Digital Investigation 18 (2016) S76–S86

Contents lists available at ScienceDirect

Digital Investigation

journal homepage: www.elsevier.com/locate/diin

DFRWS USA 2016 — Proceedings of the 16th Annual USA Digital Forensics Research Conference

Time is on my side: Steganography in filesystem metadata

Sebastian Neuner [a, *], Artemios G. Voyiatzis [a], Martin Schmiedecker [a], Stefan Brunthaler [a], Stefan Katzenbeisser [b], Edgar R. Weippl [a]

[a] SBA Research, Vienna, Austria
[b] Technische Universität Darmstadt, Germany

Keywords:
Digital forensics
Data hiding
Steganography
Storage forensics
File system forensics
Real-world data corpus

A B S T R A C T

We propose and explore the applicability of file timestamps as a steganographic channel. We identify an information gap between storage and usage of timestamps in modern operating systems that use high-precision timers. Building on this, we describe a layered design of a steganographic system that offers stealthiness, robustness, and wide applicability. The proposed design is evaluated through theoretical, evidence-based, and experimental analysis for the case of NTFS using datasets comprising millions of files. We report a proof-of-concept implementation and confirm that the embedded information is indistinguishable from that of a normal filesystem use. Finally, we discuss the digital forensics analysis implications of this new information-hiding technique.

© 2016 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Introduction

The need for protected information exchange and storage in the digital world is constantly increasing. Cryptographic techniques can provide information confidentiality, authenticity, and integrity. However, they do leave evidence of the information exchange.

Steganographic techniques are able to hide the existence of information passing through communication channels or resting in storage media for later access. These techniques are useful in a wide range of real-world scenarios, including but not limited to: circumventing censorship and restrictions imposed by governments and other adversaries (Akgül and Kırlıdoğ; Anderson, 2012), assisting whistleblowers when disclosing documents (Greenwald, 2014), and supporting businesses to protect strategic corporate information during transmission (Cox et al., 2007).

Numerous steganographic techniques have been proposed and analyzed in the research literature (Zielińska et al., 2014). The analysis focuses on criteria such as the achieved secrecy on specific application scenarios, the steganographic channel capacity, and the information channel utilization.

Storage or format-oriented steganographic techniques hide information in logical channels by utilizing redundant or unused fields in format specifications. This includes, among others, the master boot record (MBR) of non-bootable hard disks and the unused disk space caused by the misalignment of hard disk sector size and file size (Khan et al., 2011).

Modern filesystems support a wealth of operations that span beyond the primitive of mapping files into sequences of hard disk sectors. The filesystem specifications define additional data structures (i.e., “metadata”) to describe information like the owner, the access permissions, and the date and time when important file events took place.

In this paper, we propose and explore, to the best of our knowledge for the first time in literature, the applicability

* Corresponding author.
E-mail addresses: sneuner@sba-research.org (S. Neuner), avoyiatzis@sba-research.org (A.G. Voyiatzis), mschmiedecker@sba-research.org (M. Schmiedecker), sbrunthaler@sba-research.org (S. Brunthaler), katzenbeisser@seceng.informatik.tu-darmstadt.de (S. Katzenbeisser), eweippl@sba-research.org (E.R. Weippl).

http://dx.doi.org/10.1016/j.diin.2016.04.010
1742-2876/© 2016 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

of *filesystem timestamps* as a steganographic channel. More specifically, we make the following contributions:

1. We analyze the granularity of the timestamps that modern filesystems implement, and we evaluate their applicability for steganographic applications.
2. We propose the use of timestamps as a means to hide information in NTFS and other filesystems with sub-second timestamp granularity.
3. We describe a system design and a proof-of-concept implementation that support different levels of possible capacity to securely hide data on NTFS volumes.
4. We validate the proposed system using real-world and synthetic datasets, and we show that the embedded steganographic information cannot be distinguished from the information produced by normal filesystem operations.
5. We discuss the digital forensics implications of this new steganographic method.

The rest of this paper is organized as follows: Section (Background) provides a literature review on steganography with emphasis on storage artefacts. Section (Timestamps in modern filesystems) analyzes the use of timestamps on modern filesystems. Section (Steganography based on file timestamps) proposes a novel steganographic channel based on file timestamps. Section (Evaluation of the TOMS system) evaluates the security of the proposed system. Section (Experimental system validation) describes a proof-of-concept implementation aiming at NTFS filesystems. Section (Implications for forensics analysis) discusses the implications on the digital forensics process. Finally, Section (Conclusions and future work) concludes the paper and presents the future directions of our work.

Background

Data hiding

Early works on digital steganography focused on hiding data in the clear, deriving and discussing different methods of embedding data, and arguing how steganography is and probably will be used in the present and in the near future (Katzenbeisser and Petitcolas, 2000; Zielińska et al., 2014). Such works did not anticipate the widespread use of the personal digital devices and the role of the Internet in our daily lives (Franz et al., 1996; Anderson and Petitcolas, 1998).

A considerable amount of research was devoted to embedding unobservable communication within normal network traffic, ranging from the utilization of TCP/IP timestamps (Giffin et al., 2003) to the more general usage of TCP/IP fields (Murdoch and Lewis, 2005). Many implementations of steganography hide encrypted data in innocent-looking network traffic (e.g., ptunnel (Stodle)), header fields (Rutkowska, 2004), or use timing intervals and artificial transmission delays for information transmission (Lee et al., 2014; Berk et al., 2005; Mazurczyk and Lubacz, 2010). While it has been shown that secure steganographic protocols are feasible, we are still lacking functional implementations and widespread use of such tools (Hopper et al., 2009).

A second line of research focused on embedding unobservable information within the contents of stored files, introducing undetectable degradation of multimedia quality (e.g., manipulating the low-significance bits of pixel representation in images (Bailey and Curran, 2006)), the color palettes in GIF images (Fridrich and Du, 2000), or (possibly) encoding information in YouTube videos that look like static snow (Williams, 2015).

Filesystems

A plethora of different filesystems is available, including FAT and NTFS for Microsoft-Windows-based devices, ext4 and btrfs for GNU/Linux systems, and HFS+ for Apple OS X and iOS devices.[1] Most of them store different artefacts at various levels of granularity and detail, collectively known as “filesystem metadata”.

Filesystem metadata can be classified in five categories: *file system*, *application*, *file name*, *content*, and *generic* metadata (Carrier, 2005). *File system* metadata are information on how the filesystem is to be read and where the important data structures reside. *Application* metadata are information useful for the application utilizing the filesystem, such as the file owner and the file access permissions. *File name* metadata are information for the human-readable names mapping to logical data locations. *Content* metadata are information about the logical addressing of the files, the file allocation status, and the actual data of the files. *Generic* metadata are information mostly used internally by the filesystem for its operations. This includes information such as the timestamps of various events in the lifecycle of a file.

Steganography using filesystem metadata

The topic of hiding data in filesystem metadata was heavily discussed in the late 1990s (Anderson et al., 1998). Back then, export restrictions on the use of strong cryptographic algorithms outside the USA were in place, and there was an increased concern by the public regarding key escrow. StegFS, a steganographic filesystem compatible with the Linux ext2 filesystem, was developed (McDonald and Kuhn, 2000; Pang et al., 2003). This filesystem achieved plausible deniability of the hidden content thanks to its indistinguishability from unused content. This behavior was achieved by applying encryption on the content under the assumption that a good encryption algorithm ensures that encrypted data appear as random data. However, the use of StegFS is not undetectable as the needed filesystem driver is not hidden. Additionally, there is no integrity check of the data. Thus, StegFS cannot recover from any kind of intrusive data modifications.

Encoding (hiding) information in the order that a filesystem indexes the file names is explored in (Aycock and

[1] An exhaustive list is provided in the Wikipedia entry available at https://en.wikipedia.org/wiki/Comparison_of_file_systems.

Download English Version:

https://daneshyari.com/en/article/10341449

Download Persian Version:

https://daneshyari.com/article/10341449

Daneshyari.com