Quantifying Windows File Slack in Size and Stability*

Martin Mulazzani, Sebastian Neuner, Peter Kieseberg,
Markus Huber, Sebastian Schrittwieser, Edgar Weippl
SBA Research, Vienna
[1stletterfirstname][lastname]@sba-research.org

Abstract

In digital forensics, different forms of slack space can be used to hide information from either the operating system or other users, or both. While some forms are easily detectable others are very subtle, and require an experienced forensic investigator to discover the hidden information. The exact amount of information that can be hidden varies with the form of slack space used, as well as environmental parameters like file system block size or partition alignment. While some methods for slack space can be used to hide arbitrary amounts of information, file slack has tighter constraints and was thought to be rather limited in space.

In this paper we evaluate how much file slack space modern operating systems offer by default and how stable it is over time with special regards to system updates. In particular we measure the file slack for 18 different versions of Microsoft Windows using NTFS. We show that many files of the operating systems are rather static and do not change much on disk during updates, and are thus highly suitable for hiding information. We furthermore introduce a model for investigators to estimate the total amount of data that can be hidden in file slack for file systems of arbitrary size.

Keywords: Digital Forensics, Slack Space

*This is the author's preprint for the Ninth Annual IFIP WG 11.9 International Conference on Digital Forensics. The original publication is available at www.springerlink.com

1 Introduction

With ever increasing hard drive and storage capacities, slack space is now more than ever an interesting topic in digital forensics. Hard drives with 3 terabytes and more are commodity hardware now, which leaves plenty of space to hide information and making manual forensic analysis time-consuming and cumbersome. While encryption can be used and is used to make certain or all information on a hard drive inaccessible during forensic investigations [5] (e.g., Truecrypt [1], dm-crypt, FileVault or Bitlocker [13]), slack space and steganography hide information more or less in plain sight.

While special areas for hiding data like the *Host Protected Area* (HPA) and *Device Configuration Overlay* (DCO) on the hard drive can be easily detected with current tools e.g., *The Sleuth Kit (TSK)* [4] or *Encase* [3], it is much harder for an investigator to find data that was hidden in file slack space on purpose. During the natural life-cycle of files on hard drives, file slack is constantly created and overwritten, thus not necessarily stable. Multiple tools have been published that allow to hide information in file slack, even encrypted. This paper is the first work towards quantifying file slack space of common operating systems, and how slack space is affected by system updates. We focus our work on different versions of Microsoft Windows, as it is in widespread use on clients and servers. We used Windows versions from the last 10 years that are still supported by Microsoft and receive operating system updates, starting with Windows XP and Server 2003. We also included the most recent versions at the time of writing, Windows 8 and Windows Server 2012 RC. We furthermore propose a model that allows an investigator to estimate the amount of file slack capacity across all files on a hard drive, thus including files created by the user. In particular, the contributions of this paper are as follows:

- We quantify file slack space in a multitude of Microsoft operating systems.
- We evaluate the durability of OS-specific file slack space regarding system updates.
- We present a simple model to estimate the total possible file slack for investigators.

[1] `http://www.truecrypt.org/`

The rest of the paper is organized as follows: Section 2 briefly explains different forms of slack space and their background in file systems, and why file slack is particularly interesting for forensic investigations. Our method for measuring file slack space is presented in Section 3. Section 4 evaluates how much information can be hidden in commodity operating systems, and how stable OS-specific file slack is over time. Results are discussed in Section 5, as well as our model for estimating file slack space. Related work is presented in Section 6 before we conclude in Section 7.

2 Background

File slack is defined as the space between the end of the file, and the end of the allocated cluster [4]. Slack space is a byproduct of operating systems: to reduce addressing overhead they cluster multiple sectors of a hard drive together, and implicitly create file slack space for every file that does not aligns in size with the cluster size. The size of usable slack space is in general depending on the cluster size of the filesystem, the sector size of the underlying hard drive, and the padding strategy used by the operating system. Hard drives used to have a sector size of 512 bytes, with modern hard drives having a sector size of 4k. FAT32 usually has a cluster size between 4k and 32k (depending on the actual partition size), while NTFS usually has a cluster size of 4k for drives < 16 TB [14]. Windows uses padding only for the last sector at the end of each file, while the other sectors in the cluster are left untouched [4], possibly leaving up to $n - 1$ sectors untouched if the portion of the file in the last cluster (with n sectors) is smaller than the sector size.

Numerous other forms of slack space (like volume slack or partition slack) or places to hide data on a hard drive (e.g., HPA and DCO) can be encountered in a forensic investigation [2, 10, 6]. The benefit of file slack, however, is that it exists in any case on every hard drive, and can be extended in size by storing a large number of files and/or using large cluster sizes (NTFS for example supports a cluster size of up to 64 KB). File slack is also the reason why bitwise copies have to be created during image acquisition [11], as otherwise the content of file slack would be lost for the investigation.

For a forensic investigation the slack space is of interest in two cases: if the suspect used a secure deletion software like *shred* or *WipeFile* for wiping

files, slack space can contain fragments of previous data. In the second case, the suspect might have hidden data in slack space on purpose, using freely available tools. Several tools have been developed to facilitate easy storage of information in slack space, among them *slacker.exe* for NTFS from the Metasploit anti-forensic toolkit, or *bmap* for Linux.

3 Quantifying OS File Slack Space

To analyze file slack in NTFS for different operating systems we used *fiwalk* [7] by Simson Garfinkel which is now part of *SleuthKit*, and which is based on the DFXML language for digital forensics [8]. For our evaluation we used 18 different versions of Microsoft Windows, as it is still by far the most predominantly operating system in use. We evaluated client and server versions alike, and used different service pack levels as starting point to measure slack file persistency. Each system used the NTFS default cluster size of 4KB with underlying 512 bytes sector size and was installed using the default settings. We then patched each installed system with the available system updates and available service packs, including optional updates like the .NET framework and additional language packs, and ran *fiwalk* on the acquired disk images. The full list of operating systems and the total number of security updates and service packs that have been installed during our evaluation can be seen in Table 1.

Our scripts parsed the XML outputs from *fiwalk*[2], and calculated the slack space for each disk image. During calculating the usable file slack capacity we omitted NTFS file system metadata objects like the $MFT or $BitClus, and files that are less than a full disc cluster in size to avoid files that are possibly resident in the NTFS master file table ($MFT) and thus not directly usable for file slack. For calculating file slack persistency we checked the SHA-1 hash values as well as the file's timestamp for last write access before and after the updates. Again, we only included files that use more than one disc cluster in size.

[2]Note to the reviewers: we will release our datasets and scripts online, and will include the link here once the paper is accepted for publication

Operating System	Upd.	SPs	Operating System	Upd.	SPs
Windows XP Pro.	+189	+2	Windows 7 Pro. SP1	+106	—
Windows XP Pro. SP2	+164	+1	Windows 7 Ent.	+212	+1
Windows XP Pro. SP3	+177	—	Windows 7 Ent. SP1	+167	—
Vista Business	+246	+2	Windows 8 RC	+11	—
Vista Business SP1	+72	+1	Server 2003 R2 Std. SP2	+163	—
Vista Business SP2	+143	—	Server 2003 R2 Ent. SP2	+167	—
Vista Ent. SP1	+207	+1	Server 2008 R2 Std.	+148	+1
Vista Ent. SP2	+143	—	Server 2008 R2 Std. SP1	+103	—
Windows 7 Pro.	+156	+1	Server 2012 RC	+6	—

Table 1: List of evaluated operating systems and their updates

4 Evaluation

For our evaluation we first quantify file slack in Windows before we analyze file slack stability during system updates.

4.1 Quantification of OS Slack Space

The amount of available file slack space depends on the number of files and thus on the complexity of the underlying operating system. The base installation of Windows XP (the oldest system in our evaluation) can be used to hide about 22 megabytes of data in file slack on average, the newest versions of Windows at the time of writing (i.e., Windows 8 and Server 2012 RC) can be used to hide 86 and 70 megabytes respectively. Windows Vista in its different versions (Business vs. Enterprise) allows approximately 62 megabytes in file slack, and Windows 7 (Professional and Enterprise) a little more than 69 megabytes. A similar trend is observable for Windows server: Windows Server 2003 R2 has approximately 20 megabytes in file slack capacity, while Server 2008 R2 has about 73 megabytes.

The amount of file slack increases tremendously with system updates, especially with service packs, as the old system files are usually kept referenced in the file system in case something goes wrong during the update process. Table 1 shows which version of Windows has received a service pack since its release, and the number of system updates that have been installed for our evaluation. Many system updates affect only a small number of files, while service packs are major updates. An increase in file slack capacity of more than 100% during the lifetime of an operating system is not uncommon, on

average the slack space doubles in total. At the end of the evaluation process Windows XP had more than 30 megabytes on average, Vista 105 megabytes and Windows 7 Professional 100 megabytes on average. Windows 7 Enterprise showed an exceptional increase of more than 500%, from around 72 megabytes to more than 400 megabytes. Windows 8 and Server 2012 RC were stable, as there are not many updates available yet. The detailed results can be seen in Table 4.2.

The details of the cumulative slack space are visualized in Figure 1: we use a standard box-plot over all collected states in the evaluation process, grouped by the product lines. This graph can be used by forensic examiners in assessing the amount of possible file slack for a given operating system. Depending on the install date of the underlying operating system, one hundred to two hundred megabytes of file slack space is possible for newer operating systems, even with the default cluster size of 4k in NTFS. We will discuss the amount of file slack for larger cluster sizes in Section 5. Note that the number of samples (respectively update steps) for the different product lines was as follows: Windows XP had 15, Vista had 32, Windows 7 had 19, 8 RC had 4. For the Windows Server: Server 2003 R2 had 10, Server 2008 R2 had 9 and Server 2012 RC had 4.

4.2 Stability of OS Slack Space

To assess the stability of file slack in the files of the operating system we compared SHA-1 hash values as well as the timestamps for each file in the images. While the number of files increased heavily (especially due to service packs), the stability of the initial file slack was high: in some cases more than 90% of the slack areas were not modified, and the amount of stable file slack was 50 megabytes and more. While the older versions of Windows i.e., XP and Server 2003 had 20 megabytes and less that persisted all system updates, Vista, Windows 7 and Server 2008 had 50 megabytes and more. On average and across all different Windows versions, 44 megabytes and 78% of initial file slack were still available at the end of the evaluation process. This means that up to two third of the file slack is still available even today e.g., for Windows XP SP2, which was already released in 2004. For Server 2003 R2 we measured that up to 80% would be still available today. Vista SP2 from 2009 as well as Windows 7 SP1 from 2011 kept more than 90% of their files

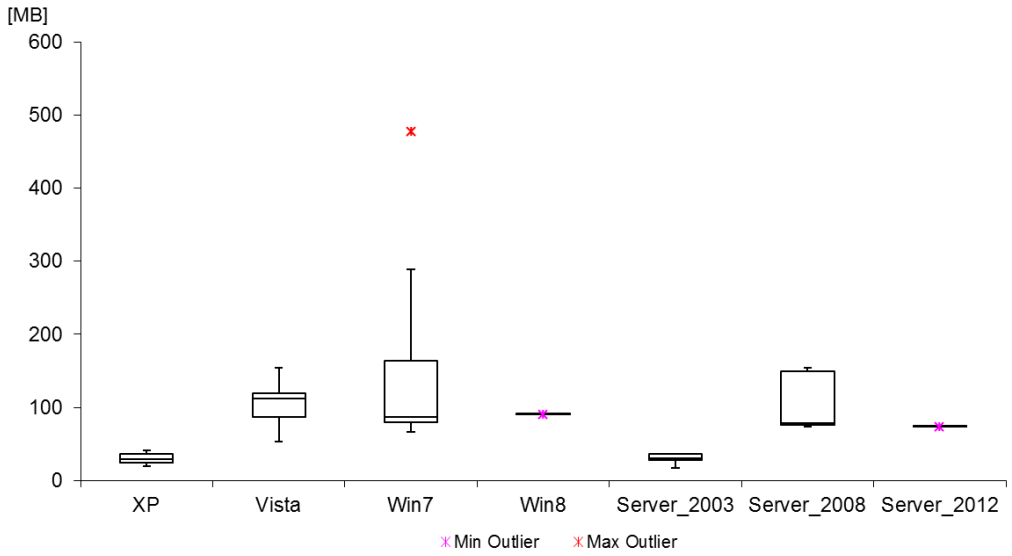

Figure 1: Box plot of quantified file slack space

intact.

File slack stability for a selection of different Windows versions and the corresponding number of update steps is shown in Figure 2. Again, the details of our data can be found in Table 4.2. Please note that even though the total file slack capacity increases with the system updates and update steps, the stability of file slack from the initial installation is represented as a monotonically decreasing function as the update can change existing files, but not add files to the previous.

5 Discussion

All tested operating systems rely on a large number of files for storage: while the earliest versions of Windows XP had only a little more than 10.000 files that are bigger then the default cluster size of 4k, Windows 7 has already more than 40.000 and Windows 8 more than 55.000. The number of files increases heavily with service packs: Windows Vista, which started with 35.000

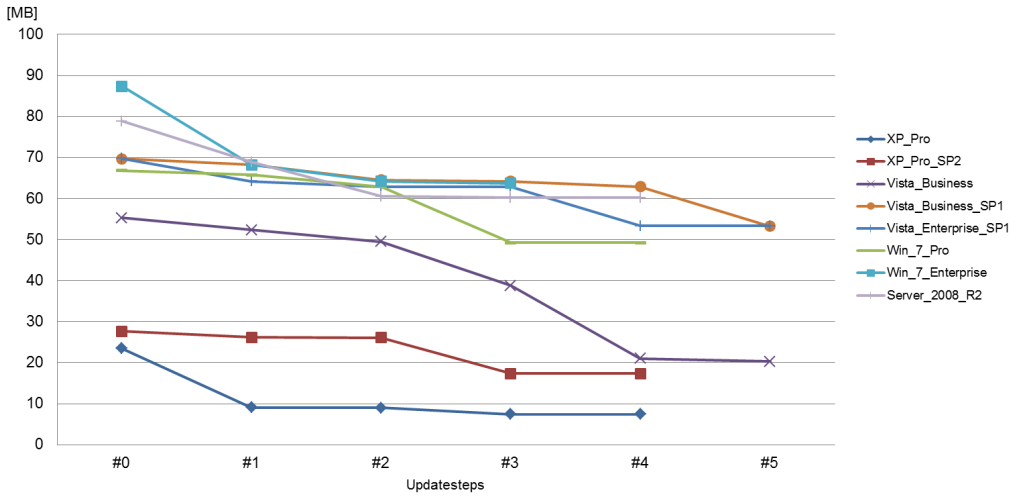

Figure 2: Stability of file slack across updates

files had more than 90.000 after installing two service packs. For forensic investigators this is particularly interesting as on one hand, the amount of information that could possibly be extracted from these files can be large. On the other hand these vast number of files offer a lot of slack space that can be used with readily available tools to hide information.

An adversary that actively uses file slack could furthermore increase the cluster size of NTFS - up to 64kb are supported by NTFS. A quick estimation by running our scripts on a xml file chosen at random showed that increasing the cluster size to 32kb on the same Windows Vista with more than 90.000 files could increase file slack to 1.25 gigabytes. This is 8.11 times larger compared to the 150 megabytes in our results with the default cluster size of 4kb, just using the files from the operating system itself. Forensic investigators should thus pay attention to artificially large cluster sizes, as they can be instrumented to hide possibly very large amount of data in the slack space.

Operating System	Initial Slack	Final Slack	Stability
Windows XP Pro.	22.36 MB	36.97 MB (165%)	7.09 MB/31.7%
Windows XP Pro. SP2	26.31 MB	29.49 MB (112%)	16.49 MB/62.7%
Windows XP Pro. SP3	18.72 MB	23.15 MB (124%)	14.21 MB/75.9%
Vista Business	52.70 MB	147.13 MB (279%)	19.34 MB/36.7%
Vista Business SP1	66.49 MB	119.89 MB (180%)	50.77 MB/76.4%
Vista Business SP2	50.89 MB	82.99 MB (163%)	48.13 MB/94.7%
Vista Ent. SP1	66.51 MB	140.35 MB (211%)	50.82 MB/76.4%
Vista Ent. SP2	71.73 MB	113.76 MB (159%)	67.36 MB/93.9%
Windows 7 Pro.	63.71 MB	115.16 MB (181%)	46.96 MB/73.7%
Windows 7 Pro. SP1	65.03 MB	77.80 MB (120%)	60.73 MB/93.4%
Windows 7 Ent.	83.33 MB	454.62 MB (546%)	60.74 MB/72.9%
Windows 7 Ent. SP1	65.10 MB	381.56 MB (586%)	60.77 MB/93.3%
Windows 8 RC	86.40 MB	87.06 MB (101%)	65.10 MB/75.3%
Server 2003 R2 Std. SP2	24.42 MB	33.90 MB (140%)	20.13 MB/82.4%
Server 2003 R2 Ent. SP2	16.55 MB	35.13 MB (212%)	15.20 MB/91.8%
Server 2008 R2 Std.	75.16 MB	146.80 MB (195%)	57.43 MB/76.4%
Server 2008 R2 Std. SP1	69.82 MB	73.03 MB (105%)	69.19 MB/99.1%
Server 2012 RC	70.16 MB	70.58 MB (101%)	70.01 MB/99.8%

Table 2: Detailed results of our file slack quantification

5.1 A Model for File Slack Space Estimation

Following we will give an estimation on the expected size of the slack space. Let n be the number of files and k be the cluster size (i.e. the number of sectors inside a cluster). Furthermore, s denotes the number of bytes inside a sector. Since only the last cluster of a file may not be filled completely (i.e. all clusters of a file except for one do not provide slack space), the slack space that can be expected from one file is completely independent from the actual size of the file. Thus we arrive at the following expectation S_E for the average slack space of a file: $S_E = s \cdot \mathbb{E}(X)$, where $\mathbb{E}(X)$ is the expectancy with respect to the underlying statistical distribution of the fill-grade of the last cluster.

In general it can be expected that the fill-grade of the last cluster is equally distributed among all possible values. Since the first sector inside a cluster is always filled in the case of the NTFS, the number of free sectors

that can be used lies in $\{1, \ldots, k-1\}$, thus yielding

$$S_{(n,s,k)} = n \cdot s \cdot \mathbb{E}(X) = n \cdot s \cdot \sum_{x=1}^{k-1} x \cdot \frac{1}{k} = n \cdot s \cdot \frac{1}{k} \cdot \frac{(k-1)k}{2} = n \cdot s \cdot \frac{k-1}{2}$$

Using a typical sector size of 512 bytes (s) and 8 sectors per cluster (k), we can reduce this formula to $S_n = 1792 \cdot n$.

5.2 Limitations

Our approach for measuring file slack has some limitations: for one, we deliberately ignored the fact that modern hard drives have a default sector size of 4 kb. This was necessary to be able to include operating systems still in widespread use (like Windows XP) that are yet completely unaware of the fact that hard drives can have larger sectors [15]. We also ignored user activity and did not simulate it, as we came to the conclusion that there is no sufficient method to simulate user activity in a realistic fashion. We also ignored additional files that may be found on a PC, either by the user or by software installed by the user. In particular we did not install any software that might be commonly found on Windows PCs or additional services for the Windows Servers to mimic real world systems, in particular Microsoft Office, alternative browsers like Chrome or Firefox, or services like IIS or a SQL server. Thus our results cannot be considered to be set in stone, but should be considered a conservative upper bound for OS-specific file system slack, and a conservative lower bound when including user data.

6 Related Work

In recent work regarding slack space, file fragmentation was used to hide data in FAT partitions [12]. FragFS on the other hand used slack space in $MFT entries to hide data, as $MFT entries have a fixed length of usually 1024 bytes per entry. In their estimation, a total of about 36 megabytes of data could be hidden in $MFT slack [17] in systems running Windows XP, as most $MFT entries use less than 450 bytes on average. Slack space in cloud environments has been recently discussed, as it can be used to retrieve possibly sensitive data like deleted SSH keys [1] or to hide data without leaving traces at the client or in the log files of the service operator [16].

Slack space was also found to be not securely deleted on the second hand market of hard drives [9].

6.1 Future Work

For future work we want to address the limitations discussed above, and plan to conduct a large-scale survey to measure slack space in real-world systems. So far we have not evaluated user files and how stable they are over time. We also plan to extend our evaluations to other modern operating systems and file systems that use clustering of sectors, in particular HFS+ on OS X and ext4 on Linux.

7 Conclusion

In this paper we have evaluated the amount and stability of file slack in NTFS-based operating systems. We especially considered system updates, and to what extend the file slack persists and evaluated it with 18 different versions of Microsoft Windows. All operating systems that we tested offered initial slack space in the tens of megabytes, which was largely immune to system updates: on average 44 megabytes, respectively 78%, of initial file slack was available after installing all available system updates, even full service packs. Especially with newer versions of Windows like Windows 7 or Server 2008, between 100 and 200 megabytes were available with the default cluster size of 4 kb.

Acknowledgements

This research was funded by the Austrian Research Promotion Agency under COMET K1, as well as Grant No. 825747.

References

[1] M. Balduzzi, J. Zaddach, D. Balzarotti, E. Kirda, and S. Loureiro. A security analysis of amazon's elastic compute cloud service. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pages 1427–1434. ACM, 2012.

[2] H. Berghel. Hiding data, forensics, and anti-forensics. *Commun. ACM*, 50(4):15–20, Apr. 2007.

[3] S. Bunting and W. Wei. *EnCase Computer Forensics: The Official EnCE: EnCaseCertified Examiner Study Guide*. Sybex, 2006.

[4] B. Carrier. *File system forensic analysis*. Addison-Wesley Professional, 2005.

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

[6] S. Garfinkel. Anti-forensics: Techniques, detection and countermeasures. In *The 2nd International Conference on i-Warfare and Security (ICIW)*, pages 77–84, 2007.

[7] S. 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.

[8] S. Garfinkel. Digital forensics XML and the DFXML toolset. *Digital Investigation*, 2011.

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

[10] E. Huebner, D. Bem, and C. Wee. Data hiding in the ntfs file system. *Digital Investigation*, 3(4):211–226, 2006.

[11] K. Kent, S. Chevalier, T. Grance, and H. Dang. Guide to integrating forensic techniques into incident response. *NIST Special Publication*, pages 800–86, 2006.

[12] H. Khan, M. Javed, S. Khayam, and F. Mirza. Designing a cluster-based covert channel to evade disk investigation and forensics. *Computers & Security*, 30(1):35–49, 2011.

[13] J. D. Kornblum. Implementing bitlocker drive encryption for forensic analysis. *Digital Investigation*, 5(3):75–84, March 2009.

[14] Microsoft. Default cluster size for NTFS, FAT, and exFAT. Online at `http://support.microsoft.com/kb/140365`, retrieved Sept. 2012.

[15] Microsoft. Information about Microsoft support policy for large-sector drives in Windows. Online at `http://support.microsoft.com/kb/2510009`, retrieved Okt. 2012.

[16] M. Mulazzani, S. Schrittwieser, M. Leithner, M. Huber, and E. Weippl. Dark clouds on the horizon: Using cloud storage as attack vector and online slack space. In *USENIX Security*, volume 8, 2011.

[17] I. Thompson and M. Monroe. Fragfs: An advanced data hiding technique. *BlackHat Federal, January.[online] http://www.blackhat. com/presentations/bh-federal-06/BH-Fed-06-Thompson/BH-Fed-06-Thompson-up. pdf*, 2006.