FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

New challenges in digital forensics: online storage and anonymous communication

PhD THESIS

submitted in partial fulfillment of the requirements for the degree of

Doctor of Technical Sciences

by

Martin Mulazzani
Registration Number 0225055

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Privatdoz. Dipl.-Ing. Mag.rer.soc.oec. Dr.techn. Edgar Weippl

The dissertation has been reviewed by:

(Univ.-Prof. Dipl.-Math. Dr. Stefanie Rinderle-Ma)

(Univ.-Prof. Dipl.-Ing. DDr. Gerald Quirchmayr)

Vienna, 30.01.2014

(Martin Mulazzani)

Declaration of Authorship

Martin Mulazzani
Kreutzergasse 5, 3400 Klosterneuburg

I hereby declare that I have written this Doctoral Thesis independently, that I have completely specified the utilized sources and resources and that I have definitely marked all parts of the work - including tables, maps and figures - which belong to other works or to the internet, literally or extracted, by referencing the source as borrowed.

(Vienna, 30.01.2014) (Martin Mulazzani)

Acknowledgements

I'm very grateful to my advisor Edgar Weippl, who supervised me throughout my studies and always supported me in my own research ideas. Even in the busiest times, feedback and support were always available and the discussion of ideas helped me a lot. I would also like to thank all my colleagues at SBA Research - in particular Markus Huber, Sebastian Schrittwieser and Sebastian Neuner, whom I have the honor to work with on a regular basis. Besides discussing ideas and working together numerous late-nights close to paper deadlines, colleagues like them cannot be taken for granted and are among the reasons why I enjoyed this period in my life as much as I did. I'm also thankful for the all the opportunities I received at SBA Research during these years, in particular teaching skillful students and working with them to develop their own research ideas. I'm also grateful that I did get the opportunity to study and work two semesters at Purdue University, and I would like to thank Prof. Elisa Bertino as well as Prof. Christina Nita-Rotaru for their supervision and mentoring. Furthermore, this work would not have been possible without funding from COMET K1 and project number 825747 (INFORM) by the FFG Austrian Research Agency.

I'm grateful for the support from my family, my parents Eva and Marco Mulazzani as well as my brothers Matthias, Max and Michael, and above all Kathi and our wonderful children Ferdinand and Liselotte for their continuous love and help during the creation of my thesis. Without you I would not be where I am today.

Martin Mulazzani
Vienna, 30.01.2014

Abstract

This thesis is based on seven publications related to the area of digital forensics which were published at conferences or in journals by ACM, IEEE, USENIX and IFIP. Digital forensics as research field has received increasing attention in recent years, as more and more crimes are committed exclusively or with the involvement of computers. At the same time, new challenges emerge constantly, e.g. the prevalent use of encryption, mobile devices of various nature, online cloud storage services and readily available tools that facilitate counter-forensics. In particular, this thesis tries to mitigate current challenges for digital forensics in the areas of online data storage and anonymous communication.

Regarding anonymous communication we analyzed the well-known online anonymity tool Tor, which employs onion routing and is expected to be used by hundreds of thousands of users every day: firstly how it is used, and secondly what can be learnt from the publicly available server information. We were able to show that the majority of users are not employing Tor as recommended by the Tor community, and we found many information leaks that can endanger the users' anonymity. We also studied how the underlying infrastructure, which is run by volunteers, can be monitored to provide useful metrics of interest. We furthermore derived and implemented a new attack on online storage systems abusing client-side data deduplication and analyzed how it can be used to thwart digital forensic investigations which in turn can be used for forensic investigations. We showed its feasibility on Dropbox, one of the largest cloud storage providers with more than 200 million users worldwide at the time of writing this thesis. We quantified slack space on numerous Windows systems, assessed it's stability over time regarding system updates and found that up to 100 megabytes of slack space are readily available in files of the operating system. We furthermore implemented a digital alibi framework with a social interaction component which in our opinion can be easily overlooked in forensic analysis as conducted today. Finally we analyzed browser artifacts and how they can be used for browser fingerprinting. We then used browser fingerprinting to enhance HTTP session security by binding the session on the server to specifics of the particular browser used.

Kurzfassung

Diese Dissertation baut auf sieben Arbeiten auf, die auf Konferenzen und in Journalen von ACM, IEEE, USENIX und der IFIP veröffentlicht wurden. Digitale Forensik als Forschungsdisziplin hat sich in den letzten Jahren mehr und mehr etabliert, da kriminelle Handlungen mittlerweile ausschließlich mit oder unter Zuhilfenahme von Computern begangen werden. Gleichzeitig werden durch die Verbreitung von starker Verschlüsselung, einer Vielzahl an neuen mobilen Geräten und das stetig steigende Datenvolumen neue Herausforderungen an die digitale Forensik gestellt. Diese Arbeit beschäftigt sich im Speziellen mit den Problemen der digitalen Forensik hinsichtlich Speicherdienste im Internet und anonymer Kommunikation.

Im Bereich der anonymen Kommunikation untersucht diese Arbeit Tor, ein sehr weit verbreiteter Anonymisierungsdienst im Internet. Es konnte belegt werden, dass Tor meist nicht wie empfohlen verwendet wird und die Gefahr einer kompletten (unbeabsichtigten) Deanonymisierung für die Anwender hoch ist. Wir haben verschiedene Metriken für die dem Tor Netzwerk zugrundeliegende Infrastruktur mit derzeit ca. 5.000 Knoten erstellt, da diese von Freiwilligen betrieben und nicht zentral kontrolliert wird. Im Bereich der digitalen Forensik haben wir eine neue Angriffsmethode auf Internet-Speicherdienste entwickelt und implementiert. Dieser Angriff nützt die Datendeduplizierung auf der Anwenderseite aus, um einen Angreifer unberechtigten Zugriff auf Daten zu ermöglichen. Die Anwendbarkeit unseres Angriffs wurde anhand von Dropbox belegt, einem der größten Speicherdienste mit derzeit mehr als 200 Millionen Anwendern. Wir haben weiters die Gesamtspeicherkapazität von Fragmentierungsartefakten (“slack space”) von Microsoft Windows vermessen und über einen längeren Zeitraum die Stabilität in Bezug auf Systemupdates ermittelt. Zusätzlich haben wir ein Framework implementiert, das die Erzeugung eines digitalen Alibis ermöglicht. Unser Ansatz beinhaltete eine soziale Kommunikationskomponente, die eine forensische Untersuchung täuschen könnte. Im Bereich der sicheren Online-Kommunikation haben wir Webbrowser untersucht und neuartige Identifizierungsmöglichkeiten entdeckt. Auf diesen Ergebnissen aufbauend erhöhten wir die Sicherheit von Online-Sitzungen, indem die Sitzung Server-seitig an die Charakteristika des Browsers gebunden wird.

Contents

Introduction

The Internet as it is used today has become broadly fragmented regarding how it is used and with a multitude of different software protocols. Browsing the web for information, watching videos and listening to music or communicating using tools like e-mail, instant messaging or social platforms like Twitter or Facebook are nowadays one of the core use-cases for millions of people around the world. The "dark side" of the Internet can be observed in everyday news. Spam, 0day exploits, underground marketplaces, identity theft and many other problems and attack vectors as well as exploitation techniques are used nowadays to conduct mischief on a daily basis. Users are furthermore lured into giving more and more of their private data to companies that use them to generate revenue, although many of these companies struggle to adequately protect their users' data from a technical point of view. Password breaches with millions of affected users have become mainstream: RockYou lost 32 million passwords in 2009, Adobe lost 150 million passwords in 2013 [74] and up to 40 million customers' credit card information has been stolen in late 2013 [10]. When such incidents occur, methods of digital forensics are used by law enforcement, forensic investigators and system administrators to reconstruct the timeline of events, identify possible traces left behind and identify the impact of breaches. Digital forensics, as defined by NIST, is the application of science to the law, in particular *"the identification, collection, examination, and analysis of data while preserving the integrity of the information and maintaining a strict chain of custody for the data"* [71]. The tools and methods used are definitely not new and have been employed for decades [117]. However, digital forensics has recently manifested as a subfield of computer science, and curricula are set up all over the world to allow students to study in this field.

Digital forensics have received increasing attention in recent years as more and more crimes are committed exclusively or with the involvement of computers. Digital traces help courts and law enforcement agencies to capture valuable evidence. Existing research as well as applications in the area of digital forensics focus on file systems, log files, network traffic, databases and, more recently, mobile devices like smartphones and tablets [61, 84]. The traces that are left on mobile devices in particular can contain a plethora of information, e.g. pinpoint past communications as well as exact locations due to the use of GPS and wireless networks, regardless of the exact communication channel used (GSM, instant messaging or e-mail, just to name a few). The traditional approach of digital forensics is to acquire data from a device in a forensic lab using special hard- and software. The current approach of analyzing local, residual artifacts has however two major shortcomings: for one, not only the seizure of a suspect's device is a required, but it has also to be accessible to the investigators. Secondly, and even more importantly,

the emergence of new online services extend the traditional means of data storage, information sharing and communication. These services often do not operate under the same jurisdiction as the user, which can make it difficult to obtain the data. Law enforcement can ask service operators to release certain information, but they are usually not obliged to answer requests from other countries. However, there have been documented cases in Austria were this method was successful [124]. Furthermore, the more prevalent use of encryption can make data acquisition and analysis very hard while counter-forensic tools can be readily obtained from the Internet. As such, digital forensics are often behind the current state of technology.

Background

The forensic process usually starts with the identification and verification of an incident, the acquisition of the devices affected and the analysis in a forensic lab [71]. If the device is found running, the process of data acquisition starts by copying the content of the volatile memory since it contains important dynamic information that is not stored anywhere else. This is specified as the *"order of volatility"*, which dictates that volatile information is a priority during the acquisition process [19]. The content of the RAM is often incredibly useful during investigations as it contains the list of running processes, open network connections, encryption keys etc. Methods that can be used are for example special software or loadable kernel modules, a cold boot attack [58] or modular extension cards that support *direct memory access* (DMA) like Firewire. The acquisition process has to be done carefully, as many approaches modify the content of RAM. If a software-based approach is used (running acquisition software on the machine), this is problematic as it modifies the RAM itself - the process needs to be executed and thus needs space in RAM that can potentially overwrite other information in RAM. The investigator furthermore has to trust the underlying operating system and hardware, as they could be modified to thwart the acquisition process. Hardware-based approaches that rely on DMA can be problematic too, as the Northbridge that executes DMA can be tricked into seeing different memory content as the CPU [113]. This leaves the analyst with the cold boot attack, but it requires an enormous effort compared to the other approaches and as such is expensive and error-prone. While the cold boot attack was originally derived to extract cryptographic keys from memory, it is also suitable to acquire an image of the system memory. Even though it was commonly believed that the content of RAM is deleted once power is shut down - since RAM cells need to be refreshed periodically by the system bus to store information - the cold boot attack showed that RAM content is retrievable even 10 minutes after shutdown (and possibly even more) if the RAM is cooled down [58]. It was shown that this method can even be used to parse the RAM on the fly and reboot the system into a forensically controlled environment without service interruption [25]. Recently, the cold boot attack was extended to be used on mobile devices running Android [95]. The content of RAM is furthermore not reproducible, as the repetition of the exact steps that were taken by a user to arrive in a particular RAM state can lead to a different state due to the operating system, background processes or external influence not under the control of the user, e.g. network connections. A recent paper discusses anti-forensic capabilities for memory acquisition and presents a new acquisition method based on direct page table manipulation and PCI hardware introspection [118].

The system is then abruptly powered down to prevent the execution of shutdown scripts or dead man switches that could possibly modify data. Afterwards the hard drives are copied multiple times using hardware write blockers to prevent unintended data changes and to preserve the integrity of the information [71]. This includes at least a working copy and a backup copy. Every step is fully documented to preserve a full chain of custody [71] and to allow the evidence to be usable in court. Hash values like MD5 [108] or SHA1 [42] are commonly used to guarantee the integrity of the data. Prior to imaging it, a hash value is calculated over the entire hard drive. This hash value is used after imaging to verify that the newly created image contains in fact the same data as the original, but also to proof that the imaging process has not altered any data. Once the working copy has been created, special tools like *EnCase* by Guidance Software[1], *FTK* by AccessData[2] or the open source tool *The Sleuth Kit*[3] are used to assist in the process of analyzing the data, to index the hard drives and process them to extract all possible information of relevance including file system metadata and the data stored in the files itself. However, the fundamental change in the computing environment from PCs and local servers to outsourced services, Web 2.0 applications and cloud computing requires fundamentally new technology regarding digital forensics.

The worst case for a traditional forensic analysis is, at the time of writing, a computer without a hard drive and turned off on arrival. Readily available boot media can be used to start a live Linux distribution, whereas all data is exclusively stored online. Without network traces or a memory dump, there is simply no locally stored information to analyze. Storage providers like Dropbox, Microsoft's SkyDrive or Google Drive offer enough storage capacity to store data online, which can be downloaded to a RAM drive that is in turn deleted upon reboot. This is similar to a turned-off computer with encrypted hard drives - without further knowledge of the key or its copy in RAM, it is infeasible to get the data in clear text. This is for example why the raid on Max Butler (aka Iceman) [103] was accompanied by a team of forensic experts from Carnegie Mellon University to acquire the RAM of the running computers, as it was known to law enforcement that Max Butler was using the software *DriveCrypt* to encrypt his hard drives. It is possible to use encryption without storing the key in RAM, e.g. for RSA [55] or full disk encryption using AES [93]. This has been shown to be feasible for PCs [94] as well as for mobile platforms using ARM [57], the key is stored in a CPU register and never written to RAM. Other artifacts like the list of running processes, open network connections etc. are however unencrypted in RAM.

Network forensics rely (mostly) on the analysis of network captures. Usually this means that the upper network layers are analyzed, in particular the application layer as well as the Transport and Internet layer (according to the TCP layering model [101, 102]). While the application layer contains the actual user communication, including important online protocols like HTTP for web browsing and POP/IMAP for e-mail, the network layer contains the technical information needed for the communication transmission, e.g. IP addresses and port numbers. Both

[1] `https://www.encase.com/encase-forensic.htm`
[2] `http://www.accessdata.com/products/digital-forensics/ftk`
[3] `http://www.sleuthkit.org/`

are vital in an investigation, given that either can contain relevant information. If encryption at the application layer is used, the network layer still reveals metadata and contains information on who is communicating with whom [30]. Network analyzer software like *wireshark*[4] can be used to extract information from the network capture on all layers. It can also be used to decrypt encrypted information if the key is known, for example in case of TLS and HTTPS. More advanced (open-source) tools like *Xplico*[5] can automatically generate statistics and reports of common interest for a multitude of use-cases. *PyFlag* is a tool which is able to reconstruct the target's view in the browser by parsing HTTP traffic [27]. This is especially useful as the entire web session can be vividly presented from the targets point of view to persons not familiar with the technical details of the Internet, e.g. for juries and lawyers, for expert witness presentations in court. Sometimes it is also possible to find data that contains network data structures [14] on hard drives, which can happen if RAM content is swapped to disc and has not yet been overwritten.

Problem Description

The problems of digital forensics are manifold: new devices, operating systems and file systems (e.g. btrfs [109]), in particular currently on mobile devices and smartphones, can make the acquisition of data cumbersome. Cloud computing [11] and the emergence of cloud services render the acquisition of hard drives useless, as vital information is often stored in anonymous data centers around the world - without direct access for law enforcement agencies - and not on local hard drives anymore. Another problem is that it's infeasible to acquire an entire data center, not only logistically but probably also legally. Data formats, operating systems and hardware are often proprietary and custom-built, and existing software solutions would have to be adapted for analysis. Comparably, forensic analysis can already be challenging for large systems like e-mail- or storage servers, as terabytes of data need to be processed and often expensive hardware like special RAID controllers are needed to obtain access to the information. Large service providers like Facebook, Google, Yahoo and others thus comply with law enforcement and extract the data for the requestor, while the agency has to trust the service provider that the methods used are forensically sound and all the relevant data is extracted. The use of encryption can furthermore complicate data acquisition, and legal boundaries are often problematic as the Internet by its very nature is international and knows no boundaries.

Another upcoming problem for digital forensics is the scale of data to analyze and the time required for conducting a forensic analysis: a commodity 4TB hard drive can be bought nowadays for less than US $200, but it takes many hours to simply create an image for analysis or calculate a hash value over the entire hard drive. The overall amount of information that needs to be processed is increasing exponentially, therefore automated tools with large throughput will become more and more important [52]. *bulk_extractor* [53] by Simson Garfinkel for example is a tool that can search unstructured data and hard drives for e-mail addresses, credit card numbers and more, using recursive scanners. *bulk_extractor* is designed to extensively build upon

[4] `https://www.wireshark.org/`
[5] `http://www.xplico.org/`

multithreading, thus yielding a very high overall performance. An anecdotal story tells that the tool was able to pin a server with 48 CPU cores, thus effectively parallelizing data analysis. Another approach to reduce the manual work for the analyst is the use of white listing hash values of benign files on a hard drive. NIST is quarterly releasing the National Software Reference Library reference data set (NSRL RDS)[6] which contains more than 110 million hash values for files, executables and software libraries of common operating systems and software products. Another approach that seems promising is the use of sector hashing: instead of hashing files, the sectors of the hard drives are hashed individually. This has been shown to be a promising approach [50, 128], as current use file systems like NTFS or FAT are sector-aligned [21]. To reduce the addressing overhead, NTFS, FAT as well as other file systems logically combine several sectors to clusters which would be another vector for hashing since files are split into clusters on the hard drive. NTFS uses 4 kilobyte clusters as default value for file systems smaller than 16 terabytes, which means that on older hard drives 8 512 byte sectors are combined into a 4 kilobyte cluster. ATA hard drive sectors used to be 512 bytes in size, but newer hard drives are transitioning to 4 kilobytes sectors as the new default value. Even though sector hashing seems promising in terms of accuracy and precision (many files do not share common sectors [128]), one of the limitations of this approach is the size of the hash values and the overhead to query large hash samples. A hard drive with one terabyte capacity has around 250 million 4 kilobyte sectors, resulting in 250 million hash values. Even though the use of GPUs [80] or MapReduce [32] could facilitate large-scale analyses and distributed processing of hash value comparisons, this has not been evaluated regarding performance and accuracy (precision/recall). False-positives and false-negatives could have a severe impact in such large data operations, as they could lead to expensive manual inspections and potential base rate fallacy. Furthermore, multiple different hash window sizes could be beneficial, as different sources of hash values can then be considered as sources: Dropbox uses up to 4 megabyte file chunks for hashing, and many P2P file sharing applications like *Gnutella* or *BitTorrent* use variable hashing windows [77] depending on file size and number of files.

One of the solutions to analyze the ever increasing file numbers and storage capacity are so-called "approximate hash functions", also known as *fuzzy hashing*. Compared to cryptographic hash functions like MD5 and SHA-1, fuzzy hashing has the benefit that the change of a single bit in the input does not entirely change the resulting hash value. Instead, they are designed to calculate a score value between 0 and 100 on how similar two different files are - 100 if two files are entirely similar, and 0 if no similarity can be found. The benefit is that related files can possibly be identified, as well as previous versions of the same file. It is possible to retrieve files in different versions, for example if the file has been moved across partitions, the file system has been defragmented (in the case of a fragmented file), or if it has been extended and the old clusters have not yet been overwritten. This is also true for copy-on-write file systems like btrfs [109], or in SSD hard drives, as they by design update files at different locations than the original file content. The most important fuzzy hashing tools so far are *ssdeep* [73] and *sdhash* [110]. While ssdeep generates a constant 80-byte output for each file, sdhash generates a similarity digest of variable output size. Both can then be used to calculate the similarity score

[6] `http://www.nsrl.nist.gov/`

between files by comparing the fuzzy hash values of files either by using the edit distance (ssdeep) or the Hamming distance (sdhash). Numerous comparisons between those two tools have been published [111, 18], and generally both have their advantages. NIST also releases fuzzy hashes for a subset of the NSRL RDS using sdhash[7] and ssdeep[8] as well as certain blocks of files like MD5 on the first 4 kilobytes of the corpus files[9].

Counter-forensic tools are another front that forensic examiners have to battle. Many different approaches can be used to hinder forensic analysis, e.g. encryption, online network traffic anonymization or file signature analysis. Tools to encrypt hard drives like *LUKS*, *FileVault*, *BitLocker* or *TrueCrypt*, are readily available for all major operating systems and can render an analysis something between hard and close to impossible [22]. Communication content can be encrypted using protocols like TLS (HTTPS, IMAPS, SMTPS) or OTR [17], and online communication can be anonymized using Tor [37], JonDonym or I2P [107]. Tor is a very active field for research [120, 69] with regards to online privacy and network censorship resistance, and so far one of the effective countermeasures against spying according to files revealed by Edward Snowden [13]. Steganographic methods can be furthermore used to hide the existence of information in plain sight [70, 62], for example in pictures or videos, and network traffic can be shaped to look like something completely different [83, 121, 123] if the attacker is using traffic analysis to infer information about encrypted information content [12, 96]. If not done correctly, however, these methods can be defeated: very recent approaches like StegoTorus [123], SkypeMorph [83] and CensorSpoofer [121] have been shown to be vulnerable to detection [63]. Furthermore, cryptography, if implemented incorrectly, can considerably harm these tools, which has been shown recently on CryptoCat[10]. While all these problems seem troublesome in regard to digital forensics, they are essentially troublesome (to say the least) in oppressive regimes where the lives of dissidents are in danger. As such, this thesis is not judging on the way digital forensic methods are employed all over the world: just like every coin has two sides, digital forensics can be easily employed for or against certain interests. Thus, the usage of counter-forensic tools is nothing that should be considered to be negative per-se.

The discussion about privacy as the "right to be left alone" [122] is nowadays more prevalent than ever. In particular the revelations by Edward Snowden about the surveillance conducted by the NSA as well as other secret services has lead to public discussions on online security and privacy, as well as mobile phone security and dragnet-surveillance on a global scale in general. This will also affect research in computer security in the near future, as well as digital forensics. The average user is unprotected against such powerful adversaries, and often without a chance to even detect attacks until it is too late to mitigate or reduce the possible impact. Not only is the NSA actively exploiting software weaknesses on target's computers (codename *Quantum*), they are also collecting unencrypted as well as encrypted communication content of users on a large scale (codenames *Upstream*, *Tempora* and *Prism*) and between data centers of large Internet

[7] `http://www.nsrl.nist.gov/morealgs/sdhash_3.3/sdhash.html`
[8] `http://www.nsrl.nist.gov/ssdeep.htm`
[9] `http://www.nsrl.nist.gov/morealgs.htm`
[10] `http://tobtu.com/decryptocat.php`

companies like Google and Yahoo (codename *Muscular*). The NSA is also accused of weakening the standardization process of a cryptographic pseudorandom number generator published by NIST (codename *Bullrun*) DUAL_EC_DRBG [116], which was the default PRNG in many products including RSA's [56]. The full extent of the NSA's surveillance programs is still unclear, and new revelations are constantly released.

The naive approach to conduct network forensics has several limitations, as simply encrypting the data in transit is insufficient. Anonymizing networks like Tor, as well as SSL MITM attack can be used to inspect data, and secret service agencies can be expected to do that on a regular basis. HTTPS is one of the most important protection mechanisms for data in transit. It uses the TLS (and SSL) protocol to secure online communication, with TLS 1.2 [34] being currently the most recent version of TLS. Client and server can authenticate themselves to each other, use the TLS protocol to derive an encryption key as well as agree on a symmetric encryption algorithm like RC4 or AES [115] by using public key cryptography [35]. However, it is still not commonly used for a large portion of websites, with just a few notably exceptions: Gmail uses it by default for all users since January 2010 [114], whereas Twitter followed in February 2012 [119]. Facebook announced in July 2013 that they enabled HTTPS by default for all users [106], while Yahoo announced to follow sometime early 2014 [100]. HTTPS also has been subject to a plethora of attacks in recent time [26]: RC4 has been found to be weak and insecure [6], CBC as the preferred mode of operation has its problems with attacks named BEAST [39] and Lucky13 [7]. Numerous attacks targeted compressed plaintexts prior to encryption, e.g. CRIME [40] and BREACH [104]. TLS errors are hard to understand for users [3, 4], the trust chain when using commercial certificate authorities can be troublesome [41, 8] and implementations in mobile apps like Android are often insecure [47, 46]. As such, HTTPS is not suitable to defend against powerful adversaries. Especially with targeted attacks as well as broad, Orwellian surveillance manifesting itself as a global, passive adversary, defense in depth is the only option. Encryption alone is often insufficient, and protection mechanisms are needed on multiple levels.

To counter these developments, this thesis aims at providing insights into online storage providers as well as extend the current state of the art in digital forensics for this kind of scenarios. The traditional approach to conduct a forensic analysis of online storage systems has several limitations: for one, the operator is involved in retrieving the information, which is problematic as it needs to put trust in the operator who is also often not obliged to help due to cross-country jurisdictions. Furthermore, it can take weeks or even months to obtain information. The ultimate goal for this thesis is to develop novel information extraction techniques as well as to critically assess existing processes and methods, in particular for distributed environments. Counter-forensic methods are evolving, and the increasing prevalence of mobile computers and online connectivity will further challenge forensic investigations. The ever more pervasive usage of encryption will furthermore advance the difficulties when analyzing computer systems.

Proposed Solutions

This chapter describes the goals and methods used to obtain the results of my evaluations.

Goals

The goal of this thesis is to tackle current problems of digital forensics in connection to online services, like anonymizer- and online storage systems, and work towards their solution in a broader picture. The goals for this dissertation are to enhance the current research on digital forensics, with special focus on the following areas:

- Information extraction from online- and **cloud services** as well as the corresponding area of **browser forensics** in a sound way, so that the information is usable for examinations and in court.
- Analysis of **anonymization networks**, in particular Tor: how they are used as counter-forensic tools, and what information still leaks from their usage.
- Examining **online storage** services like Dropbox and how traditional file forensic methods can be applied.
- Assess the feasibility of **fully automated digital alibis** with regards to counter-forensics.
- Enhance **browser fingerprinting methods** to reliably detect a given browser based on its characteristics.

So far, cloud forensics is conducted mostly in a passive way - a request is sent to the service provider, and trust has to be put in the answer to meet forensics requirements, in particular regarding data completeness, presentation and preservation. In a recent paper, an API-based approach has been proposed to conduct forensic analysis on social networks [64]. This has the benefit that no direct involvement of the service operator is required, and data acquisition can be considered repeatable. The API often allows access to additional information compared to webpages, like exact timestamps and additional fields of information. Regarding online storage services, we would like to assess whether a given file is stored at the service or not, and, additionally (if possible), by which user. This allows us to test a set of possibly critical files, without the need for dragnet surveillance or passive deep-packet inspection that has to be already set up

prior to an incident. This approach is successfully used for example by PhotoDNA[11] (on Twitter, Facebook and bing) which uses a form of proprietary image hash to test image similarity to a known set of images in connection with sexual exploitation of children.

With regards to browser artifacts, the reliable identification of a browser is still non-trivial. Even though the browser identifies itself via a string (the UserAgent string), this is not a security feature - it can be easily spoofed or modified by the user. As such, webservers that log the UserAgent of users cannot be certain that the browser used was indeed the one proclaimed. We would like to identify new methods that allow us and others to draw conclusions on the specific browser of a user by looking at its characteristics. These methods can be either active or passive, e.g. for the web server or for captured network traffic. In a forensic context this would be of importance for example in cases where a web server gets hacked, as the access- and error logs usually contain not only the IP address of the attacker but also the UserAgent. On the other hand we would like to use our findings to improve the overall security of session management, as broken session- and authentication management is currently among the most prevalent threats online ("A2 – Broken Authentication and Session Management" in the OWASP Top 10 from 2013) [99].

Regarding online anonymization services like Tor, many parameters of its usage are in the dark. It is unknown why people use it and why they donate bandwidth to run Tor relays, even though many hundreds of thousands of people use it every day and there are currently more than 5,000 Tor relays. Related work found that it is mostly used for browsing the web and downloading files using P2P protocol [81], but this work dates back to 2008 when the Tor network was much smaller. Furthermore, P2P network usage has been found to be possibly dangerous on Tor as many torrent clients leak identifiable information like the client IP address [16]. We would like to understand how Tor is used nowadays, and especially if it is used in a secure way as recommended by the Tor project itself. Furthermore, as Tor could be used as a counter-forensic tool, we would like to understand the implications of its usage for digital forensics, in particular network forensics, and if the data transmitted still leaks information that could be used to reduce online anonymity.

Finally, we would like to draw attention to the fact that traces left on hard drives are not necessarily coming from a user or background processes, but can merely come from a purportedly fully automated program to thwart analysis. To show this, we will implement a digital alibi engine that incorporates social interactions like chatting, writing e-mails and using local programs just like a normal user. Traditional forensic processes that rely on file system metadata and network traffic analysis are expected to be easily tricked by such a program into detecting patterns of user interaction in cases where no user was present. Another goal of our work is to study slack space, a well known artifact of digital forensics. While current analysis methods target slack space implicitly instead of explicitly (keyword search on entire hard drives by ignoring file system metadata), free tools are available to store data in slack space. These tools also allow to encrypt data prior to storing it in slack space, thus defeating the implicit analysis with keywords.

[11] `http://www.microsoft.com/en-us/news/presskits/photodna/`

There is no existing prior work towards assessing the amount of slack space created by modern operating systems, with tens of thousands of system files. Given that many of these files are static and do not change over time, the slack space can be considered persistent and protected from getting accidently overwritten.

If possible, our methods and obtained results should be quantified and have to be compared to previous research in this field regarding effectiveness and applicability. If flaws and vulnerabilities in implementations or protocols are discovered, they have to be communicated to the creators and developers as soon as possible to improve overall security of the affected products. Research results in general (papers, data and tools) will be published as openly as possible. Ethical guidelines and codes of conduct from, e.g. ACM [9] and IEEE [68] are honored as well as ethical principles that are agreed upon by the majority of the scientific community. Many ideas discussed in the literature as well as published in the papers presented here have limitations and restrictive conditions that hinder the general applicability. As such, the goal has always been to find the most permissive set of preconditions and requirements while making sure that the prospect and implications are still generally applicable and not restricted to particular technologies. An example from one of the papers presented here would be the data deduplication attack, which was demonstrated on the cloud service Dropbox: even though it was specific to the implementation and particular protocol used by Dropbox, the attack itself is based on the naive use of client-side hashing for data deduplication [60], compared to more complicated data possession proofs [130, 59]. We showed that trusting the client to do this faithfully and correctly is not to be taken for granted and a possible attack vector for unauthorized data access [91].

Methodology

The following methodology was used:

- **Extensive literature review** on the current state of forensic research and the identification of urgent challenges that need to be solved.
- **Prototype implementation** for proof of concepts like new data extraction techniques or analyzation methods.
- **Empirical analysis** of feature predominance and assessment of expected information increase in forensic examinations.
- **Prototype dissemination** with an **open source license like the GPL** so that the forensic community can use the outcomes and enhance their functionality as needed.

The majority of the results are based on quantitative research; a-priori research questions [29] were typically formulated in this way. This is in particular true for novel attack vectors, as a proof-of-concept is often needed to not only show the general feasibility of the attack, but also to assess the possible impact on a larger scale. Most data collections and evaluations are based

on or built around proof-of-concept implementations. The designs of the experiments are described in the corresponding papers in a way that they are reproducible in general; available data and used tools were openly published to make the results reproducible as well. However, responsible disclosure was employed: tools that exploit security weaknesses or could potentially harm users or service operators in any way will not be released until the underlying issues are communicated to the vendor.

We started our work on the online storage analysis by evaluating the protocol of the most popular software at the time: Dropbox. Our initial observation was that once a file has been stored on the Dropbox servers there is in general no need to re-upload it. Even if it was deleted and re-added to the Dropbox folder, somehow the client software tracked which files were already stored online. This was also true for cross-account uploads: another client (without any relation to the first user account but) with the exactly same file was not asked to upload the files to the servers. It was deduplicated, and the same exact file had to be uploaded only once. Upon looking at the obfuscated executable we were able to see that files are split into chunks of 4 megabytes of which each was hashed using the SHA-256 cryptographic hash function. The hash was then sent to the server, and if the chunk was already stored on the servers, retransmission was omitted. As the hash calculation was done locally, we were able to manipulate it and to obtain unauthorized access to data. Knowing the hash value of a file of interest, it could be obtained from Dropbox without prior possession of the file. This was of interest for digital forensics regarding two particular methods: for one, it was possible to assess if a given file was stored on Dropbox, even though we did not find a way to identify the account that uploaded it; secondly, data could be hidden online without leaving any local traces, as long as the hash values are remembered. At a later point in time, this weakness could be used to download it again. Thus up to 4 megabytes were retrievable by one 256 bit hash sum. To evaluate the file retrieval attack, we downloaded the most popular files (without any obvious copyright on them) from The Pirate Bay torrent tracker[12] and downloaded the corresponding files from Dropbox. We also evaluated to what extend files could be hidden using another flaw in the protocol: once a file is uploaded, it was linked to the user account by a final request. By omitting this final linking request we were able to upload the files successfully to Dropbox, but without linking them to our account. Even weeks and months later the data was still retrievable. All in all, these issues were not only worrying from a security point of view, but also for digital forensics as a chance and a valid method for counter-forensics. We proposed countermeasures, in particular a data possession protocol to prevent our attacks while at the same time allowing deduplication. Dropbox fixed it, however, by preventing deduplication altogether, and every file since is uploaded to their servers. We speculate that this was an easy fix for them, given that Amazon, who is running the underlying infrastructure, is not charging customers for inbound traffic.

Based on browser fingerprinting artifacts we found that it is possible to identify a given browser by actively probing it with JavaScript. While browser fingerprinting is currently a very active area of research [43, 127, 86], we derived novel fingerprinting vectors that are three orders of magnitude faster than related work [85]. In our work we use fingerprinting of the underly-

[12] `www.thepiratebay.se`

ing JavaScript engine as well as upcoming browser features and standards based on HTML5 and CSS3, as these are not yet uniformly implemented in all major browsers and thus very suitable for identifying the browser currently used by the client. We started by analyzing the JavaScript engines for different browsers (Firefox, Internet Explorer, Opera, Chrome and Safari) on multiple operating systems (Windows 7, Windows XP and Mac OS X) and how they differ in standard accordance to the ECMAScript standard [44] for JavaScript. We then collected, for each browser/operating system combination (more than 150), the outcome of the official TC39 test suite, *test262*[13] and stored the results in a database. From that data we derived two different methods to minimize the computational overhead and runtime on the client. Furthermore we were able to show that UserAgent string modifications are easily detectable by our method. Based on these results we extended the scope of our browser fingerprinting to include upcoming web standards, HTML5 and CSS3. While some features are already uniformly agreed upon and implemented by the browser vendors, some are not even yet standardized, but already implemented in some browsers. We used that additional information to increase the precision of our active tests and built a framework to increase HTTP session security, by tying a HTTP session to the specific browser. Session hijacking becomes thus harder, as the attacker has to run the same exact browser, or needs additional information despite traditional session management information like session cookies. The framework, which was named SHPF, can thus detect session hijacking on the server side and implement proper countermeasures like requesting the user to re-authenticate or simply terminating the session for that user. We used this framework to also include additional encryption, as proposed in [2]. The framework is released under an open source license and can be easily incorporated in existing websites and frameworks with just a few lines of code.

Regarding anonymous communication and their (counter-)forensic effects we evaluated the Tor network in depth. We ran a Tor relay (and in fact are currently still running a non-exit Tor relay) and analyzed publicly available information like the Tor consensus information[14]. The consensus information contains all necessary information for the clients to build their paths through the Tor network and is essential to prevent intersection attacks where clients only know a partial state of the entire network [37]. While the Tor network's infrastructure is run by volunteers, we found that it is very stable in total numbers and bandwidth. We incorporated our data collection process into the tor-status website which used a Perl script to connect to a local Tor relay to periodically extract all information about the Tor network. tor-status is now superseded by Tor Metrics[15] and Atlas[16]. We furthermore designed analysis methods that allow to present the most relevant information on a first sight, like total number of relay, total number of exit relays or country-specific numbers. To analyze possible artifacts of user data we ran an on-the-fly analysis script on a Tor exit relay. We did this to find out if Tor is used in a safe way and as recommended by the Tor Project, and also to see if users can be deanonymized by merely looking at the unencrypted network traffic. Our data collection was built to be as least invasive to

[13] `http://test262.ecmascript.org`
[14] `https://metrics.torproject.org/data.html`
[15] `https://metrics.torproject.org/network.html`
[16] `https://atlas.torproject.org/`

user privacy as possible, and we analyzed only the HTTP requests issued over the Tor network without permanently storing them. Since Tor is used in many countries to circumvent online censorship, we found many connections to, e.g. Facebook or other social media websites that directly contain user information and were unencrypted by default at the time of the analysis. We also found unencrypted file downloads for software with known vulnerabilities (PDF, office files or executables) which could be piggybacked or exchanged with malware on the fly by an adversary running the Tor exit node.

To analyze the quantity and persistency of slack space, we installed 18 different versions of the Windows operating system ranging from Windows XP and Server 2003 R2 to the most recent Windows 8 RC and Server 2012 RC. Our goal was to analyze how many files are changed during system updates and how this affects the corresponding slack space. In total, more than 2500 system updates were installed, including 10 service packs. We used virtual machines to run that many versions of Windows and exported the file system to a raw image prior to analysis. Slack space is an artifact of clustering file systems like NTFS or FAT [21] and can be used to hide data in the last sectors of files, as these sectors are allocated to a file, but often unused since files hardly align exactly in size with the allocated space. Many tools are available to hide information in slack space, e.g. *bmap* or *slacker.exe*. We then collected the file system metadata using fiwalk [51] and estimated total capacity and stability. For all Windows versions we analyzed, 44 megabytes were available on average across all steps in our evaluation. From the initial file slack 78% were still available at the end of the evaluation process. Creating digital alibis with social interaction were another research problem we wanted to encounter: currently, in particular in court or in companies, digital forensic methods are used to find out what happened in what order of events. However, the traditional analysis processes can be thwartened with automated processes that manipulate timestamps on disks as well as generate network traffic. We built a proof-of-concept framework in Python that runs fully automated and is able to interact online with multiple means of communication (e-mail, chat). Our framework is capable of interacting with the computer like an actual user by sending key strokes and mouse actions as simulated input. As such, it can also be used to generate test data for new and upcoming software, not only hard drive images, but also network traffic of various kinds.

Scientific contributions

The scientific contributions of this thesis can be roughly categorized according to for parameters - they are either active or passive, and work online or offline. The passive contributions work with data or knowledge that is already there but has not yet been used in this particular fashion. Instead of modifying, e.g. source code or websites, the tools can work with log data or information that are already collected and processed. Active on the other side means that either on the client- or the server side additional code has to be executed and information collected for them to work properly, and existing frameworks and source codes need to be extended. Online solutions refer to contributions that require Internet connection, or work only in online environments, whereas offline means that they work without network connection of any kind. A graphical representation of the contributions according to this scheme can be seen in Figure1.

The passive online contributions contain the research on Tor - both the paper on infrastructure monitoring [88] as well as the paper on Tor usage and information leakage [67]. While the Tor infrastructure monitoring works with existing data, the network consensus, it can be used offline. However, since the Tor network consensus is updated once every hour, it can be beneficial to analyze the data as soon as it is published, and for that some form of network connection is required. For the historical data, which is available online from the beginning of 2006[17], the data can be analyzed offline. That is why Tor infrastructure monitoring is located in the graph close to the line between online and offline. The Tor information leakage analysis works only in online environments, but is completely passive in nature. As such it is located in the upper left corner. The lower left quarter contains the work on Windows file slack analysis and persistency [89], as it neither requires network connectivity nor an active component besides the collection tool. Measuring slack space can be even done on the fly in the running operating system (for most of the files), which means that it can be calculated passively and offline. The rest of the papers are located in the upper right quarter, as they are active in nature and require network connectivity. Automated alibis [15] can make use of the fact that a computer is online by generating social interaction with online communication protocols like e-mail, browsing the web or instant messaging. Without Internet connection, only local information is modified and programs that work on local data are run. The online framework regarding JavaScript engine fingerprinting [90], online storage deduplication and slack space [91], as well as the SHPF framework [92] work purely online and need modifications of existing code and frameworks. Further distinctions for these papers would be whether the modifications have to be done on the server- or the client side

[17] `https://metrics.torproject.org/data.html`

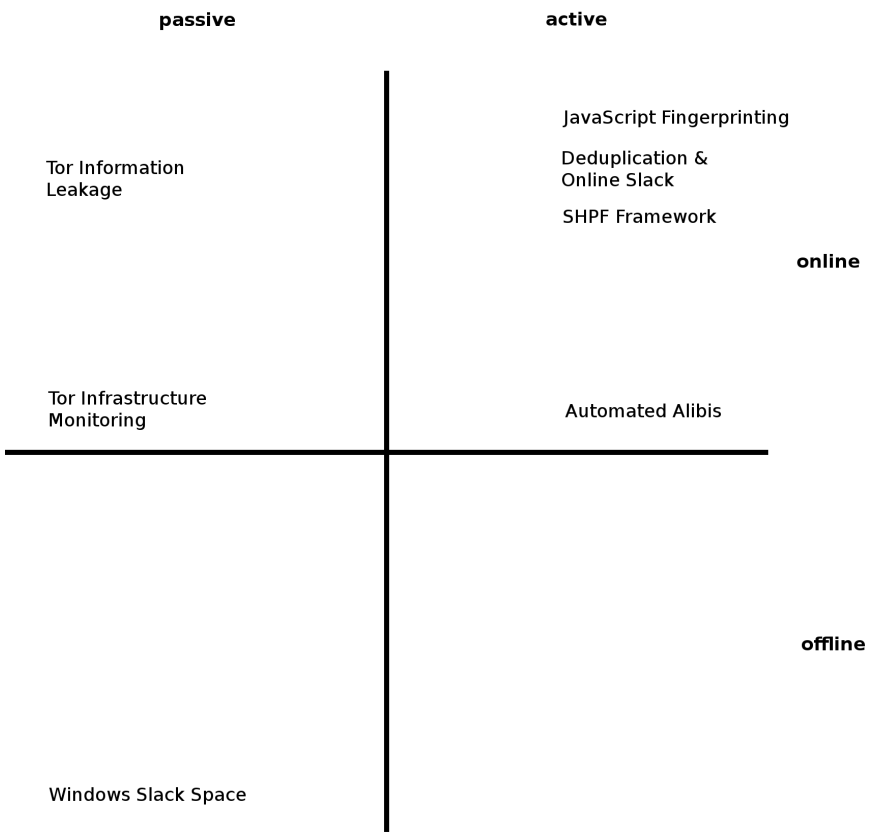

Figure 1: Classification of Contributions

and if the additional code needs to be executed at the client, at the server or at both, but was omitted for reasons of brevity. SHPF can be added to existing webpages with a few lines of code to invoke the SHPF framework. JavaScript engine fingerprinting needs to add the required tests to the webpage's source code, and online data deduplication with the possibility of online slack space requires local modifications to the source code, in our case Dropbox.

The papers and in particular their context are discussed in detail in the following, categorized accordingly to the previously described research fields: online storage, anonymous communication and digital forensics.

Digital Forensics on Online Storage

The paper **Dark Clouds on the Horizon: Using Cloud Storage as Attack Vector and Online Slack Space**, which was published at the *USENIX Security Symposium* [91] in 2011, describes

three different attacks against Dropbox[18], a popular cloud storage service. The first attack abuses the local hash computation on the client side to get unauthorized access to remotely stored files - if the hash value(s) of the file are known to the attacker. While this scenario can pose as a hen-and-egg problem, it could be used for file sharing and covert communication as (independently) implemented by Dan DeFelippi [33, 82]. This attack could further be used for stealth data exfiltration, as only a hash value has to be secretly transmitted to the attacker instead of the entire file. While the underlying problem for this attack has been independently found by Harnik et al. [60], our work had a proof-of-concept implementation and showed that the attack is applicable on the (at the time) largest cloud storage service provider with more than 10 million users. Dropbox changed their protocol soon after our notification. As of July 2013, Dropbox has 175 million users [28] and is still subject of active research [72, 38, 105]. The second attack is based on the observation that Dropbox did not validate or tie the so-called hostID to the specifics of a system once it is set - if this somehow becomes known to the adversaries, they can access all files of the victim. This attack has also been independently discovered by Derek Newton [97]. Both attacks happen completely transparent to the victim, who cannot detect these attacks as they are targeting the protocol and the servers only. The third attack abuses the transmission protocol by up-/downloading files without linking them to a specific account, which usually happens after uploading files. This can be used to secretly hide files at Dropbox and thus inside the Amazon cloud [38]. This attack can furthermore be used to upload files to other peoples' Dropbox if the victim's hostID is known. None of these attacks are specific to Dropbox, but apply to other cloud storage services with vulnerable implementations or protocols as well. Dropbox fixed these issues after notification by disabling client-side data deduplication entirely and encrypting the hostID at rest. Our paper finally proposed to use interactive challenge-based data possession proofs instead of relying on cryptographic hash functions as sole method to check if a client is really in possession of a new file which is possibly already stored on the server. Numerous other methods have been recently proposed in the literature [59, 130, 129], but our countermeasure specifically targets the use-case in which two parties (willingly or unwillingly) collaborate to facilitate file sharing and cryptography alone is not suitable as a sole countermeasure, given that cryptographic keys can be exchanged between collaborating parties.

The paper **Fast and Reliable Browser Identification with JavaScript Engine Fingerprinting**, which was published at the *Web 2.0 Workshop on Security and Privacy (W2SP)* [90] accompanying the IEEE Symposium on Security & Privacy in 2013, improves previous research in the area of browser fingerprinting based on the Javascript engine by three orders of magnitude. While previous research used performance and timing patterns [85] of various benchmarks, our method uses Javascript conformance tests which are available online, in particular *test262*[19]. These tests assess to what extend the Javascript engine of the browser conforms to the official ECMAScript standard, and the failed test cases are specific for browsers and particular browser versions. Recent browser versions failed at something between four and 35 cases, whereas older browsers were having problems with more than 3,500 cases out of the approximately 11,500 tests. While a full run on all these 11,500 tests only takes about 10 minutes on a modern PC

[18] `https://dropbox.com`

[19] `http://test262.ecmascript.org`

and up to 45 minutes on a smartphone, we also discussed techniques to make our tests as efficient as possible. Our tests to fingerprint a particular browser required only a few hundred lines of code to be executed on the client side, which reduces the runtime to a fraction of a second. We evaluated our approach using a plethora of browser versions and operating system combinations, resulting in more than 150 configurations. We derived techniques to build a decision tree to find the browser of a user without any a-priori knowledge like the UserAgent string. We were also able to show that the underlying Firefox version used in the Tor browser bundle [37] can be identified as it employs a modified UserAgent string to increase the size of the anonymity set [36]. From the browser bundles that were released between May 2011 and February 2013, our technique was able to identify 6 out of 9 browser versions correctly and find the modified UserAgent strings. This is however not an attack on Tor or the Tor browser, but can be used to decrease the size of anonymity sets. Still, it could be used to conduct forensic analysis on clients connecting to a webserver using Tor, but also to identify modified UserAgent strings during web sessions. Unfortunately the dataset we collected was lost due to a hardware failure.

The paper **SHPF: Enhancing HTTP(S) Session Security with Browser Fingerprinting**, which was published at the *International Conference on Availability, Reliability and Security (ARES)* [92] in 2013, builds upon previous work on browser fingerprinting and presents a framework that allows HTTP session security to be enhanced using browser fingerprinting. Traditional HTTP session security, as it is used today, has many shortcomings. This was also demonstrated with the release of numerous tools that allow automatic session hijacking, for example in unencrypted wireless networks, e.g. FaceNiff[20], Firesheep[21] or Droidsheep[22]. SHPF ties the session of a user to a characteristic set of properties of the underlying browser using browser fingerprinting and is implemented in a modular way. As such it can be extended using upcoming fingerprinting methods like HTML5 rendering differences across browsers [86] quite well. Part of the implementation were HTML5- and CSS feature fingerprinting, as well as binding a session to a particular UserAgent string or IP address. Browser fingerprinting has been recently discussed to be already in use by advertising networks [98, 1], and as such we believe that the public release of the SHPF source code can help to secure online communications. This is an extension of the previously discussed idea to identify browsers in forensic investigations and to prevent session hijacking. While session hijacking can have legitimate use-cases, e.g. to transfer a web session to an untrusted terminal without entering a password [20], it is usually an indicator for hacking attempts.

Insights into Anonymous Communication Methods

The paper **Anonymity and Monitoring: How to Monitor the Infrastructure of an Anonymity System**, which was published in the journal *IEEE Transactions Systems, Man, and Cybernetics, Part C: Applications and Reviews* [88] in 2010 describes data collection of public information

[20] `http://faceniff.ponury.net/`
[21] `http://codebutler.github.io/firesheep/`
[22] `http://droidsheep.de/`

and numerous evaluations on the Tor network [37]. Tor is an anonymizing overlay network, and the underlying infrastructure (the Tor relays) is run by volunteers. The paper clearly showed that not only the number of servers is volatile, but also that there are significant country-specific usage patterns in the type and total number of Tor relays. For example, the number of Tor relays can be considered rather static in the US during an one week period, while the total number of relays in Germany had a daily pattern: +/- 10% (out of approximately 350) of the servers were running only during the evening hours. The same was surprisingly true for exit relays. Even though we could not find any indication that these patterns are correlated, they clearly showed that Tor relaqys and how they are operated should be further investigated. The overall number of Tor relays has tripled since the paper was published, while the number of exit relays has doubled. The paper has been an extension of my master thesis conducted at the Vienna University of Technology in 2009 [87]. The Tor project incorporated parts of the results and evaluations in their official Tor metrics portal (online at `https://metrics.torproject.org/`) and independently published results and related methods [78, 79] similar to ours.

The paper **Tor HTTP Usage and Information Leakage**, which was published at the *IFIP International Conference on Communications and Multimedia Security* [67] in 2010, showed that Tor users are unwillingly leaking (possibly sensitive) information by browsing the web. It furthermore showed that Tor users are often vulnerable to MITM-like file replacement attacks, as a malicious Tor exit relay can easily replace certain requested files. Furthermore, plenty of social networking information was observable in the clear, which makes user deanonymization often trivial. Related work from 2008 analyzed [81] how Tor clients use the network in general, and a paper from 2011 showed that using Tor for P2P file sharing often leaks identifiable information that allows user deanonymization [16]. Today, Tor users are often protected by the Firefox extension *HTTPS everywhere*, which uses a whitelist of about 10.000 preconfigured websites[23] to encrypt communication content with TLS by default. Websites that use TLS for all connections, like Gmail (since 2010), Twitter (since 2012) and Facebook (since 2013), are still a minority, even though anecdotal evidence suggests that the computational overhead is small [75]. This leaves room for novel attacks, e.g. our friend-in-the-middle attack on social networks [65, 66]. With the release of the NSA files by Edward Snowden, numerous website operators and e-mail service providers have started to switch to https-by-default, including Yahoo, Microsoft and LinkedIn.

Contributions to Traditional Forensic Techniques

The paper **Quantifying Windows File Slack in Size and Stability**, which was published at the *IFIP WG 11.9 International Conference on Digital Forensics* [89] in 2013, analyzed the quantity and persistence of file slack space in Windows environments, with special regard to Windows system updates. The intuition behind this paper was that once Windows is installed, it is likely that many system files are never touched again, updated or rewritten with regular usage, e.g. font

[23] `https://gitweb.torproject.org/https-everywhere.git/tree/HEAD:/src/chrome/content/rules`

files, program libraries, help files or pictures, as well as many other file types. If a file is kept for a long time without modifications, the slack space behind the file is static. Therefore we analyzed 18 different versions of Windows, ranging from Windows XP to Windows 8 RC, respectively Server 2003 R2 to Server 2012 RC, and measured slack space capacity and stability. While the detailed results can be seen in the paper, we observed that tens of megabytes, sometimes even more than a hundred megabytes of file slack space are available just by inspecting the files of the operating system itself. This is due to the fact that Windows is a complex system, using tens of thousands of files, but the results are applicable to any other operating system that users sector clustering for file system efficiency. Slack space is a well known phenomenon in digital forensics, where fragments of files can be possibly recovered once the cluster was marked deleted, reused and the old data was not fully overwritten [54]. Identifying file fragments regarding their file type or what file they originally belong to is a closely related and very active area of research at the time of writing [112, 128].

The paper **Towards Fully Automated Digital Alibis With Social Interaction**, which was published at the *IFIP WG 11.9 International Conference on Digital Forensics* [15] in 2014, raises awareness regarding the problem that forensic investigations can be foiled with fully automated, digital alibis. Digital evidence is often regarded as per-se authentic and tamperproof, at least to my general impression, in particular in court cases in which the analysis is often done by expert witnesses. We proposed and released a framework as a proof-of-concept to show that digital alibis cannot be blindly trusted. Our framework is, among other things, built upon social interactions, whereas chatting in Skype or interacting on Facebook is simulated. A similar technique to our approach has been previously discussed as an attack vector for fully automated social engineering [76]. Captured network traffic, as well as hard drive analysis [19], cannot easily tell the presence of our framework if certain (non-trivial) precautions are met. We suspect however that advanced analysis techniques as well as statistical analysis are able to detect the usage of the framework. Compared to previous work in this area, our framework is not dependent on particular operating systems like Android [5], OS X [23] or Windows [31, 24], as it is written in Python. However, it was recently announced that the Skype desktop API will be retired by the end of 2013 [45]. Since our implementation uses this API, we will have to change that usage towards GUI automation. Another potential use-case for this framework is the fully automated generation of disk images and network captures for educational purposes. While there exist numerous standardized forensic corpora [49] that are used to develop new methods and evaluations, all of them have some form of limitation [125]. For one, the real world corpus holds currently in total approximately 70 TB of disk images [48], but access to them is tricky - for one, sending and processing that vast amount of information is non-trivial. Secondly, as the owner's institution is located in the US and the hard drives could contain potentially sensitive and personal information, a US-based institutional review board (IRB) approval is required. However, many other countries in the world (and Austria in particular) do not have these form of approval process, which can as such be considered an obstacle, or at least time-consuming. Our framework allows the realistic generation of organic disk images, since the operating system and the applications on top of it are really executed, and as such contain every piece of data and metadata that an actual user would leave behind. This is different to related work, which creates synthetic hard

drive images by simulating activities of users using Markov chains [126]. While the synthetic approach is faster and has more parameters that can be configured, the organic hard drive image has the benefit that instead of the creation process itself the user is simulated. It could also be used for the creation of realistic network captures, since an actual browser is used to surf the web, and real applications are used to communicate. It is extensible and can be easily adapted to include custom applications, additional features or scripting actions. We released our framework as open source[24].

[24] https://github.com/mmulazzani/alibiFramework

Conclusion

This thesis demonstrates new techniques for forensic investigations and highlights current challenges, in particular for online environments. With data being ubiquitously stored in the cloud and web interfaces being used for data access across numerous different devices, the forensic process as it is conducted today can be easily overwhelmed. Depending on the threat model and the capabilities of the person to be investigated, counter-forensic methods, its tools and the increasingly present use of encryption can hinder and even prevent forensic analysis. Nevertheless, this field of research is currently very active, and new methods and data extraction techniques are direly needed.

The results in particular demonstrate findings with forensic context on the Tor network and how it is used, how browsers can be fingerprinted and how this fingerprinting can be used to enhance HTTP session security as it is implemented today. This thesis also adds to traditional forensics by analyzing the size and stability of file slack space and presenting a framework that was used to contribute towards automated alibi generation by adding social interactions using numerous communication channels.

Overview of Research Contribution

Publications

List of published papers, in chronological order:

- Anonymity and Monitoring: How to Monitor the Infrastructure of an Anonymity System, published in the *IEEE Journal on Transactions Systems, Man, and Cybernetics, Part C: Applications and Reviews* in 2010 [88]
- Tor HTTP Usage and Information Leakage, published at the *IFIP International Conference on Communications and Multimedia Security* in 2010 [67]
- Dark Clouds on the Horizon: Using Cloud Storage as Attack Vector and Online Slack Space, published at the *USENIX Security Symposium* in 2011 [91]
- Quantifying Windows File Slack in Size and Stability, published at the *IFIP WG 11.9 International Conference on Digital Forensics* in 2013 [89]
- Fast and Reliable Browser Identification with JavaScript Engine Fingerprinting, published at the *Web 2.0 Workshop on Security and Privacy (W2SP)* in 2013 [90]
- SHPF: Enhancing HTTP(S) Session Security with Browser Fingerprinting, published at the *International Conference on Availability, Reliability and Security (ARES)* in 2013 [92]
- Towards Fully Automated Digital Alibis With Social Interaction, to be published at the *IFIP WG 11.9 International Conference on Digital Forensics* in 2014 [15]

Released Source Code and Tools

- Source code of our framework for enhancing HTTP(S) session security with browser fingerprinting at `https://github.com/mmulazzani/SHPF`
- Source code of CSS and HTML5 fingerprinting, as part of the SHPF source code
- Source code of our digital alibi framework using social interaction at `https://github.com/mmulazzani/alibiFramework`

Released Data Sets

- Data set of our NTFS slack space evaluation at `http://sba-research.org/wp-content/uploads/publications/slackspaceDataset.7z`
- Data set for JavaScript engine fingerprinting has unfortunately been lost due to a hardware failure

(Co-)Instructed Master Theses

- Stefanie Beyer, master thesis at the Technical University of Vienna: "Towards automated digital alibis"
- Ioannis Kapsalis, master thesis at Aalto University: "Security of QR Codes"
- Robert Koch, master thesis at the Technical University of Vienna, funded by Google Summer of Code: "On WebSockets in Penetration Testing"
- Christian Kadluba, master thesis at the University of Applied Sciences Technikum Wien: "Windows Installer Security"
- Reinhard Kugler, master thesis at the University of Applied Sciences Campus Wien: "Analysis of Android Apps"
- Philipp Reschl, master thesis at the University of Applied Sciences Campus Wien: "Identifizierung der Webbrowser Version anhand des Verhaltens der JavaScript Engine"
- Thomas Unger, master thesis at the University of Applied Sciences Campus Wien: "HTTP Session Hijacking Prevention"
- Herbert Brunner, master thesis at the Technical University of Vienna (in progress): "Detecting Privacy Leaks in the Private Browsing Mode of Modern Web Browsers through Process Monitoring"
- Andreas Juch, master thesis at the Technical University of Vienna (in progress): "Btrfs Filesystem Forensics"
- Richard Köwer, master thesis at the University of Applied Sciences Wien Campus (in progress): "HoneyConnector - Detecting Eavesdropping Nodes in the Tor Network"
- Robert Annessi, master thesis at the Technical University of Vienna (in progress), funded by Google Summer of Code: "Improvements on path selection in the Tor network"

Bibliography

[1] Gunes Acar, Marc Juarez, Nick Nikiforakis, Claudia Diaz, Seda Gürses, Frank Piessens, and Bart Preneel. Fpdetective: dusting the web for fingerprinters. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 1129–1140. ACM, 2013.

[2] Ben Adida. Sessionlock: securing web sessions against eavesdropping. In *Proceedings of the 17th international conference on World Wide Web*, pages 517–524. ACM, 2008.

[3] Devdatta Akhawe, Bernhard Amann, Matthias Vallentin, and Robin Sommer. Here's my cert, so trust me, maybe?: understanding tls errors on the web. In *Proceedings of the 22nd international conference on World Wide Web (WWW)*, pages 59–70. International World Wide Web Conferences Steering Committee, 2013.

[4] Devdatta Akhawe and Adrienne Porter Felt. Alice in warningland: A large-scale field study of browser security warning effectiveness. In *Proceedings of the 22th USENIX Security Symposium*, 2013.

[5] Pietro Albano, Aniello Castiglione, Giuseppe Cattaneo, Giancarlo De Maio, and Alfredo De Santis. On the construction of a false digital alibi on the android os. In *Intelligent Networking and Collaborative Systems (INCoS), 2011 Third International Conference on*, pages 685–690. IEEE, 2011.

[6] Nadhem AlFardan, Daniel Bernstein, Kenneth Paterson, Bertram Poettering, and Jacob Schuldt. On the security of rc4 in tls. In *USENIX Security Symposium*, 2013.

[7] Nadhem J AlFardan and Kenneth G Paterson. Lucky thirteen: Breaking the tls and dtls record protocols. In *IEEE Symposium on Security and Privacy*, 2013.

[8] Bernhard Amann, Robin Sommer, Matthias Vallentin, and Seth Hall. No Attack Necessary: The Surprising Dynamics of SSL Trust Relationships. 2013.

[9] Ronald E Anderson. Acm code of ethics and professional conduct. *Communications of the ACM*, 35(5):94–99, 1992.

[10] Target Press Announcement. Target confirms unauthorized access to payment card data in u.s. stores, December 2013. Online at `http://pressroom.target.com/news/target-confirms-unauthorized-access-to-payment-card-data-in-u-s-stores`.

[11] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, et al. Above the clouds: A berkeley view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.

[12] Adam Back, Ulf Möller, and Anton Stiglic. Traffic analysis attacks and trade-offs in anonymity providing systems. In *Information Hiding*, pages 245–257. Springer, 2001.

[13] James Ball, Bruce Schneier, and Glenn Greenwald. Nsa and gchq target tor network that protects anonymity of web users, October 2013. Online at `http://www.theguardian.com/world/2013/oct/04/nsa-gchq-attack-tor-network-encryption`.

[14] Robert Beverly, Simson Garfinkel, and Greg Cardwell. Forensic carving of network packets and associated data structures. *digital investigation*, 8:S78–S89, 2011.

[15] Stefanie Beyer, Martin Mulazzani, Sebastian Schrittwieser, Markus Huber, and Edgar Weippl. Towards fully automated digital alibis with social interaction. In *Tenth Annual IFIP WG 11.9 International Conference on Digital Forensics*, 1 2014.

[16] Stevens Le Blond, Pere Manils, Chaabane Abdelberi, Mohamed Ali Dali Kaafar, Claude Castelluccia, Arnaud Legout, and Walid Dabbous. One bad apple spoils the bunch: exploiting p2p applications to trace and profile tor users. *Proceedings of the 4th USENIX conference on Large-scale exploits and emergent threats*, 2011.

[17] Nikita Borisov, Ian Goldberg, and Eric Brewer. Off-the-record communication, or, why not to use pgp. In *Proceedings of the 2004 ACM workshop on Privacy in the electronic society*, pages 77–84. ACM, 2004.

[18] Frank Breitinger, Georgios Stivaktakis, and Harald Baier. Frash: A framework to test algorithms of similarity hashing. *Digital Investigation*, 10:S50–S58, 2013.

[19] D Brezinski and Tom Killalea. Guidelines for evidence collection and archiving. *Request For Comments*, 3227, 2002.

[20] Elie Bursztein, Chinmay Soman, Dan Boneh, and John C Mitchell. Sessionjuggler: secure web login from an untrusted terminal using session hijacking. In *Proceedings of the 21st international conference on World Wide Web*, pages 321–330. ACM, 2012.

[21] Brian Carrier. *File system forensic analysis*, volume 3. Addison-Wesley Boston, 2005.

[22] Eoghan Casey and Gerasimos J Stellatos. The impact of full disk encryption on digital forensics. *ACM SIGOPS Operating Systems Review*, 42(3):93–98, 2008.

[23] A. Castiglione, G. Cattaneo, R. De Prisco, A. De Santis, and K. Yim. How to forge a digital alibi on Mac OS X. *Multidisciplinary Research and Practice for Information Systems*, pages 430–444, 2012.

[24] Aniello Castiglione, Giuseppe Cattaneo, Giancarlo De Maio, Alfredo De Santis, Gerardo Costabile, and Mattia Epifani. The forensic analysis of a false digital alibi. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, pages 114–121. IEEE, 2012.

[25] Ellick Chan, Shivaram Venkataraman, Francis David, Amey Chaugule, and Roy Campbell. Forenscope: A framework for live forensics. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 307–316. ACM, 2010.

[26] Jeremy Clark and Paul C. van Oorschot. Sok: Ssl and https: Revisiting past challenges and evaluating certificate trust model enhancements. 2013.

[27] MI Cohen. Pyflag–an advanced network forensic framework. *digital investigation*, 5:S112–S120, 2008.

[28] Josh Constine. Dropbox now has 175 million users, up from 100m in november 2012, July 2013. Online at `http://techcrunch.com/2013/07/09/dropbox-dbx-conference/`.

[29] John W Creswell. Educational research: Planning, conducting, and evaluating quantitative and qualitative research. 2002.

[30] George Danezis and Richard Clayton. Introducing traffic analysis, 2007.

[31] A. De Santis, A. Castiglione, G. Cattaneo, G. De Maio, and M. Ianulardo. Automated construction of a false digital alibi. *Availability, Reliability and Security for Business, Enterprise and Health Information Systems*, pages 359–373, 2011.

[32] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[33] Dan DeFelippi. Dropship on github, April 2011. Online at `https://github.com/driverdan/dropship`.

[34] Tim Dierks. The transport layer security (tls) protocol version 1.2. 2008.

[35] Whitfield Diffie and Martin Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22(6):644–654, 1976.

[36] Roger Dingledine and Nick Mathewson. Anonymity Loves Company: Usability and the Network Effect. In *Proceedings of the Fifth Workshop on the Economics of Information Security (WEIS 2006)*, June 2006.

[37] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[38] Idilio Drago, Marco Mellia, Maurizio M Munafo, Anna Sperotto, Ramin Sadre, and Aiko Pras. Inside dropbox: understanding personal cloud storage services. In *Proceedings of the 2012 ACM conference on Internet measurement conference*, pages 481–494. ACM, 2012.

[39] Thai Duong and Juliano Rizzo. Here come the XOR Ninjas. *Ekoparty*, 2011.

[40] Thai Duong and Juliano Rizzo. The crime attack. *Ekoparty*, 2012.

[41] Zakir Durumeric, James Kasten, Michael Bailey, and Alex Halderman. Analysis of the HTTPS Certificate Ecosystem. In *Internet Measurement Conference (IMC)*, 2013.

[42] Donald Eastlake and Paul Jones. Secure hash algorithm 1 (sha1), 2001.

[43] Peter Eckersley. How Unique is Your Web Browser? In *Privacy Enhancing Technologies*, pages 1–18. Springer, 2010.

[44] E. ECMAScript, European Computer Manufacturers Association, et al. ECMAScript Language Specification. Online at `http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf`.

[45] Noah Edelstein. Feature evolution and support for the skype desktop api, Nov 2013. Online at `http://blogs.skype.com/2013/11/06/feature-evolution-and-support-for-the-skype-desktop-api/`.

[46] Manuel Egele, David Brumley, Yanick Fratantonio, and Christopher Kruegel. An empirical study of cryptographic misuse in android applications. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 73–84. ACM, 2013.

[47] Sascha Fahl, Marian Harbach, Thomas Muders, Matthew Smith, Lars Baumgärtner, and Bernd Freisleben. Why eve and mallory love android: An analysis of android ssl (in) security. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 50–61. ACM, 2012.

[48] Simson Garfinkel. Lessons learned writing digital forensics tools and managing a 30tb digital evidence corpus. *Digital Investigation*, 9:S80–S89, 2012.

[49] Simson Garfinkel, Paul Farrell, Vassil Roussev, and George Dinolt. Bringing science to digital forensics with standardized forensic corpora. *digital investigation*, 6:S2–S11, 2009.

[50] Simson Garfinkel, Alex Nelson, Douglas White, and Vassil Roussev. Using purpose-built functions and block hashes to enable small block and sub-file forensics. *digital investigation*, 7:S13–S23, 2010.

[51] Simson L Garfinkel. Automating disk forensic processing with sleuthkit, xml and python. In *Systematic Approaches to Digital Forensic Engineering, 2009. SADFE'09. Fourth International IEEE Workshop on*, pages 73–84. IEEE, 2009.

[52] Simson L Garfinkel. Digital forensics research: The next 10 years. *Digital Investigation*, 7:S64–S73, 2010.

[53] Simson L Garfinkel. Digital media triage with bulk data analysis and< i> bulk_extractor</i>. *Computers & Security*, 2012.

[54] Simson L Garfinkel and Abhi Shelat. Remembrance of data passed: A study of disk sanitization practices. *Security & Privacy, IEEE*, 1(1):17–27, 2003.

[55] Behrad Garmany and Tilo Müller. Prime: private rsa infrastructure for memory-less encryption. In *Proceedings of the 29th Annual Computer Security Applications Conference*, pages 149–158. ACM, 2013.

[56] Dan Goodin. Stop using nsa-influenced code in our products, rsa tells customers, Sep 2013. Online at `http://arstechnica.com/security/2013/09/stop-using-nsa-influence-code-in-our-product-rsa-tells-customers/`.

[57] Johannes Gotzfried and Tilo Muller. Armored: Cpu-bound encryption for android-driven arm devices. In *Availability, Reliability and Security (ARES), 2013 Eighth International Conference on*, pages 161–168. IEEE, 2013.

[58] J. Alex Halderman, Seth D. Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A. Calandrino, Ariel J. Feldman, Jacob Appelbaum, and Edward W. Felten. Lest we remember: Cold boot attacks on encryption keys. In *Proceedings of the USENIX Security Symposium*. USENIX Association, 2008.

[59] Shai Halevi, Danny Harnik, Benny Pinkas, and Alexandra Shulman-Peleg. Proofs of ownership in remote storage systems. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 491–500. ACM, 2011.

[60] Danny Harnik, Benny Pinkas, and Alexandra Shulman-Peleg. Side channels in cloud services: Deduplication in cloud storage. *Security & Privacy, IEEE*, 8(6):40–47, 2010.

[61] Andrew Hoog. *Android forensics: investigation, analysis and mobile security for Google Android*. Access Online via Elsevier, 2011.

[62] Nicholas Hopper, Luis von Ahn, and John Langford. Provably secure steganography. *Computers, IEEE Transactions on*, 58(5):662–676, 2009.

[63] Amir Houmansadr, Chad Brubaker, and Vitaly Shmatikov. The parrot is dead: Observing unobservable network communications. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, 2013.

[64] Markus Huber, Martin Mulazzani, Manuel Leithner, Sebastian Schrittwieser, Gilbert Wondracek, and Edgar Weippl. Social snapshots: Digital forensics for online social networks. In *Proceedings of the 27th Annual Computer Security Applications Conference*, pages 113–122. ACM, 2011.

[65] Markus Huber, Martin Mulazzani, and Edgar Weippl. Who on earth is "mr. cypher": Automated friend injection attacks on social networking sites. In *Security and Privacy–Silver Linings in the Cloud*, pages 80–89. Springer, 2010.

[66] Markus Huber, Martin Mulazzani, Edgar Weippl, Gerhard Kitzler, and Sigrun Goluch. Friend-in-the-middle attacks: Exploiting social networking sites for spam. *Internet Computing, IEEE*, 15(3):28–34, 2011.

[67] Markus Huber, Martin Mulazzani, and Edgar R. Weippl. Tor http usage and information leakage. In *Proceedings of IFIP CMS 2010*, pages 245–255. Springer, 2010.

[68] IEEE. Code of ethics. Online at `http://www.ieee.org/about/corporate/governance/p7-8.html`.

[69] Aaron Johnson, Chris Wacek, Rob Jansen, Micah Sherr, and Paul Syverson. Users get routed: traffic correlation on tor by realistic adversaries. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 337–348. ACM, 2013.

[70] Stefan Katzenbeisser, Fabien AP Petitcolas, et al. *Information hiding techniques for steganography and digital watermarking*, volume 316. Artech house Norwood, 2000.

[71] Karen Kent, Suzanne Chevalier, Tim Grance, and Hung Dang. Guide to integrating forensic techniques into incident response. *NIST Special Publication 800-86*, 2006.

[72] Dhiru Kholia and Przemysław Wegrzyn. Looking inside the (drop) box. In *Proceedings of the 7th USENIX conference on Offensive Technologies (WOOT)*. USENIX Association, 2013.

[73] Jesse Kornblum. Identifying almost identical files using context triggered piecewise hashing. *digital investigation*, 3:91–97, 2006.

[74] Brian Krebs. Adobe breach impacted at least 38 million users. Online at `http://krebsonsecurity.com/2013/10/adobe-breach-impacted-at-least-38-million-users/`.

[75] Adam Langley. Overclocking ssl. imperial violet blog, june 25, 2010. Online at `https://www.imperialviolet.org/2010/06/25/overclocking-ssl.html`.

[76] Tobias Lauinger, Veikko Pankakoski, Davide Balzarotti, and Engin Kirda. Honeybot, your man in the middle for automated social engineering. In *LEET'10, 3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats, San Jose*, 2010.

[77] Marc Liberatore, Robert Erdely, Thomas Kerle, Brian Neil Levine, and Clay Shields. Forensic investigation of peer-to-peer file sharing networks. *digital investigation*, 7:S95–S103, 2010.

[78] Karsten Loesing. Measuring the tor network from public directory information. *Proc. HotPETS, Seattle, WA*, 2009.

[79] Karsten Loesing, Steven J Murdoch, and Roger Dingledine. A case study on measuring statistical data in the tor anonymity network. In *Financial Cryptography and Data Security*, pages 203–215. Springer, 2010.

[80] Lodovico Marziale, Golden G Richard III, and Vassil Roussev. Massive threading: Using gpus to increase the performance of digital forensics tools. *digital investigation*, 4:73–81, 2007.

[81] Damon McCoy, Kevin Bauer, Dirk Grunwald, Tadayoshi Kohno, and Douglas Sicker. Shining light in dark places: Understanding the tor network. In *Privacy Enhancing Technologies*, pages 63–76. Springer, 2008.

[82] Cade Metz. Dropbox snuffs open code that bypassed file-sharing controls, April 2011. Online at `http://www.theregister.co.uk/2011/04/26/dropbox_moves_to_squash_open_source_dropship_project/`.

[83] Hooman Mohajeri Moghaddam, Baiyu Li, Mohammad Derakhshani, and Ian Goldberg. Skypemorph: Protocol obfuscation for tor bridges. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 97–108. ACM, 2012.

[84] Sean Morrissey. *iOS Forensic Analysis: for iPhone, iPad, and iPod touch*. Apress, 2010.

[85] Keaton Mowery, Dillon Bogenreif, Scott Yilek, and Hovav Shacham. Fingerprinting information in javascript implementations. In *Proceedings of Web*, volume 2, 2011.

[86] Keaton Mowery and Hovav Shacham. Pixel perfect: Fingerprinting canvas in html5. *Proceedings of W2SP*, 2012.

[87] Martin Mulazzani. Anonymity & monitoring. *Master thesis, Vienna University of Technology*, 2009.

[88] Martin Mulazzani, Markus Huber, and Edgar R. Weippl. Anonymity and monitoring: How to monitor the infrastructure of an anonymity system. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 40(5):539–546, 9 2010.

[89] Martin Mulazzani, Sebastian Neuner, Peter Kieseberg, Markus Huber, Sebastian Schrittwieser, and Edgar R. Weippl. Quantifying windows file slack in size and stability. In *Ninth Annual IFIP WG 11.9 International Conference on Digital Forensics*, 1 2013.

[90] Martin Mulazzani, Philipp Reschl, Markus Huber, Manuel Leithner, Sebastian Schrittwieser, and Edgar R. Weippl. Fast and reliable browser identification with javascript engine fingerprinting. In *Web 2.0 Workshop on Security and Privacy (W2SP)*, 5 2013.

[91] Martin Mulazzani, Sebastian Schrittwieser, Manuel Leithner, Markus Huber, and Edgar R. Weippl. Dark clouds on the horizon: Using cloud storage as attack vector and online slack space. In *Proceedings of the USENIX Security Symposium*. USENIX Association, 2011.

[92] Martin Mulazzani, Thomas Unger, Edgar R. Weippl, Sebastian Schrittwieser, Markus Huber, and Dominik Frühwirt. Shpf: Enhancing http(s) session security with browser fingerprinting. In *Proceedings of the Eighth International Conference on Availability, Reliability and Security (ARES)*, 9 2013.

[93] Tilo Müller, Andreas Dewald, and Felix C Freiling. Aesse: a cold-boot resistant implementation of aes. In *Proceedings of the Third European Workshop on System Security*, pages 42–47. ACM, 2010.

[94] Tilo Müller, Felix C Freiling, and Andreas Dewald. Tresor runs encryption securely outside ram. In *USENIX Security Symposium*, 2011.

[95] Tilo Müller and Michael Spreitzenbarth. Frost – forensic recovery of scrambled telephones. In *Applied Cryptography and Network Security*, pages 373–388. Springer, 2013.

[96] Steven J Murdoch and George Danezis. Low-cost traffic analysis of tor. In *Security and Privacy, 2005 IEEE Symposium on*, pages 183–195. IEEE, 2005.

[97] Derek Newton. Dropbox authentication: insecure by design, April 2011. Online at `http://dereknewton.com/2011/04/dropbox-authentication-static-host-ids/`.

[98] Nick Nikiforakis, Alexandros Kapravelos, Wouter Joosen, Christopher Kruegel, Frank Piessens, and Giovanni Vigna. Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In *IEEE Symposium on Security and Privacy*, 2013.

[99] The open web application security Project. Owasp top 10. 2013.

[100] Andrea Peterson, Barton Gellman, and Ashkan Soltani. Yahoo to make ssl encryption the default for webmail users. finally., Oct. 2013. Online at `http://www.washingtonpost.com/blogs/the-switch/wp/2013/10/14/yahoo-to-make-ssl-encryption-the-default-for-webmail-users-finally/`.

[101] Jon Postel. Internet protocol. *Request For Comments*, 791, 1981.

[102] Jon Postel. Transmission control protocol. *Request For Comments*, 793, 1981.

[103] Kevin Poulsen. *Kingpin: How one hacker took over the billion-dollar cybercrime underground*. Random House Digital, Inc., 2012.

[104] Angelo Prado, Neal Harris, and Yoel Gluck. SSL, gone in 30 seconds - a BREACH beyond CRIME. 2013.

[105] Darren Quick and Kim-Kwang Raymond Choo. Dropbox analysis: Data remnants on user machines. *Digital Investigation*, 2013.

[106] Scott Renfro. Secure browsing by default, Jul. 2013. Online at https://www.facebook.com/notes/facebook-engineering/secure-browsing-by-default/10151590414803920.

[107] Thorsten Ries, Andriy Panchenko, Thomas Engel, et al. Comparison of low-latency anonymous communication systems: practical usage and performance. In *Proceedings of the Ninth Australasian Information Security Conference-Volume 116*, pages 77–86. Australian Computer Society, Inc., 2011.

[108] Ronald Rivest. The md5 message-digest algorithm. *Request For Comments*, 1321, 1992.

[109] Ohad Rodeh, Josef Bacik, and Chris Mason. Btrfs: The linux b-tree filesystem. *Trans. Storage*, 9(3), August 2013.

[110] Vassil Roussev. Data fingerprinting with similarity digests. In *Advances in Digital Forensics VI*, pages 207–226. Springer, 2010.

[111] Vassil Roussev. An evaluation of forensic similarity hashes. *digital investigation*, 8:S34–S41, 2011.

[112] Vassil Roussev and Candice Quates. File fragment encoding classification—an empirical approach. *Digital Investigation*, 10:S69–S77, 2013.

[113] Joanna Rutkowska. Beyond the cpu: Defeating hardware based ram acquisition. *Proceedings of BlackHat DC 2007*, 2007.

[114] Sam Schillace. Default https access for gmail. Online at http://gmailblog.blogspot.co.at/2010/01/default-https-access-for-gmail.html.

[115] Bruce Schneier. Applied cryptography, 1996.

[116] Dan Shumow and Niels Ferguson. On the possibility of a back door in the nist sp800-90 dual_ec_prng. *27th Annual International Cryptology Conference (CRYPTO)*, 2007. Informal presentation.

[117] Cliord P Stoll. The cuckoo's egg: Tracing a spy through the maze of computer espionage, 1989.

[118] Johannes Stüttgen and Michael Cohen. Anti-forensic resilient memory acquisition. *Digital Investigation*, 10:S105–S115, 2013.

[119] Twitter. Securing your twitter experience with https, Feb. 2012. Online at https://blog.twitter.com/2012/securing-your-twitter-experience-with-https.

[120] Chris Wacek, Henry Tan, Kevin Bauer, and Micah Sherr. An empirical evaluation of relay selection in tor. In *Proceedings of the Network and Distributed Security Symposium (NDSS)(February 2013)*, 2013.

[121] Qiyan Wang, Xun Gong, Giang TK Nguyen, Amir Houmansadr, and Nikita Borisov. Censorspoofer: asymmetric communication using ip spoofing for censorship-resistant web browsing. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 121–132. ACM, 2012.

[122] Samuel D Warren and Louis D Brandeis. The right to privacy. *Harvard law review*, 4(5):193–220, 1890.

[123] Zachary Weinberg, Jeffrey Wang, Vinod Yegneswaran, Linda Briesemeister, Steven Cheung, Frank Wang, and Dan Boneh. Stegotorus: a camouflage proxy for the tor anonymity system. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 109–120. ACM, 2012.

[124] Kleine Zeitung Wolfsberg. Facebook rückte daten über neonazi heraus, December 2011. Online at `http://www.kleinezeitung.at/kaernten/wolfsberg/2911087/facebook-rueckte-daten-ueber-neonazi-heraus.story`.

[125] York Yannikos, Lukas Graner, Martin Steinebach, and Christian Winter. On the availability of data corpora for digital forensic education and research. In *Tenth Annual IFIP WG 11.9 International Conference on Digital Forensics*, 1 2014.

[126] York Yannikos, Christian Winter, and Markus Schneider. Synthetic data creation for forensic tool testing: Improving performance of the 3lspg framework. In *Availability, Reliability and Security (ARES), 2012 Seventh International Conference on*, pages 613–619. IEEE, 2012.

[127] Ting-Fang Yen, Yinglian Xie, Fang Yu, Roger Peng Yu, and Martin Abadi. Host Fingerprinting and Tracking on the Web: Privacy and Security Implications. In *Proceedings of the 19th Annual Network and Distributed System Security Symposium (NDSS 2012)*, February 2012.

[128] Joel Young, Simson Garfinkel, Kristina Foster, and Kevin Fairbanks. Distinct sector hashes for target file detection. 2012.

[129] Yihua Zhang and Marina Blanton. Efficient dynamic provable possession of remote data via balanced update trees. In *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*, pages 183–194. ACM, 2013.

[130] Qingji Zheng and Shouhuai Xu. Secure and efficient proof of storage with deduplication. In *Proceedings of the second ACM conference on Data and Application Security and Privacy*, pages 1–12. ACM, 2012.

Dark Clouds on the Horizon: Using Cloud Storage as Attack Vector and Online Slack Space

The paper *Dark Clouds on the Horizon: Using Cloud Storage as Attack Vector and Online Slack Space* was published at the USENIX Security Symposium 2011 in San Francisco[25].

You can find the paper online at `http://static.usenix.org/events/sec11/tech/full_papers/Mulazzani6-24-11.pdf`. USENIX also offers video recordings[26] as well as the slides of the presentation[27] online.

[25] `https://www.usenix.org/conference/usenixsecurity11`

[26] `https://www.usenix.org/conference/usenix-security-11/dark-clouds-horizon-using-cloud-storage-attack-vector-and-online-slack`

[27] `http://static.usenix.org/events/sec11/tech/slides/mulazzani.pdf`

Fast and Reliable Browser Identification with JavaScript Engine Fingerprinting

The paper *Fast and Reliable Browser Identification with JavaScript Engine Fingerprinting* was published at the Web 2.0 Workshop on Security and Privacy (W2SP) 2013 co-located with the 2013 IEEE Symposium on Security and Privacy in San Francisco[28].

You can find the paper online at `http://www.w2spconf.com/2013/papers/s2p1.pdf`. The slides of the presentation are available online as well[29].

[28] `http://www.w2spconf.com/2013/`

[29] `http://www.w2spconf.com/2013/slides/s2p1-slides.pdf`

SHPF: Enhancing HTTP(S) Session Security with Browser Fingerprinting

The paper *SHPF: Enhancing HTTP(S) Session Security with Browser Fingerprinting* was published at the Eighth International Conference on Availability, Reliability and Security (ARES) 2013 in Regensburg[30].

You can find the paper online at `http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6657249`. An extended preprint is available on the author's homepage at `http://www.sba-research.org/wp-content/uploads/publications/shpf_extendedPreprint.pdf`.

[30] `http://www.ares-conference.eu/ares2013/www.ares-conference.eu/conf/index.html`

Anonymity and Monitoring: How to Monitor the Infrastructure of an Anonymity System

The paper *Anonymity and Monitoring: How to Monitor the Infrastructure of an Anonymity System* was published in the IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Volume 40, Issue 5 in 2010[31].

You can find the paper online at http://dx.doi.org/10.1109/TSMCC.2010.2045372. A preprint is available on the author's homepage at http://www.sba-research.org/wp-content/uploads/publications/IEEE_SMC_Tor_finalPreprint.pdf

[31] http://www.ieeesmc.org/publications/

Tor HTTP Usage and Information Leakage

The paper *Tor HTTP Usage and Information Leakage* was published at the 11th Joint IFIP TC6 and TC11 Conference on Communications and Multimedia Security (CMS) 2010 in Linz[32].

You can find the paper online at `http://dx.doi.org/10.1007/978-3-642-13241-4_22`. A preprint is available on the author's homepage at `http://www.sba-research.org/wp-content/uploads/publications/2010-Huber-TorHTTPUsage.pdf`.

[32] `http://www.cms2010.net/`

Quantifying Windows File Slack Size and Stability

The paper *Quantifying Windows File Slack Size and Stability* was published at the Ninth Annual IFIP WG 11.9 International Conference on Digital Forensics 2013 in Orlando, Florida[33].

You can find the paper online at `http://dx.doi.org/10.1007/978-3-642-41148-9_13`. A preprint is available on the author's homepage at `http://www.sba-research.org/wp-content/uploads/publications/ifipSlack_2013_preprint.pdf`

[33] `http://www.ifip119.org/Conferences/`

Towards Fully Automated Digital Alibis with Social Interaction

The paper *Towards Fully Automated Digital Alibis with Social Interaction* was published at the Tenth Annual IFIP WG 11.9 International Conference on Digital Forensics 2014 in Vienna, Austria[34].

A preprint is available on the author's homepage at `http://www.sba-research.org/wp-content/uploads/publications/alibigenerator_preprint.pdf`.

[34] `http://www.ifip119.org/Conferences/`