Fingerprinting Techniques for Target-oriented Investigations in Network Forensics

Dominik Herrmann, Karl-Peter Fuchs, and Hannes Federrath
Vogt-Kölln-Straße 30, D-22527 Hamburg
{herrmann, fuchs, federrath}@informatik.uni-hamburg.de

Abstract: Fingerprinting techniques are receiving widespread attention in the field of information security. In this paper we argue that they may be of specific interest for the field of network forensics. In three case studies, we explore the use of fingerprinting techniques to improve and extend current investigative methods and showcase why fingerprinting allows for more target-oriented investigations than current practices. In each case study, we review the applicability of the current state of the art from the field of information security. The paper is intended to be a starting point for a discussion about the opportunities and concerns that may result from using evidence gained by fingerprinting techniques in criminal investigations.

1 Introduction

In the late 19th century it was discovered that fingerprints of humans are a distinctive biometric trait [Fau80, Gal92]. Today fingerprints play an important role in criminal investigations, and they serve as convincing evidence for the association of a suspect with a crime in court all over the world.

Apart from the biometric fingerprints mentioned above the computer science community is also aware of *device fingerprints* that capture characteristic traits of devices in a technical system. The notion of such fingerprints came up in the era of the Cold War, where defense forces became interested in deducing the type and make of missiles and satellites solely based on their radio echo. It was found that different devices exhibit distinctive patterns, “radar fingerprints”, that can be observed during their flight [Lac67].

Fingerprinting techniques are increasingly gaining attention: According to Google Scholar, in the years 2010, 2011 and 2012 more than 200 security-related academic papers containing the term “fingerprinting” in the title have been published annually. The motives for studying these techniques relate to both, their opportunities, e. g., their utility for network intrusion detection systems [HBK04] or as an additional authentication factor [XGMT09], as well as to the risks entailed, e. g., data leakage [DCGS09] or infringement of personal privacy [BFGI11].

Based on talks with lawyers and investigators we believe that the utility of fingerprinting for criminal investigations is relatively unknown so far. Historically, computer forensics was mainly concerned with analyzing hard disks found in computers seized by means of a warrant, e. g., after raiding the home of a purported suspect [Cas09]. However, due to full-disk encryption (FDE), obtaining incriminating data from seized hard disks has become more difficult recently [CFGS11]. Moreover, data is increasingly stored “in the cloud”, where legal accessibility often depends on coordination of and cooperation between authorities and service providers in multiple countries. Due to unresolved legal issues relating to such multi-jurisdiction investigations policy makers are concerned that investigators may be barred from obtaining relevant pieces of evidence in the future [CL10].

Due to this development investigators have started to turn their attention to the field of *network forensics*, which studies the probative value of network traffic in criminal investigations [Cas04, DH12]. For this purpose law enforcement agencies (LEAs) have several investigative measures at their disposal (in the following, we use Germany as an example): Firstly, given an IP address that has been involved in criminal activities they can *request customer subscription data from an ISP* to determine the identity of the respective account holder (“Bestandsdatenauskunft”, cf. § 113 TKG and § 100j StPO). Secondly, they can request an ISP to perform *lawful interception*, i. e., to store and disclose the network traffic of a specific customer (“Telekommunikationsüberwachung”, cf. §§ 100 a and 100 b StPO). Having access to the network traffic, investigators are facing **two common challenges**: Firstly, they may want to establish an association between criminal activities detected on the network and an actual perpetrator **(C1: “who did it?”)**. Secondly, they may be struggling to find evidence for criminal activities, when traffic is encrypted during transport **(C2: “what was done?”)**.

Further, policy makers around the world have tried to overcome these challenges by introducing new data retention laws (“Vorratsdatenspeicherung”) as well as legalizing the use of police-operated malware to intercept traffic before it is encrypted and sent over the network (“Bundestrojaner”). Critics argue that such efforts are disproportionate, because they lay the ground for uncontrolled surveillance, which also resulted in strict decisions of the German constitutional court (BVerfG, 27.2.2008 – 1 BvR 370/07; BVerfG, 2.3.2010 – 1 BvR 256/08). Before considering the deployment of disproportionate measures we suggest to fully leverage the potential of already existing investigative measures that are both, more target-oriented and incident-related. In this paper we demonstrate in three case studies that the application of fingerprinting techniques can play a vital role in this regard.

The rest of the paper is organized as follows: We outline the commonly used fingerprinting approach in Sect. 2, proceeding with our three case studies in Sects. 3, 4 and 5. After that, we discuss limitations and concerns in Sect. 6, before we conclude the paper in Sect. 7.

2 Fundamentals: Fingerprinting and Related Techniques

Fingerprinting techniques allow for the (re-)identification of a subject or an object (or “entity” in general) based on characteristic traits, i. e., its *fingerprint*. In contrast to *explicit* identifiers such as a serial number the fingerprint typically captures *implicit properties*.

Typically the fingerprinting approach consists of two stages. In the first stage, also called *training stage*, the fingerprints of a set of entities are recorded and stored together with the identity of the entity in a database. In the second stage, the *identification stage*, the characteristics of an entity whose identity is unknown are observed. The identity of the unknown entity is then inferred by comparing its fingerprint with all known fingerprints.

Fingerprinting research has been called both, an *art* as well as *engineering* [TDH03]. The art refers to designing a set of *suitable features* that facilitates identification of entities. Like in biometrics, good features are *distinctive* and *permanent* at the same time [JRP04], i. e., the features of each entity are characteristic enough so that one can differentiate various entities based on them, while one given entity exhibits the same (or at least a very similar) fingerprint every time it is encountered. On the other hand, the efficient extraction and robust matching of fingerprints is facilitated by the application of concepts drawn from engineering, e. g., pattern matching, statistical modeling and machine learning.

Watermarking and Correlation Techniques We point out that the fingerprinting approach outlined above is different from *watermarking techniques* [WCJ07, HB11, HKB12, HB13], which allow an investigator to trace back received packets to their true source, even when an adversary obfuscates his identity by using multi-hop anonymization networks such as Tor (http://torproject.org, [DMS04]) or compromised hosts, so-called “stepping stones” [YE00, ZP00]. To this end the investigator *tags* singular network flows at one location, e. g., by means of artificially delaying packets, and tries to detect them again at a different location, e. g., on the dial-up line of a household. If the watermark is embedded at a location, where the fact that all network traffic relates to criminal activities is known for sure (e. g., on the uplink of an online shop for hard drugs), the investigator can use this information as indication that a suspect has been involved in criminal activities.

Watermarking techniques require active interference in network communications, which may be problematic from a legal perspective. Although less effective, *flow correlation techniques* [WRW02, JWJ$^+$13] may be more suitable for the purpose of establishing an association between criminal activities and a suspect, because they can be applied by a *passive* observer. Flow correlation techniques extract and match *already existing* patterns by observing packet timings and traffic volumes. However, like watermarking techniques they require the investigator to have the capability to observe traffic at (at least) *two locations* in the network at the same time: Apart from intercepting the traffic of the suspect – which is also a requirement for fingerprinting techniques – investigators have to additionally tap the traffic of the server, whose location is typically either unknown (e. g., in the “cloud”, where virtual machines are relocated on the fly) or under the authority of a foreign jurisdiction. Therefore, the utility of watermarking and flow correlation techniques for criminal investigations is more limited in comparison to fingerprinting techniques, which can be applied by a passive observer with access to a single site only.

3 Case Study 1: Inferring Content of Encrypted Communications

In this first case study we demonstrate the utility of fingerprinting for Challenge C2: “what was done?”. We assume a scenario in which investigators suspect that Mallory, whose identity has been disclosed to police by his ISP, is consuming child pornography on the

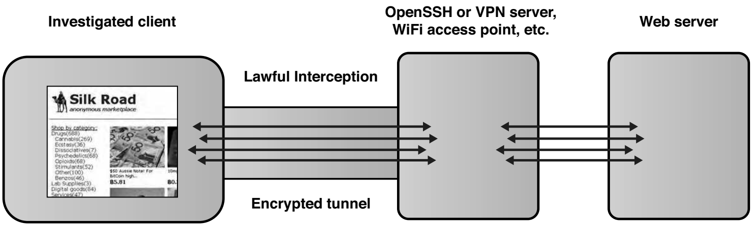

Figure 1: Website Fingerprinting Scenario

Internet. The prosecutor wants to obtain corroborating evidence that confirms the accusations before bringing the case to court. Thus, the objective of the prosecution is to find evidence that specific activities have been carried out by the suspect, e. g., that he visited a website, downloaded a movie or entered specific search terms into a search engine [OLL11]. A classical investigative measure involving digital forensics would consist in raiding Mallory's apartment, seizing his computer and trying to extract the contents of the browser history and cache files [Per09]. However, in our scenario no evidence can be obtained, because Mallory uses FDE on his machine.

As an alternative measure investigators might opt for lawful interception, i. e., compel Mallory's ISP to tap and disclose his network traffic to the LEA. However, in our case study this approach will not result in any corroborative evidence, either, because Mallory makes sure that all potentially incriminating activity is perpetrated via encrypted channels: He uses the VPN service of an offshore provider whenever he connects to the Internet, and "serious business" is conducted only via the Tor network (cf. Fig. 1).

Apparently, in such a scenario the only opportunity for law enforcement to obtain evidence for Mallory's criminal involvement seems to consist in questionable measures like infecting his machine with police-operated malware to perform a remote search. However, there are fingerprinting techniques that can be applied by investigators to infer (part of) Mallory's activities, e. g., whether he accessed a specific site or content, solely based on the traffic metadata that is not encrypted. This approach is known as *traffic analysis* [Ray00].

3.1 Packet-based Website Fingerprinting

Packet-based website fingerprinting exploits the fact that many websites are a unique composition, consisting of multiple source and media files. The concrete number and size of the individual files of a site has been shown to be a distinctive and permanent feature suitable for fingerprinting [Hin02, SSW$^+$02]. When a site is downloaded within an encrypted channel such as a VPN, the sizes of individual files cannot be determined by an observer anymore. However, fingerprints can still be built from features derived from the encrypted TCP flows. *Inter-arrival times* of packets have been shown to be characteristic [BLJL05], but due to network latency this feature is not very robust. The *distribution of the size of the IP packets* is a more promising candidate (cf. Fig. 2).

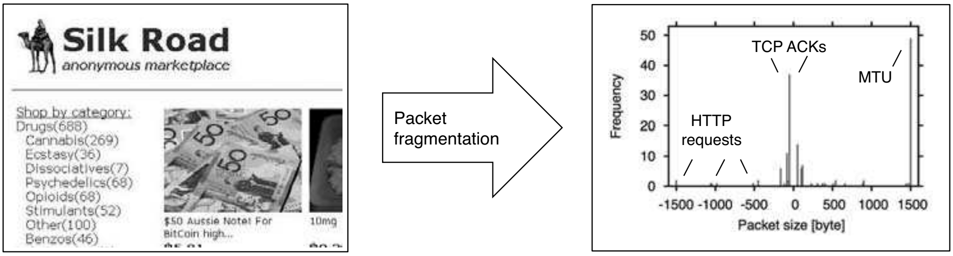

Figure 2: Characteristic distribution of IP packet sizes due to packet fragmentation [HWF09]

In [LL06] Liberatore and Levine, who focus on IP packet sizes and packet direction, have shown for a sample of 1000 sites that observers can infer the websites downloaded via OpenSSH tunnels with a Naïve Bayes classifier or the Jaccard index. Their best technique scores a website identification accuracy of 73 %. In a similar experiment inspired by their work we have improved the detection accuracy to 97 % by applying suitable transformations to the raw data [HWF09]. Our results indicate that fingerprinting is not limited to OpenSSH tunnels, but also effective for other popular techniques (e. g., IPsec and OpenVPN). Based on our dataset, the work of Panchenko et al. is one of the first to demonstrate that fingerprinting may also work in the Tor network [PNZE11]. Since then this problem (also termed "encrypted page identification" [XRYX13]) has received much attention (cf. [DCRS12] and [WG13] for an overview).

3.2 DNS-based Website Fingerprinting

DNS-based website fingerprinting serves a different purpose. It can be useful for forensic investigators that need to determine the **search terms** of a suspect whose traffic has been seized using lawful interception. Until recently, traffic to web search engines was not encrypted. However, in 2013 the leading search provider Google enabled SSL for all queries by default, which obscures the entered query terms in the intercepted traffic. The only remaining option for investigators to obtain the queries of a suspect consists in forcing Google to disclose them, which may be infeasible due to legal cross-border issues.

However, under certain circumstances DNS-based website fingerprinting can be used to infer the search terms entered into search engines, even when requests to the search engine are encrypted with HTTPS. This possibility was first studied by Krishnan and Monrose [KM10, KM11], who present results for the Firefox and Chrome browsers with the Google search engine. DNS-based fingerprinting exploits the fact that some browsers not only issue the DNS queries required to load the content embedded in the currently rendered site, but they also issue DNS queries for the links (`<a>` tags) that are contained in a site. The rationale for this *DNS pre-resolution* is to speed up the download of the next site when the user clicks a link. In addition, the Chrome browser tries to guess the links the user will most likely click on, in order to download the corresponsing sites in the background (DNS queries due to "site prefetching").

Table 1: Inferring Search Terms via DNS-based Website Fingerprinting on Google (HTTPS enabled)

Browser	Link pre-resolution	Site prefetching	Speculative resolution
Firefox 25.0.1	no (HTTP only)	no	yes
Chrome 31.0.1650.57	no (HTTP only)	yes	yes
Safari 6.1 (8537.71)	no (HTTP only)	no	no

Krishnan and Monrose demonstrate that DNS-based fingerprinting works especially well for search engines, whose primary purpose is to display lists of links for the relevant sites and for ads. For many search queries the set of sites displayed on the first results page is a distinctive and stable feature. In order to infer the search terms by DNS-based fingerprinting, the investigator would have to issue the interesting queries himself and establish a database containing the observed DNS queries. Inferring the search terms may also be feasible if the database is incomplete, because in some cases the DNS queries issued by the browser *literally repeat* the search terms (e. g., searching for “gun powder” may induce the search engine to insert an ad for the site *gunpowder.buycheaper.biz*).

Moreover, some browsers, i. e., Firefox, Chrome, and Safari, allow the user to enter search terms directly into the address bar. This comfortable feature can also cause the search terms to be leaked via DNS queries, when the browser cannot rule out that an entered keyword *is* the hostname of a web server (*speculative resolution*).

The effectiveness of DNS-based website fingerprinting is subject to technical **limitations**. First of all, search engines do not necessarily display the same search results to all users, i. e., it may be difficult to match search queries based on the observed DNS queries. Moreover, in recent versions browser manufacturers have turned off pre-resolution for HTTPS sites for privacy reasons, i. e., the pre-resolution based technique is not effective any more. However, in our own experiments (Nov. 2013) we could still observe DNS queries due to speculative resolution and due to site prefetching (cf. Table 1).

4 Case Study 2: Device-based Ascription of Activities

In the second case study we showcase how fingerprinting techniques can be leveraged to address Challenge C1: “who did it?”. We assume that law enforcement has managed to gain authority over an underground marketplace that specializes in illegal drugs. Instead of taking the site down, investigators intercept and retain all incoming traffic on the web servers for a while in order to obtain new leads for tracking down the dealers. To this end they request customer subscription data from the respective ISPs for the observed source IP addresses. Of particular interest is one of the more active customers, Carol, who shares a broadband Internet connection with her unsuspecting flat mate Alice. Based on the intercepted traffic, investigators *do* know that the criminal activity originated from their apartment, however they *do not* know who of the two flat mates is involved. Police forces raid the apartment and seize the laptops of Alice and Carol, who deny any involvement in criminal activities. The forensic analysis of the hard disks does not produce any evidence,

Table 2: Device fingerprinting with individual (I) and class (C) characteristics

	Trait	Involvement	Source
I	Skew of real-time clock [KBC05]	active probing	Manuf.
C	TCP stack of operating system [CL94]	active probing	OS
C	TCP stack of operating system [Bev04]	passive logging	OS
C	JS benchmark [MBYS11], HTML 5 rendering [MS12]	active website	HW/SW
I	Browser fingerprints [Eck10]	active website	SW
C	TCP flow characteristics during surfing [YHMR09]	passive logging	SW

either: Alice, whose laptop is secured using FDE, denies to disclose her password, and on Carol's machine no traces of activity can be found, because she always entered the "private browsing" mode before engaging in unlawful endeavors.

However, in this scenario investigators may still be able to determine the computer from which the criminal activity originated, because different devices may exhibit different behavior on the network, resulting in a characteristic *device fingerprint*. Device fingerprints are present due to tolerances in the *manufacturing process* and differences in *hardware and software implementations*. They can either capture **class characteristics**, which are emitted by all devices that share the same specification, or **individual characteristics**, which allow to uniquely identify a single device [Cas11, 17].

In order to leverage device fingerprints to ascribe the criminal activities to one of the two computers in question, investigators have to extract the browser and operating system fingerprints from the incoming requests while they intercept the traffic of the underground marketplace. After the raid investigators can compare the fingerprints they observed for the requests that led them to Alice and Carol (e. g., "Firefox browser on a Windows PC") with the hard- and software configuration of the two suspects. If only one of the two computers is found to match the specification indicated by the fingerprint, this can serve as evidence for involvement.

In the following we will briefly survey the most relevant device fingerprinting approaches before we provide results from our own experiments with DNS-based fingerprinting.

4.1 Existing Device Fingerprinting Techniques

Table 2 showcases the landscape of fingerprinting techniques that could be leveraged by investigators that want to associate a specific device with certain actions. **Active device fingerprinting** techniques provoke a device to emit its fingerprint in response to specially crafted probing packets or by explicitly reading out certain properties. Comer and Lin were among the first to observe that the TCP stacks of operating systems respond differently when they receive spurious packets and that they also vary in terms of timeouts and retransmission behavior [CL94], which can serve as a class characteristic. Kohno et al. discovered that multiple desktop machines can be remotely differentiated due to skew in their real-time clocks [KBC05]. While this constitutes a powerful individual characteristic, it cannot be observed passively in a reliable manner. The trend for tighter integration of browsers with the operating system has also created new fingerprinting opportunities that emit class characteristics: Mowery et al. show that hardware/software configurations can be either differentiated with a crafted JavaScript benchmark embedded in a website [MBYS11] or by analyzing the anti-aliasing artifacts of text that is rendered using the *canvas* tag in HTML 5 [MS12]. Summarizing the results of the "EFF Panopticlick" experiment Eckersly illustrates that individual browsers – and sometimes even devices – can be re-identified with high probability by a website that actively enumerates the list of installed browser plug-ins and system fonts [Eck10].

A limitation of active techniques is the fact that they require *active involvement* on the part of the investigator. In contrast, **passive device fingerprinting** needs only traits that are "voluntarily" provided by a device and thus collected easily. A well-known property of this kind is the "user agent" string in HTTP requests, which explicitly states browser and operating system. The information provided there can be forged easily, though, i. e., its probative value is questionable. More reliable are unavoidably exhibited implicit behavioral traits, e. g., the class characteristics that allow for passive operating system detection with the tool *p0f* (http://lcamtuf.coredump.cx/p0f3/) based on characteristic implementations of the TCP/IP stack [Bev04]. Similar techniques can be applied to differentiate various browser types [YHMR09].

4.2 DNS-based Device Fingerprinting

DNS-based device fingerprinting is a passive technique that exploits the fact that operating systems and browsers can be identified via the hostnames they resolve during regular background activities, e. g., time synchronization and polling for software updates [MYK13]. The application of this technique requires LEAs to access the DNS queries of a suspect, either by way of lawful interception or by legally forcing the ISP to record and disclose the DNS traffic of a suspect. Investigators can cross-correlate the class characteristics inferred by DNS-based fingerprinting with other evidence. For instance, they can detect faked user agent strings, if there are no DNS queries that conform to the stipulated operating system or browser. The results of our experiments, in which we observed the background traffic of various browsers on multiple platforms, indicate that DNS-based device fingerprinting is feasible for common configurations (cf. Tables 3 and 4 in the Appendix).

5 Case Study 3: Behavior-based Ascription of Activities

In the third case study we assume a similar scenario as in Case Study 2, however, this time investigators do not succeed in extracting device fingerprints from the requests issued to the underground marketplace. Nevertheless, they may still be able to solve Challenge C1 ("who did it?") by applying *behavior-based fingerprinting* techniques, which allow to link multiple surfing sessions of the same user as well as to distinguish sessions of different users solely based on (alleged) characteristic web usage patterns.

In this case we assume that investigators have access to Alice's and Carol's traffic via lawful interception. Starting off with an anonymous behavioral fingerprint of a surfing session that contains an incriminating action (the "initial fingerprint"), the objective of the investigator consists in identifying additional surfing sessions of the (still not identifiable) suspect that match the initial fingerprint ("suspect's sessions"). Note that it is not required that all sessions linked to the initial fingerprint contain instances of incriminating behavior; they will be useful for investigators even if they contain only innocuous behavior.

Once a sufficiently large number of suspect's sessions has been acquired, there are two conditions that may indicate to investigators that Carol, and not Alice, is the offender: either an *intersection attack* or *unintentional identity disclosure*. For an intersection attack the investigators analyze the temporal usage patterns that emerge from the suspect's sessions in order to infer at what times the offender was online. This information can be used in concert with classical investigative techniques, such as physical observation or questioning, to infer the identity of the offender, e. g., because Alice may have an alibi for some of the points in time. Intersection attacks have been applied with success in various contexts [BL02, KAP02, KAPR06, PGT12]. The second condition occurs when Carol discloses her identity in one of her sessions that have been linked to the suspect's sessions via the initial fingerprint, e. g., by logging on to a shopping portal or mail account with her personal credentials. If the credentials are unavailable, e. g., due to encryption with HTTPS, investigators can still compel the respective service provider to disclose the identity of the account based on the observed date and time of the log-in.

Note that investigators will only be able to extract the "initial fingerprint" if Carol engages at least one more time in incriminating activity (e. g., by visiting the underground marketplace) after lawful interception of traffic has commenced. In some cases, such as our drug trafficking scenario, this is a reasonable assumption.

Previous research suggests that behavior-based ascription of activities is possible in practice, i. e., multiple sessions of the same user can be linked due to characteristic website visitation patterns. The **behavioral fingerprint** of a user consists of the set of visited websites, which are the result of the distinct set of his personal interests. The potential of behavioral fingerprints for differentiating users can already be appreciated by cursory visual inspection of individual sessions (see Fig. 3).

Yang shows how machine learning techniques used for *association rule mining* can be leveraged to extract and match such behavioral patterns [Yan10]. In a controlled setting with up to 100 concurrent users the predictive accuracy of her techniques reaches up to 87 % when profiles are built using 100 training sessions per user. Accuracy drops to only 62 % if only a single training session is available.

However, according to our own research, which has been published in detail in [BHF12, HBF13], extracting and matching behavioral fingerprints *is* feasible even when only a single session is available. We reach this conclusion based on our experiments with an anonymized dataset that contains the DNS queries issued by a set of 3862 enrolled students over the course of two months. For each user we extract a behavioral fingerprint (queried DNS hostnames and access frequencies) from one of his sessions, which is then used to identify the remaining sessions of the same user. With a Multinomial Naïve Bayes

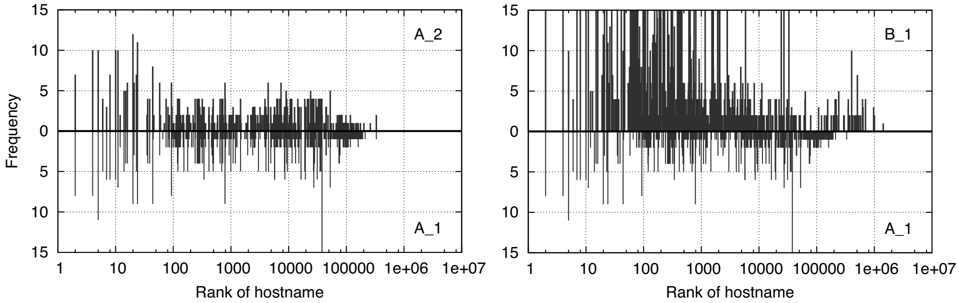

Figure 3: Higher visual similarity between website visitation behavior in two sessions of the same user (left-hand side) in comparison to sessions of different users (right-hand side)

classifier [MRS08, 258] we obtain cross-validated precision and recall values beyond 70 % for sessions of up to 24 hours and 3000 concurrent users. Further experiments have shown that accuracy of linking improves for smaller sets of users, e. g., it reaches 90 % for 100 concurrent users.

6 Limitations and Concerns

While we argue that fingerprinting techniques may be a suitable tool to improve the forensic tool-chain we stress that they are also subject to considerable limitations and thus have to be adopted with care in criminal investigations.

Firstly, the probative value of the results obtained by fingerprinting is often unclear. On the one hand, techniques that involve machine learning algorithms may suffer from poor explainability, when the investigator cannot fully explain how the classifier reached a decision. On the other hand, high accuracy values obtained in scientific papers may convey a certain level of confidence and objectivity, but usually they are only meant to give an indication of the performance of a technique for a specific closed-world scenario in a controlled setting, i. e., it is unknown how they perform “in the wild”. Standardized corpora and strict evaluation methodologies have to be established to increase the confidence of the results.

Secondly, once law enforcement agencies have learned to appreciate the qualities of fingerprinting techniques for the purpose of criminal investigations, security policy makers may move forward and call for mandatory online dragnet investigations based on fingerprints, which could even be argued to be “privacy-preserving” – because only abstract fingerprints and not the actual content are screened, after all. Nevertheless such measures would be the equivalent of pre-emptive blanket surveillance. Therefore, instead of only focusing on the improvement of fingerprinting techniques researchers should also work on usable countermeasures to keep the balance.

7 Conclusions

In this paper, we have explored the adoption of state of the art fingerprinting techniques from information security research to the field of network forensics. Our analysis suggests that fingerprinting can be used to establish associations between criminal activities and their perpetrators as well as to find corroborative evidence for criminal activities even if only encrypted traffic can be observed. In two of our three case studies (cf. Sects. 3 and 5) we have shown that evidence can be obtained with fingerprinting that would otherwise require measures of questionable proportionality, such as data retention or remote search via state malware. In our third scenario (cf. Sect. 4) no client-side lawful interception is required at all.

These findings suggest that, instead of deploying heavy artillery, the adoption of fingerprinting techniques may lead the way to more target-oriented and incident-related investigations in the future and might also become a viable enhancement for current investigative methods. However, several challenges must be met before fingerprinting will become dependable and practical for these scenarios.

Acknowledgements

This paper was inspired by talks and discussions at Dagstuhl Seminar 13482 on Forensic Computing. The authors are grateful to all participants for their insightful ideas and fruitful feedback. We also thank our colleague Christoph Gerber for inspiring our discussions of the case studies.

References

[Bev04] R. Beverly. A Robust Classifier for Passive TCP/IP Fingerprinting. In C. Barakat and I. Pratt, editors, *PAM*, volume 3015 of *LNCS*, pages 158–167. Springer, 2004.

[BFGI11] K. Boda, A. M. Földes, G. G. Gulyás, and S. Imre. User Tracking on the Web via Cross-Browser Fingerprinting. In P. Laud, editor, *NordSec*, volume 7161 of *LNCS*, pages 31–46. Springer, 2011.

[BHF12] C. Banse, D. Herrmann, and H. Federrath. Tracking Users on the Internet with Behavioral Patterns: Evaluation of Its Practical Feasibility. In D. Gritzalis, S. Furnell, and M. Theoharidou, editors, *SEC*, volume 376 of *IFIP Advances in Information and Communication Technology*, pages 235–248. Springer, 2012.

[BL02] O. Berthold and H. Langos. Dummy Traffic Against Long Term Intersection Attacks. In Dingledine and Syverson [DS03], pages 110–128.

[BLJL05] G. D. Bissias, M. Liberatore, D. J. Jensen, and B. N. Levine. Privacy Vulnerabilities in Encrypted HTTP Streams. In G. Danezis and D. Martin, editors, *Privacy Enhancing Technologies*, volume 3856 of *LNCS*, pages 1–11. Springer, 2005.

[Cas04] E. Casey. Network traffic as a source of evidence: tool strengths, weaknesses, and future needs. *Digital Investigation*, 1(1):28–43, 2004.

[Cas09] E. Casey. *Handbook of Digital Forensics and Investigation*. Academic Press, 2009.

[Cas11] E. Casey. *Digital Evidence and Computer Crime – Forensic Science, Computers and the Internet*. Academic Press, 3rd edition, 2011.

[CFGS11] E. Casey, G. Fellows, M. Geiger, and G. Stellatos. The growing impact of full disk encryption on digital forensics. *Digital Investigation*, 8(2):129–134, 2011.

[CL94] D. Comer and J. C. Lin. Probing TCP Implementations. In *USENIX Summer*, pages 245–255, 1994.

[CL10] D. D. Clark and S. Landau. The Problem Isn't Attribution: It's Multi-stage Attacks. In *Proceedings of the Re-Architecting the Internet Workshop*, ReARCH '10, pages 11:1–11:6, New York, NY, USA, 2010. ACM.

[DCGS09] M. Dusi, M. Crotti, F. Gringoli, and L. Salgarelli. Tunnel Hunter: Detecting Application-Layer Tunnels with Statistical Fingerprinting. *Computer Networks*, 53(1):81–97, 2009.

[DCRS12] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton. Peek-a-Boo, I Still See You: Why Efficient Traffic Analysis Countermeasures Fail. In *IEEE Symposium on Security and Privacy*, pages 332–346. IEEE, 2012.

[DH12] S. Davidoff and J. Ham. *Network Forensics: Tracking Hackers through Cyberspace*. Pearson Education, 2012.

[DMS04] R. Dingledine, N. Mathewson, and P. F. Syverson. Tor: The Second-Generation Onion Router. In *USENIX Security Symposium*, pages 303–320. USENIX, 2004.

[DS03] R. Dingledine and P. F. Syverson, editors. *Privacy Enhancing Technologies, Second International Workshop, PET 2002, San Francisco, CA, USA, April 14-15, 2002, Revised Papers*, volume 2482 of *LNCS*. Springer, 2003.

[Eck10] P. Eckersley. How Unique Is Your Web Browser? In M. J. Atallah and N. J. Hopper, editors, *Privacy Enhancing Technologies*, volume 6205 of *LNCS*, pages 1–18. Springer, 2010.

[Fau80] H. Faulds. On the Skin-Furrows of the Hand. *Nature*, 22(574):605, 1880.

[Gal92] F. Galton. *Finger Prints*. Macmillan and Co., London, 1892.

[HB11] A. Houmansadr and N. Borisov. SWIRL: A Scalable Watermark to Detect Correlated Network Flows. In *NDSS*. The Internet Society, 2011.

[HB13] A. Houmansadr and N. Borisov. The Need for Flow Fingerprints to Link Correlated Network Flows. In E. De Cristofaro and M. Wright, editors, *Privacy Enhancing Technologies*, volume 7981 of *LNCS*, pages 205–224. Springer, 2013.

[HBF13] D. Herrmann, C. Banse, and H. Federrath. Behavior-based tracking: Exploiting characteristic patterns in DNS traffic. *Computers & Security*, 39A:17–33, November 2013.

[HBK04] J. Hall, M. Barbeau, and E. Kranakis. Enhancing Intrusion Detection in Wireless Networks Using Radio Frequency Fingerprinting. In M. H. Hamza, editor, *Communications, Internet, and Information Technology*, pages 201–206. IASTED/ACTA Press, 2004.

[Hin02] A. Hintz. Fingerprinting Websites Using Traffic Analysis. In Dingledine and Syverson [DS03], pages 171–178.

[HKB12] A. Houmansadr, N. Kiyavash, and N. Borisov. Non-blind Watermarking of Network Flows. *CoRR*, abs/1203.2273, 2012.

[HWF09] D. Herrmann, R. Wendolsky, and H. Federrath. Website Fingerprinting: Attacking Popular Privacy Enhancing Technologies with the Multinomial Naïve-Bayes Classifier. In R. Sion and D. Song, editors, *CCSW*, pages 31–42. ACM, 2009.

[JRP04] A. K. Jain, A. Ross, and S. Prabhakar. An Introduction to Biometric Recognition. *IEEE Trans. Circuits Syst. Video Techn.*, 14(1):4–20, 2004.

[JWJ+13] A. Johnson, C. Wacek, R. Jansen, M. Sherr, and P. F. Syverson. Users Get Routed: Traffic Correlation on Tor by Realistic Adversaries. In A. Sadeghi, V. D. Gligor, and M. Yung, editors, *ACM Conference on Computer and Communications Security*, pages 337–348. ACM, 2013.

[KAP02] D. Kesdogan, D. Agrawal, and S. Penz. Limits of Anonymity in Open Environments. In F. A. P. Petitcolas, editor, *Information Hiding*, volume 2578 of *LNCS*, pages 53–69. Springer, 2002.

[KAPR06] D. Kesdogan, D. Agrawal, D. V. Pham, and D. Rautenbach. Fundamental Limits on the Anonymity Provided by the MIX Technique. In *IEEE Symposium on Security and Privacy*, pages 86–99. IEEE, 2006.

[KBC05] T. Kohno, A. Broido, and K. C. Claffy. Remote Physical Device Fingerprinting. In *IEEE Symposium on Security and Privacy*, pages 211–225. IEEE, 2005.

[KM10] S. Krishnan and F. Monrose. DNS Prefetching and Its Privacy Implications: When Good Things Go Bad. In *Proceedings of the 3rd USENIX Conference on Large-scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More*, LEET' 10, Berkeley, CA, USA, 2010. USENIX Association.

[KM11] S. Krishnan and F. Monrose. An Empirical Study of the Performance, Security and Privacy Implications of Domain Name Prefetching. In *DSN*, pages 61–72. IEEE, 2011.

[Lac67] E. A. Lacy. Radar Signature Analysis. *Electronics World*, 77(2):23–26, February 1967.

[LL06] M. Liberatore and B. N. Levine. Inferring the Source of Encrypted HTTP Connections. In A. Juels, R. N. Wright, and S. De Capitani di Vimercati, editors, *ACM Conference on Computer and Communications Security*, pages 255–263. ACM, 2006.

[MBYS11] K. Mowery, D. Bogenreif, S. Yilek, and H. Shacham. Fingerprinting Information in JavaScript Implementations. In *Proceedings of Web 2.0 Security and Privacy 2011 (W2SP)*, San Franciso, May 2011.

[MRS08] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008.

[MS12] K. Mowery and H. Shacham. Pixel Perfect: Fingerprinting Canvas in HTML5. In M. Fredrikson, editor, *Proceedings of W2SP 2012*. IEEE, May 2012.

[MYK13] T. Matsunaka, A. Yamada, and A. Kubota. Passive OS Fingerprinting by DNS Traffic Analysis. In L. Barolli, F. Xhafa, M. Takizawa, T. Enokido, and H. Hsu, editors, *AINA*, pages 243–250. IEEE, 2013.

[OLL11] J. Oh, S. Lee, and S. Lee. Advanced Evidence Collection and Analysis of Web Browser Activity. *Digital Investigation*, 8:62–70, August 2011.

[Per09] M. T. Pereira. Forensic Analysis of the Firefox 3 Internet History and Recovery of Deleted SQLite Records. *Digital Investigation*, 5(3-4):93–103, 2009.

[PGT12] F. Pérez-González and C. Troncoso. Understanding Statistical Disclosure: A Least Squares Approach. In S. Fischer-Hübner and M. Wright, editors, *Privacy Enhancing Technologies*, volume 7384 of *LNCS*, pages 38–57. Springer, 2012.

[PNZE11] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel. Website Fingerprinting in Onion Routing Based Anonymization Networks. In Y. Chen and J. Vaidya, editors, *WPES*, pages 103–114. ACM, 2011.

[Ray00] J. Raymond. Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems. In H. Federrath, editor, *Workshop on Design Issues in Anonymity and Unobservability*, volume 2009 of *LNCS*, pages 10–29. Springer, 2000.

[SSW$^+$02] Q. Sun, D. R. Simon, Y. Wang, W. Russell, V. N. Padmanabhan, and L. Qiu. Statistical Identification of Encrypted Web Browsing Traffic. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, Washington, DC, USA, 2002. IEEE.

[TDH03] K. I. Talbot, Paul R. Duley, and M. H. Hyatt. Specific Emitter Identification and Verification. *Technology Review Journal*, 2003.

[WCJ07] X. Wang, S. Chen, and S. Jajodia. Network Flow Watermarking Attack on Low-Latency Anonymous Communication Systems. In *IEEE Symposium on Security and Privacy*, pages 116–130. IEEE, 2007.

[WG13] T. Wang and I. Goldberg. Improved Website Fingerprinting on Tor. In A. Sadeghi and S. Foresti, editors, *WPES*, pages 201–212. ACM, 2013.

[WRW02] X. Wang, D. S. Reeves, and S. F. Wu. Inter-Packet Delay Based Correlation for Tracing Encrypted Connections through Stepping Stones. In D. Gollmann, G. Karjoth, and M. Waidner, editors, *ESORICS*, volume 2502 of *LNCS*, pages 244–263. Springer, 2002.

[XGMT09] L. Xiao, L. J. Greenstein, N. B. Mandayam, and W. Trappe. Fingerprints in the Ether: Using the Physical Layer for Wireless Authentication. *CoRR*, abs/0907.4877, 2009.

[XRYX13] W. Xia, Y. Ren, Z. Yuan, and Y. Xue. TCPI: A Novel Method of Encrypted Page Identification. In *1st International Workshop on Cloud Computing and Information Security (CCIS 2013).* Atlantis Press, 2013.

[Yan10] Y. Yang. Web user behavioral profiling for user identification. *Decision Support Systems*, 49:261–271, 2010.

[YE00] K. Yoda and H. Etoh. Finding a Connection Chain for Tracing Intruders. In F. Cuppens, Y. Deswarte, D. Gollmann, and M. Waidner, editors, *ESORICS*, volume 1895 of *LNCS*, pages 191–205. Springer, 2000.

[YHMR09] T. Yen, X. Huang, F. Monrose, and M. K. Reiter. Browser Fingerprinting from Coarse Traffic Summaries: Techniques and Implications. In U. Flegel and D. Bruschi, editors, *DIMVA*, volume 5587 of *LNCS*, pages 157–175. Springer, 2009.

[ZP00] Y. Zhang and V. Paxson. Detecting Stepping Stones. In *Proceedings of the 9th Conference on USENIX Security Symposium*, pages 13–13, Berkeley, CA, USA, 2000. USENIX Association.

A Results for DNS-based Device Fingerprinting

In order to evaluate the feasibility of DNS-based device fingerprinting we recorded the hostnames that are resolved autonomously in the background by commonly used web browsers and desktop operating systems. To this end we have performed experiments with MS Internet Explorer 8 and 10, Mozilla Firefox 25, Apple Safari 6.1, Google Chrome 31.0.1650.57 that have been executed on all supported platforms from the set of MS Windows XP SP3, MS Windows 7 SP1, MS Windows 8, Apple MacOS 10.8.5, Ubuntu 12.04, CentOS 6.3, and openSUSE 12.2.

According to the results for the web browsers (cf. Table 3), the studied versions of the Firefox, Internet Explorer 10 and Chrome browsers can be detected based on autonomously issued queries, while Safari and Internet Explorer 8 cannot be detected precisely. Moreover, as shown in Table 4, regardless of the type of browser used, the majority of the operating systems can be detected due to characteristic DNS queries; only Windows XP and openSUSE do not issue characteristic queries in the background.

Table 3: DNS-based fingerprinting of common desktop web browsers; *emphasized hostnames* were observed for a single browser only and don't offer any web content intended for human users.

Browser	Hostnames
Firefox	*aus3.mozilla.org download.cdn.mozilla.net fhr.data.mozilla.com services.addons.mozilla.org versioncheck-bg.addons.mozilla.org versioncheck.addons.mozilla.org* addons.mozilla.org cache.pack.google.com download.mozilla.org [x].pack.google.com safebrowsing-cache.google.com safebrowsing.clients.google.com tools.google.com
IE 8	urs.microsoft.com
IE 10	*iecvlist.microsoft.com t.urs.microsoft.com* ctldl.windowsupdate.com mscrl.microsoft.com urs.microsoft.com www.bing.com
Safari	apis.google.com clients.l.google.com clients1.google.com safebrowsing-cache.google.com safebrowsing.clients.google.com ssl.gstatic.com www.google.com www.google.de www.gstatic.com
Chrome	*safebrowsing.google.com translate.googleapis.com [xxxxxxxxxx].[domain]* apis.google.com cache.pack.google.com clients[x].google.com [x].pack.google.com safebrowsing-cache.google.com safebrowsing.clients.google.com ssl.gstatic.com tools.google.com www.google.com www.google.de www.gstatic.com

Symbols: [x]: placeholder for varying numbers/strings; **[domain]:** placeholder for configured search domain.

Table 4: DNS-based fingerprinting of desktop operating systems; *emphasized hostnames* were observed for a single operating system only and don't offer any web content intended for human users.

Windows XP	
U	update.microsoft.com download.windowsupdate.com
T	time.windows.com
Windows 7	
U	*au.download.windowsupdate.com* update.microsoft.com download.windowsupdate.com ctldl.windowsupdate.com
T	time.windows.com
P	*watson.microsoft.com*
N	*ipv6.msftncsi.com* teredo.ipv6.microsoft.com www.msftncsi.com dns.msftncsi.com isatap.[domain] wpad.[domain]
M	*gadgets.live.com weather.service.msn.com money.service.msn.com*
Windows 8	
U	*au.v4.download.windowsupdate.com ds.download.windowsupdate.com* ctldl.windowsupdate.com *bg.v4.emdl.ws.microsoft.com fe[x].update.microsoft.com fe[x].ws.microsoft.com definitionupdates.microsoft.com spynet2.microsoft.com*
T	time.windows.com
P	*watson.telemetry.microsoft.com sqm.telemetry.microsoft.com*
N	teredo.ipv6.microsoft.com www.msftncsi.com dns.msftncsi.com isatap.[domain] wpad.[domain]
C	mscrl.microsoft.com crl.globalsign.net ocsp.verisign.com evsecure-ocsp.verisign.com evintl-ocsp.verisign.com
L	*clientconfig.passport.net* login.live.com go.microsoft.com
M	*ssw.live.com client.wns.windows.com appexbingfinance.trafficmanager.net appexbingweather.trafficmanager.net appexsports.trafficmanager.net appexdb[x].stb.s-msn.com de-de.appex-rf.msn.com finance.services.appex.bing.com financeweur[x].blob.appex.bing.com weather.tile.appex.bing.com*
MacOS X	
U	*swscan.apple.com swdist.apple.com swcdnlocator.apple.com su.itunes.apple.com* swcdn.apple.com r.mzstatic.com s.mzstatic.com metrics.mzstatic.com
T	*time.euro.apple.com*
P	*radarsubmissions.apple.com internalcheck.apple.com* securemetrics.apple.com
C	ocsp.apple.com ocsp.entrust.net ocsp.verisign.com EVIntl-ocsp.verisign.com EVSecure-ocsp.verisign.com SVRSecure-G3-aia.verisign.com
L	*identity.apple.com configuration.apple.com init.ess.apple.com init-p[x]md.apple.com* albert.apple.com
M	*p[x]-contacts.icloud.com p[x]-caldav.icloud.com p[x]-imap.mail.me.com [x].guzzoni-apple.com.akadns.net ax.init.itunes.apple.com a[x].phobos.apple.com keyvalueservice.icloud.com* [x]-courier.push.apple.com itunes.apple.com
Ubuntu	
U	*changelogs.ubuntu.com*
T	*ntp.ubuntu.com geoip.ubuntu.com*
P	*daisy.ubuntu.com*
L	*_https._tcp.fs.one.ubuntu.com fs-[x].one.ubuntu.com* one.ubuntu.com
CentOS	
U	*mirrorlist.centos.org*
T	*[x].centos.pool.ntp.org*
openSUSE	
U	download.opensuse.org opensuse-community.org

Symbols: [U]: polling for updates; **[T]:** time synchronization; **[P]:** problem reporting; **[N]:** discovery of network connectivity; **[C]:** certificate validation; **[L]:** activities right after log-on; **[M]:** miscellaneous background activity; **[x]:** placeholder for varying numbers/strings; **[domain]:** placeholder for configured search domain.