Auf dem Weg zu Website-Fingerprinting in der Praxis

Identifizierung von Webseiten mit dem multinomialen Naïve-Bayes-Klassifizierer

Dominik Herrmann

Lehrstuhl Management der Informationssicherheit
Universität Regensburg

11.02.2008

Überblick

Website-Fingerprinting

- Idee und Zielsetzung

Verbessertes Website-Fingerprinting-Verfahren

- Multinomialer Naive-Bayes-Klassifizierer
- Transformation der Häufigkeitsvektoren

Evaluation und Analyse

- Versuchsaufbau
- Evaluation mit OpenSSH
- Schutz durch JonDonym und Tor

Forschungsfragen

Ziel: Identifizierung verschlüsselt übertragener Webseiten

Wie sicher sind datenschutzfreundliche Übertragungsverfahren?

- Weitverbreitete Annahmen:
  - Abruf von Webseiten über verschlüsselnde Tunnel (z. B. OpenSSH, VPNs, JonDonym, Tor) gewährleistet Vertraulichkeit
  - Angreifermodell solcher Dienste: Schutz der Inhalte (auch URLs!) vor *lokalem Angreifer*

- Mit Website-Fingerprinting kann auch ein lokaler Angreifer die abgerufenen URLs ermitteln

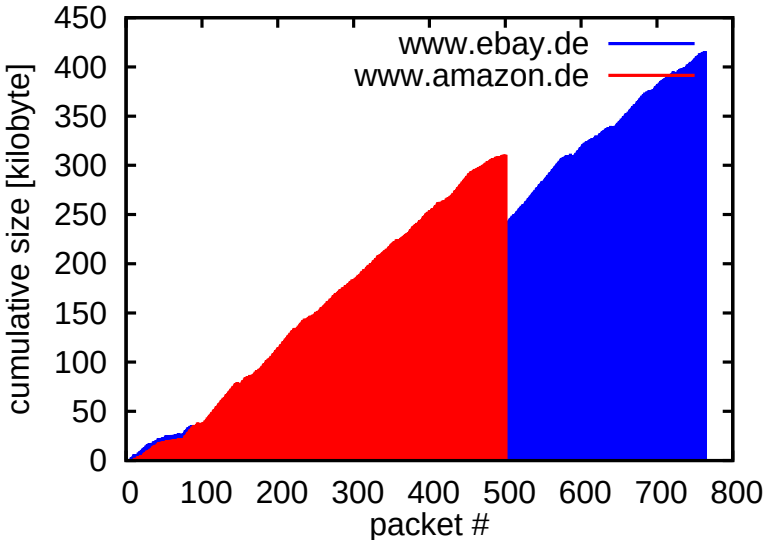
- Verschiedene Motive denkbar:
  - User-Profiling: *Welche Webseiten wurden abgerufen?*
  - Strafverfolgung: *Wurde Webseite* X *abgerufen?*

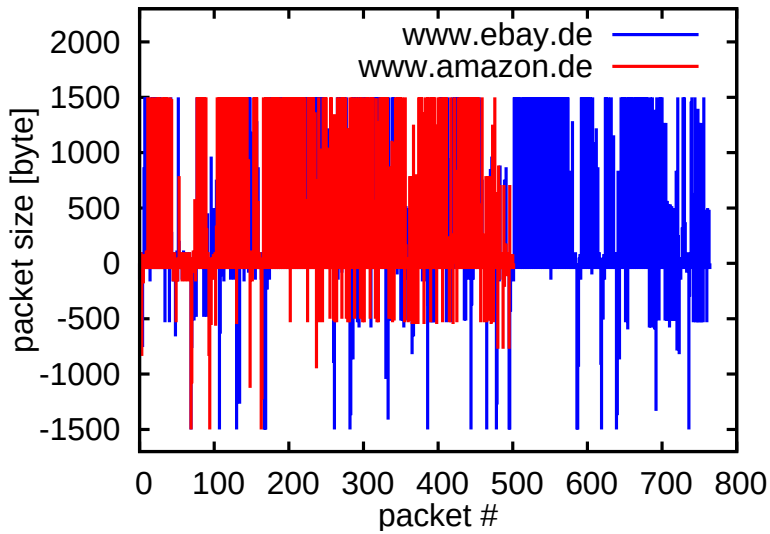
Charakteristischer Traffic beim Abruf einer Webseite

Website-Fingerprinting basiert auf der Erkennung typischer Muster im Datenverkehr

- IP-Paketgrößen als robustes Merkmal
- Pakete in Senderichtung (Client → Server): negative Größe

- Netzwerk-Traces für www.amazon.de und www.ebay.de:





Idee: Anwendung von Data-Mining-Techniken

Data-Mining-Know-How und Verfahren für Website-Fingerprinting nutzen

- Aber: Korrelation von Netzwerk-Traces nicht trivial
  $\longrightarrow$ Vielversprechendes Anwendungsfeld für Data-Mining

- Website-Fingerprinting als Klassifizierungsproblem
  - Klassen: Webseiten (URLs)
  - Instanzen: *tcpdump*-Netzwerk-Traces
  - Attribute: Häufigkeiten der IP-Paketgrößen
- Im Text-Mining verbreitet:
  Multinomialer Naïve-Bayes Klassifizierer

Basis für Naïve-Bayes: Termhäufigkeitsvektoren

Netzwerk-Traces lassen sich als (Term-)Häufigkeitsvektoren darstellen

1. Ermittlung des Vokabulars V aller Trainingsinstanzen
2. Erzeugung der Häufigkeitsvektoren $\mathbf{f}$

$$
\begin{aligned}
d_1 &= \text{"}-152\ \ 1500\ \ -52\ \ 1050\ \ 1500\ \ -80\text{"} \\
d_2 &= \text{"}-148\ \ 476\ \ 1500\ \ -80\text{"}
\end{aligned}
$$

$$
\begin{aligned}
V &= \{-152, 1500, -52, 1050, -80, -148, 476\} \\
\mathbf{f} &= \langle f_{-152}, f_{1500}, f_{-52}, f_{1050}, f_{-80}, f_{-148}, f_{476} \rangle
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{f}_{d_1} &= \langle 1, 2, 1, 1, 1, 0, 0 \rangle \\
\mathbf{f}_{d_2} &= \langle 0, 1, 0, 0, 1, 1, 1 \rangle
\end{aligned}
$$

Multinomialer Naïve-Bayes-Klassifizierer

Trotz naiver Annahmen (positional/conditional independence) gute Erkennungsraten

d gehört zu c: $P(c|d) = \frac{P(c)\,P(d|c)}{P(d)}$ hier: $P(c|d) \propto P(d|c)$

Wie gut wird c durch d repräsentiert? $\rightarrow$ Wie wahrscheinlich ist es, d's Terme in Dokumenten, die zur Klasse c gehören, anzutreffen?

$$P(d|c) = P(\mathbf{f}|c) = C_{multi} \cdot \prod_{t \in V} P(t|c)^{f_{t,d}}$$

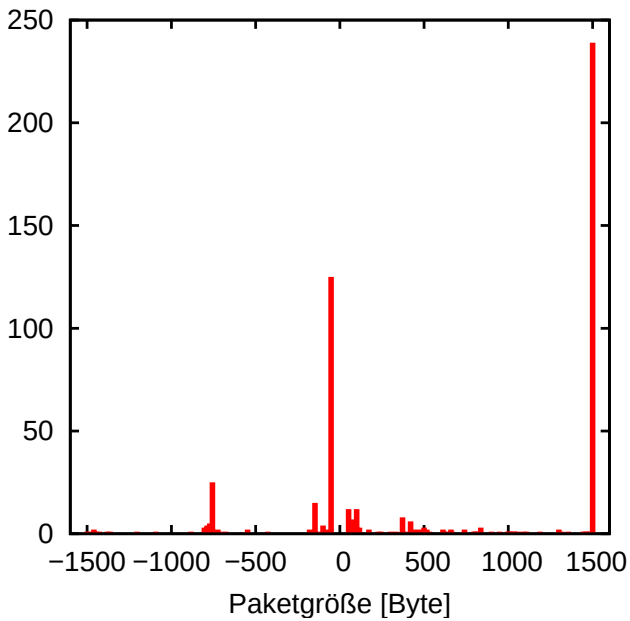
$$\hat{P}(t|c) = \frac{f_{t,c}}{\sum_{t' \in V} f_{t',c}}$$

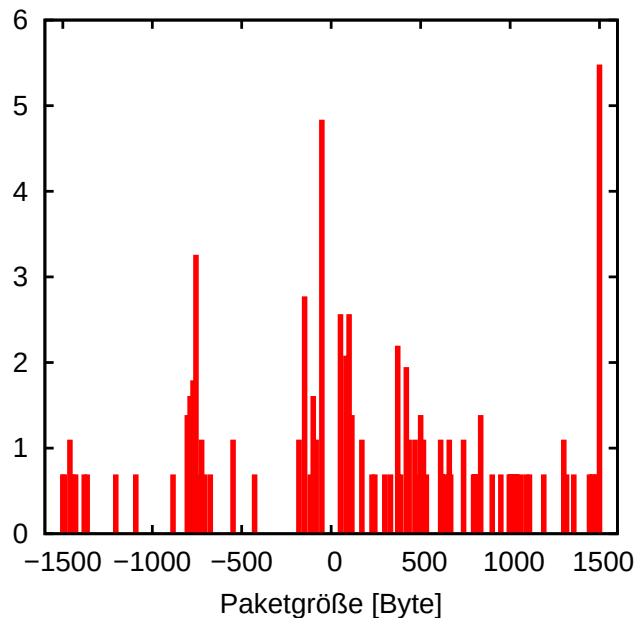
(nach [Manning])

Transformation der Häufigkeitsvektoren

Durch Transformation der Häufigkeitsvektoren lässt sich die Erkennungsleistung steigern

- TF-Transformation: $f^*_{t,d} = \log(1 + f_{t,d})$ nach [Manning]
- Normalisierung: $f^{\text{norm}}_{t,d} = \frac{f^*_{t,d}}{\left\| \left\langle f^*_{t_1,d}, \ldots, f^*_{t_n,d} \right\rangle \right\|} = \frac{f^*_{t,d}}{\sqrt{\sum_{t' \in V} (f^*_{t',d})^2}}$





Versuchsaufbau für Evaluation

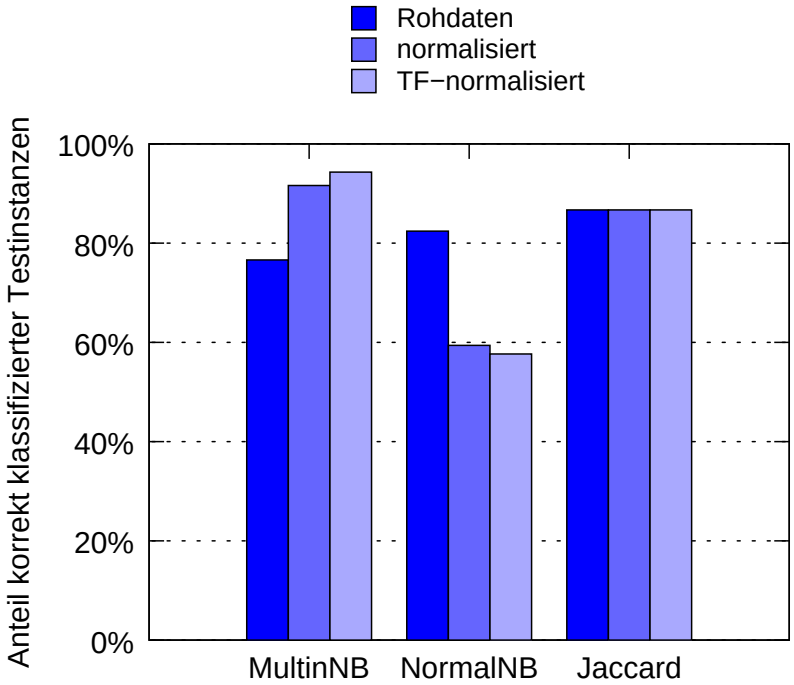
Versuchsaufbau vergleichbar mit bisherigen Veröffentlichungen

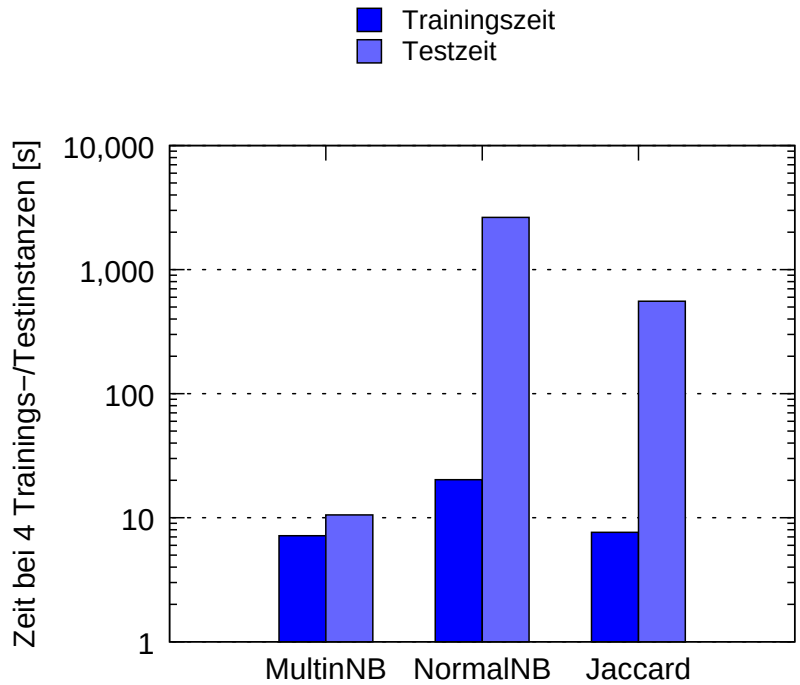
- 775 populäre Webseiten, Abruf mit Firefox [JSSh]
- Aufzeichnung der IP-Paket-Header mit *tcpdump*
- 25 Stichproben pro Versuch, jeweils 10 Testinstanzen
- Getestete Systeme: OpenSSH, Tor, JonDonym

- Benchmark: 70 % korrekt klassifizierte Instanzen bei OpenSSH [Liberatore 2006]

Vergleich mit anderen Website-Fingerprinting-Verfahren

Multinomialer Naïve-Bayes-Klassifizierer ist schneller und besser als etablierte Verfahren

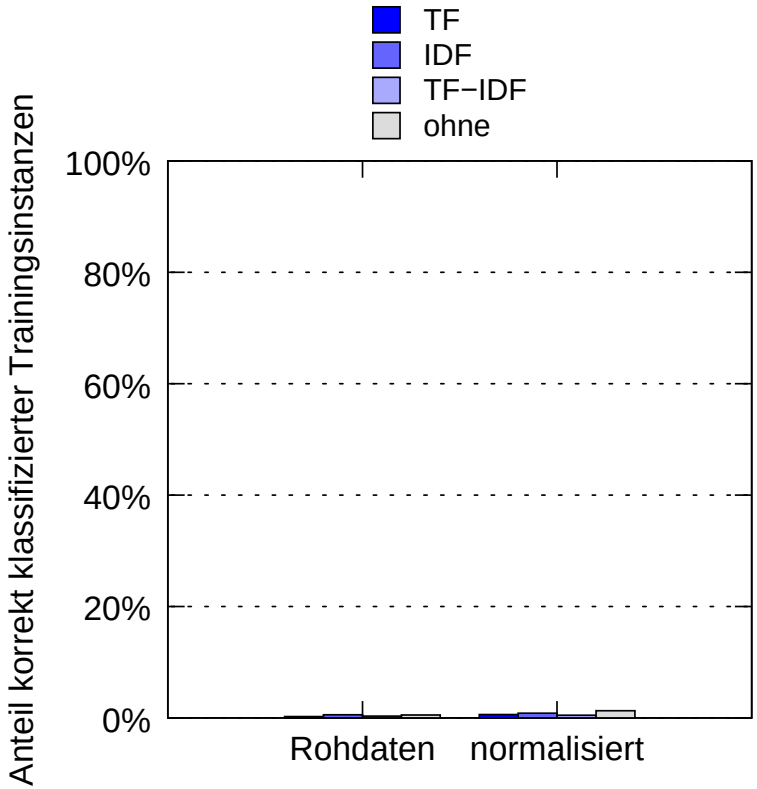




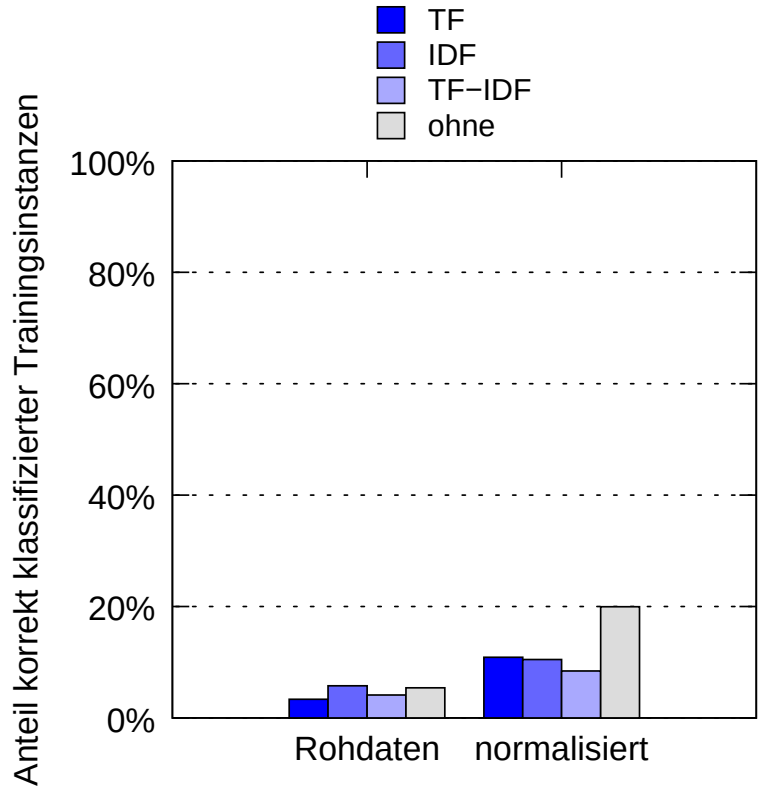
- Multinomialer-Naïve-Bayes-Klassifizierer:
  94 % Erkennungsrate bei 4 Trainingsinstanzen

Evaluierung mit JonDonym und Tor

Heutige Anonymisierer bieten bereits guten Schutz gegen Website-Fingerprinting



Tor



JonDonym

Offene Forschungsfragen

Herausforderungen für Verwendung von Website-Fingerprinting in der Praxis

Effiziente Gegenmaßnahmen Padding ist wirksam, aber teuer. Alternativen denkbar: z. B. Burst-Proxy

Gezielte Erleichterung Kann Betreiber einer Webseite Identifizierung mutwillig erleichtern?

Überbrückung_von Anonymisierern Verteilter Angriff; lassen sich Anonymisierer damit überbrücken?

Einfluss von Caching Funktioniert Fingerprinting bei aktiviertem Browser-Cache?

Einsatz in Praxis Unterstellung von Denkpausen ist idealistisch; bei Hintergrundtraffic überhaupt durchführbar?

Bedeutung der False Positive Rate Bislang Fokus auf True Positives; False Positives ebenso relevant!

Robustheit Einfluss von Browser, Plugins, ISP?

Zusammenfassung

- Text-Mining-Verfahren auf Netzwerk-Traces anwendbar
- Vorgestelltes Verfahren hat hohe Erkennungsleistung bei OpenSSH
- Existierende Anonymisierer erschweren Fingerprinting

Referenzen


M. Liberatore and B. N. Levine: Inferring the source of encrypted HTTP connections. In CCS '06: Proc. of the 13th ACM conference on Computer and communications security, p. 255-263, New York, 2006.


JSSh – TCP/IP JS Shell Server for Mozilla (http://croczilla.com/jssh)


C. D. Manning, P. Raghavan, and H. Schütze: Introduction to Information Retrieval (draft; Nov. 17, 2007). Cambridge University Press, 2007 (http://nlp.stanford.edu/IR-book/pdf/irbookprint.pdf).