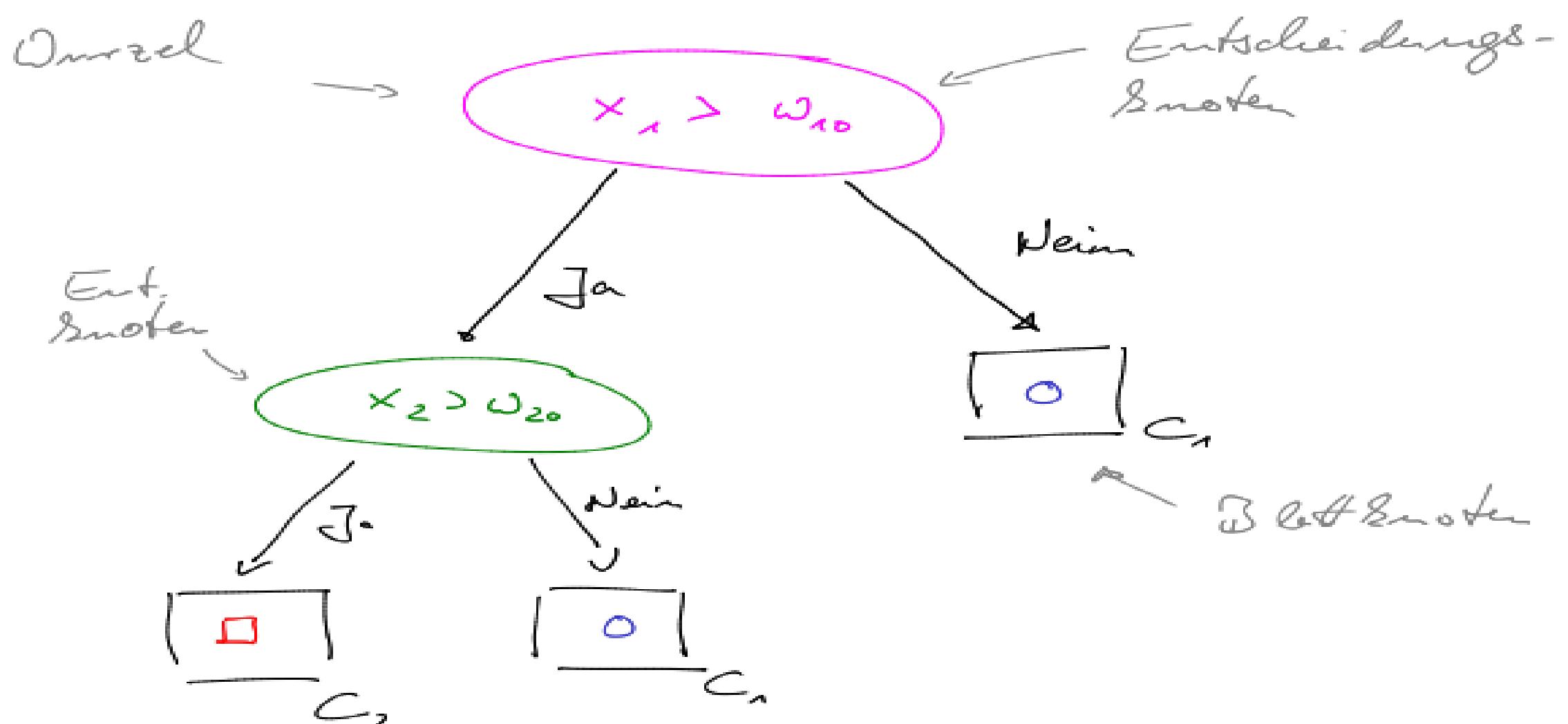
Feature Name	Feature Value
Document.createElement	True
HTMLCanvasElement.width	True
HTMLCanvasElement.height	True
HTMLCanvasElement.getContext	True
CanvasRenderingContext2D.measureText	True
Element Tag Name	Canvas
HTMLCanvasElement.width	100
HTMLCanvasElement.height	100
CanvasRenderingContext2D.measureText	7 (no. of chars.)
CanvasRenderingContext2D.measureText	N (no. of calls)

- Überwacht TABLE VIII: A sample of the dynamic features extracted from the execution trace of Script 3a.

Q FP-INSPECTOR

Design: Erkennung - Klassifizieren

- Entscheidungsbaum als Classifier
 - Knoten: Feature zum aufteilen ausw.
 - Blätter: FP / Non-FP
- Getrennte Bäume für statische und dy



Q FP-INSPECTOR

Evaluation: Genauigkeit der Erkennung

- Zur Evaluation werden gelabelte Beispiele von Non-FP-Scripte und FP-Scripte (ground truth) benötigt
- Labeln von (Non-)FP-Scripte mit angepassten Heuristiken aus Englehardt und Narayanan [54]
 - Heuristiken nie perfekt
- Data Collection mit OpenWPM
 - Alexa Top-10k + 10k (gesampled aus Top 10k-100k) -> 20k Seiten gecrawed
 - 17.629 Seiten mit 153.354 verschiedenen Scripte gefunden

Q FP-INSPECTOR

Evaluation: Genauigkeit der Erkennung

- *Enhancing Ground Truth*
 - iteratives Re-Training und manueller Korrektur der Datengrundlage
 - bei Mismatch: automatisierten Report
 - Manuelle Auswertung nach "heuristic-like behaviors"

Itr.	Initial		New Detections		Correct Detections		Enhanced	
	FP	NON-FP	FP	NON-FP	FP	NON-FP	FP	NON-FP
S1	884	142,642	150	232	103	10	977	142,549
S2	977	142,549	109	182	84	5	1,056	142,470
S3	1,056	142,470	76	158	53	1	1,108	142,418
D1	928	152,426	11	52	4	9	923	152,431
D2	923	152,431	8	35	4	1	926	152,428
D3	926	152,428	13	36	5	2	929	152,425

TABLE I: Enhancing ground truth with multiple iterations of retaining. Itr. represents the iteration number of training with static (S) and dynamic (D) models. New Detections (FP) represent the additional fingerprinting scripts detected by the classifier and New Detections (NON-FP) represent the new non-fingerprinting scripts detected by the classifier as compared to heuristics. Whereas Correct Detections (FP) represent the manually verified correct determination of the classifier for fingerprinting scripts and Correct Detections (NON-FP) represent the manually verified correct determination of the classifier for non-fingerprinting scripts.

Q FP-INSPECTOR

Evaluation: Genauigkeit der Erkennung

Classifier	Heuristics (Scripts/Websites)	Classifiers (Scripts/Websites)	FPR	FNR	Recall	Precision	Accuracy
Static	884 / 2,225	1,022 / 3,289	0.05%	15.7%	85.5%	92.7%	99.8%
Dynamic	928 / 2,272	907 / 3,278	0.005%	5.3%	96.7%	99.1%	99.9%
Combined	935 / 2,272	1,178 / 3,653	0.05%	6.1%	93.8%	93.1%	99.9%

TABLE II: FP-INSPECTOR’s classification results in terms of recall, precision, and accuracy in detecting fingerprinting scripts. “Heuristics (Scripts/Websites)” represents the number of scripts and websites detected by heuristics and “Classifiers (Scripts/Websites)” represents the number of scripts and websites detected by the classifiers. FPR represents false positive rate and FNR represent false negative rate.

- Kombination statischer und dynamischer Analyse hat sich gelohnt:
 - 94,46% der Scripte die nur durch statische Analyse erkannt wurden waren ruhend
 - 92,30% der Scripte die nur durch dynamische Analyse erkannt wurden waren obfuscated / stark minified
- Classifier erkennt 26% mehr Scripte als pure Heuristiken

Q FP-INSPECTOR

Design: Mitigation

- Gegenmaßnahmen:
 - Content Blocking
 - API Restrictions
- Website Breakage wichtig!
- FP-Inspector Mitigations-Modi:
 1. pauschale API Restriction
 2. gezielte API Restriction
 3. Request Blocking
 4. Hybrid

Q FP-INSPECTOR

Evaluation: Breakage

- Manuelles Testen
 - 50 zufälligen FP-Seiten + 11 Non-FP-Seiten (die unter damaligen Anti-FP Methoden von Firefox brechen)
 - Ausschalten der damaligen Anti-FP Methoden von Firefox
 - Testen der Seiten, jeweils im vanilla Firefox und mit jedem Mitigations-Modi

Policy	Major (%)	Minor (%)	Total (%)
Blanket API restriction	48.36%	19.67%	68.03%
Targeted API restriction	24.59%	5.73%	30.32%
Request blocking	44.26%	5.73%	50%
Hybrid	38.52%	8.19%	46.72%

TABLE III: Breakdown of breakage caused by different countermeasures. The results present the average assessment of two reviewers.

Q FP-INSPECTOR

Limitationen

- Umgehen der Entdeckung durch Scriptstreuung
 - Beziehung zwischen Scripte nicht betrachtet
- Umgehung von Gegenmaßnahmen durch Scriptverschmelzung
 - Granularität von gezielter API Restriction auf Script-Ebene

8 Fingerprinting der Top-100k Seiten

Mehr als ein Viertel der Top-Websites nehmen Fingerabdrücke von Nutzern

Rank Interval	Websites (count)	Websites (%)
1 to 1K	266	30.60%
1K to 10K	2,010	24.45%
10K to 20K	981	11.10%
20K to 50K	2,378	8.92%
50K to 100K	3,405	7.70%
1 to 100K	9,040	10.18%

TABLE IV: Distribution of Alexa top-100K websites that deploy fingerprinting. Results are sliced by site rank.

8 Fingerprinting der Top-100k Seiten

Fingerprinting ist auf seiten am häufigsten zu finden

[8] Fingerprinting der Top-100k Seiten

Fingerprinting ist auf Nachrichtenseiten am häufigsten zu finden

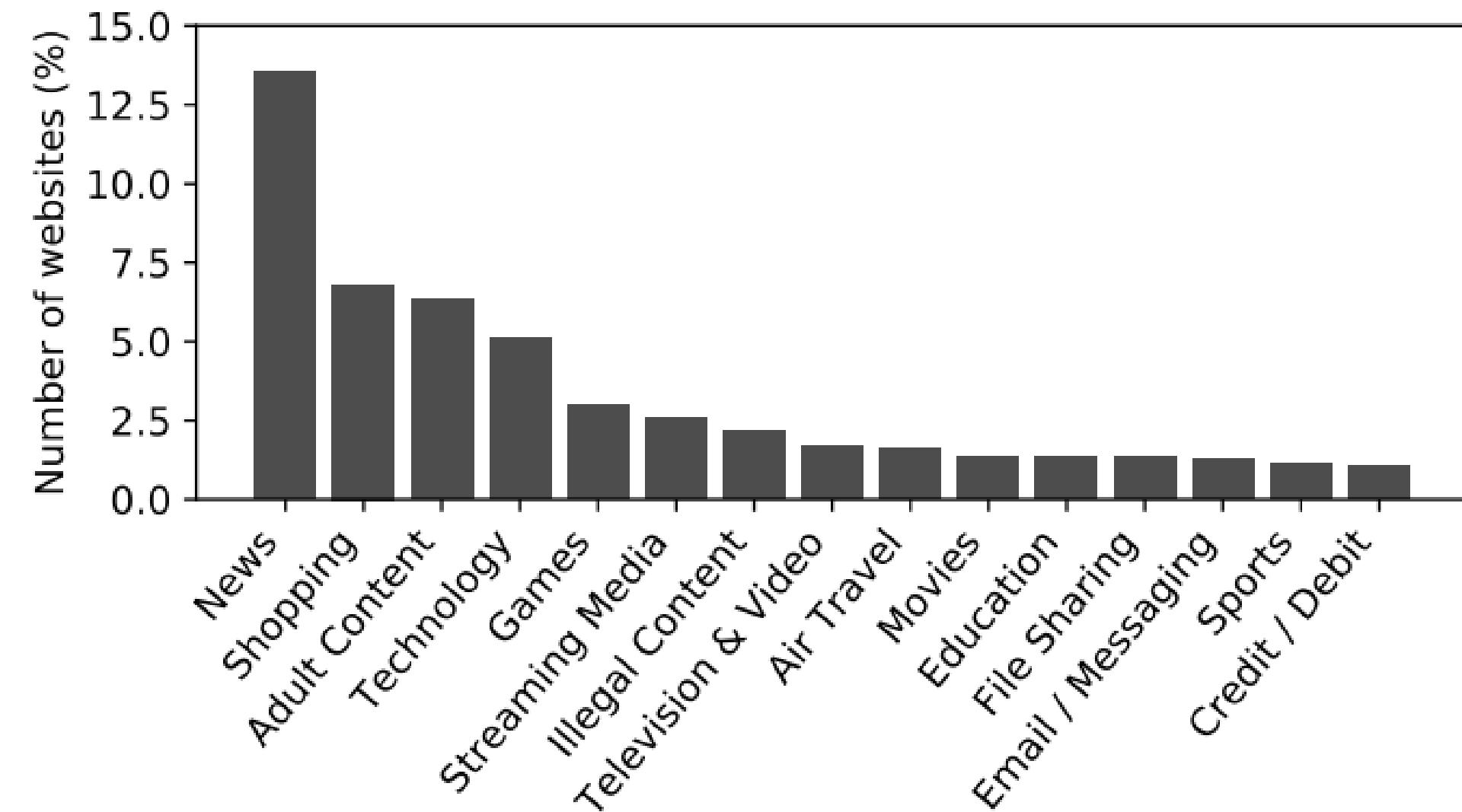


Fig. 4: The deployment of fingerprinting scripts across different categories of websites.

8 Fingerprinting der Top-100k Seiten

Fingerprinting wird zur Bekämpfung von Anzeigenbetrug eingesetzt, aber auch für potenzielles Cross-Site-Tracking

- Third Party FP Anbieter:
 - 3,78% Tracking Dienste
 - 17,28% betreiben Cookie Syncing

Vendor Domain	Tracker	Websites (count)
doubleverify.com	Y	2,130
adsafeprotected.com	Y	1,363
alicdn.com	N	523
adscos.re	N	395
yimg.com	Y	246
2,344 others	Y(86)	5,702
Total		10,359 (9,040 distinct)

TABLE V: The presence of the top vendors classified as fingerprinting on Alexa top-100K websites. Tracker column shows whether the vendor is a cross-site tracker according to Disconnect's tracking protection list. Y represents yes and N represents no.

 Analyse der für Fingerprinting missbrauchten APIs

Analyse der für Fingerprinting missbrauchten APIs

Fingerprinting der Funktionalität

1. Berechtigung Fingerprinting
2. Peripherie Fingerprinting
3. API Fingerprinting

Algorithmisches Fingerprinting

1. Timing Fingerprinting
2. Animation Fingerprinting
3. Audio Fingerprinting
4. Sensor Fingerprinting



Fazit

- FP-Inspector: Machine Learning-basierter syntaktisch und semantischer Ansatz
 - Erkennt 26% mehr FP-Scripte als pure Heuristiken
 - Bricht Webseiten bis zu 2x weniger
- Analyse der Top Seiten ergibt:
 - Fingerprinting so weit verbreitet wie nie zuvor: Von Top-100k Seiten 10.18% mit FP Scripte
 - 2,349 FP Domains wurden an Tracking Listen gesendet
- Neue FP-APIs gefunden: an Browser Anbieter und Standard Autoritäten gemeldet
- Tools für weitere Forschung veröffentlicht: (github.com/uiowa-irl/FP-Inspector)
 - Patches für OpenWPM
 - Fingerprinting Countermeasure Prototype Browsererweiterung
 - Liste neu entdeckter FP-Anbieter
 - Bug Reports für Tracking Listen und Browser Anbieter

≡ Weitere Paper

- "Did I delete my cookies? Cookies respawning with browser fingerprinting" Fouad et al.

- "Our results show that 1,150 out of the top 30, 000 Alexa websites deploy this tracking mechanism."
- "FP-Radar: Longitudinal Measurement and Early Detection of Browser Fingerprinting" Bahrami et al.

- "In this paper, we propose FP-Radar, a machine learning approach that leverages longitudinal measurements of web API usage on top-100K websites over the last decade, for early detection of new and evolving browser fingerprinting techniques."
- "EssentialFP: Exposing the Essence of Browser Fingerprinting" Sjösten et al.

- "We argue that the pattern of (i) gathering information from a wide browser API surface (multiple browser-specific sources) and (ii) communicating the information to the network (network sink) captures the essence of fingerprinting. This pattern enables us to clearly distinguish fingerprinting from similar types of scripts like analytics and polyfills. To implement EssentialFP we leverage, extend, and deploy JSFlow, a state-of-the-art information flow tracker for JavaScript, in a browser."

Vielen Dank für Ihre Aufmerksamkeit

src: U. Iqbal, S. Englehardt, and Z. Shafiq, “Fingerprinting the fingerprinters: Learning to detect browser fingerprinting behaviors”, 2020.

Vorgetragen von Antonio Sarcevic - Hackerpraktikum - WS 2021/22



FH MÜNSTER
University of Applied Sciences