Quantifying Windows File Slack Size and Stability

Martin Mulazzani, Sebastian Neuner, Peter Kieseberg, Markus Huber, Sebastian Schrittwieser, Edgar Weippl

Chapter 13

QUANTIFYING WINDOWS FILE SLACK SIZE AND STABILITY

Martin Mulazzani, Sebastian Neuner, Peter Kieseberg, Markus Huber, Sebastian Schrittwieser and Edgar Weippl

Abstract Slack space can be used to hide data from the operating system and other users. While some forms of data hiding are easily detectable, others are subtle and require an experienced forensic practitioner to discover the hidden data. The amount of data that can be hidden varies with the type of slack space and environmental parameters such as filesystem block size and partition alignment. This paper evaluates the amount of file slack space available in Windows systems and the stability of slack space over time with respect to system updates. Measurements of the file slack for eighteen versions of Microsoft Windows with the NTFS filesystem reveal that many of the files change very little during system updates and are, thus, highly suitable for hiding data. A model is presented for estimating the amount of data that can be hidden in the file slack space of Windows filesystems of arbitrary size.

Keywords: Forensic analysis, file slack space, Windows systems, data hiding

1. Introduction

With ever increasing hard drive and memory storage capacities, the examination of slack space has become an important component of digital forensic investigations. Hard drives with 3 TB or more capacity are commonly available, which leaves plenty of space to hide data and make manual disk forensic analysis cumbersome and time-consuming. While encryption can be used to make hard drive data inaccessible during a forensic examination [4], slack space data hiding and steganography [8] conceal data more or less in plain sight.

Data hidden in certain areas of a hard drive, such as the host protected area (HPA) and device configuration overlay (DCO), can be detected by current tools. However, it is more difficult to find data that was intentionally hidden in file slack space. During the natural life cycle of files on a hard drive, file slack is constantly created and overwritten, and is, thus, not necessarily stable. Moreover, several tools have been developed to hide data – including encrypted data – in file slack space.

To help address the problem of data hiding in slack space, this paper quantifies the file slack space for Microsoft Windows operating systems and how the slack space is affected by system updates. The analysis uses eighteen versions of Microsoft Windows, including Windows XP and Server 2003 through Windows 8 and Server 2012 RC. Based on the analysis, a model is presented for estimating the amount of file slack space across all files on a hard drive.

2. Background

File slack space is a byproduct of operating system filesystem design. To reduce addressing overhead in a filesystem, multiple sectors of a hard drive are clustered together; this implicitly creates "slack space" for every file whose size does not match the cluster size exactly. File slack is defined as the space between the end of the file and the end of the allocated cluster [3]. The size of usable slack space depends on the cluster size of a filesystem, the sector size of the hard drive and the padding strategy used by the operating system. Older hard drives commonly have a sector size of 512 bytes while newer hard drives have a sector size of 4 KiB. FAT32 usually has a cluster size between 4 KiB and 32 KiB depending on the partition size, while NTFS usually has a cluster size of 4 KiB for drives with less than 16 TB capacity [11]. Windows uses padding only for the last sector at the end of each file. The other sectors in the cluster are left untouched [3], 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.

Other forms of slack space (e.g., volume slack and partition slack) and data hiding locations (e.g., HPA and DCO) are also encountered in forensic examinations [2, 7]. The benefit of file slack, however, is that it exists on every hard drive that uses a filesystem that clusters sectors. Also, the file slack size can be extended arbitrarily by storing a large number of small files and/or using large cluster sizes (NTFS, for example, supports a cluster size of up to 64 KiB). File slack is also one of the reasons why bitwise copies have to be created during image acquisition [9]. Otherwise, the data stored in file slack space could be lost.

During a forensic examination, file slack space is of interest in two particular cases. The first is when secure deletion software such as

Table 1. Evaluated operating systems.

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	—

shred or WipeFile has been used to securely delete files, but the slack space may still contain fragments of previous data. In the second case, data has been intentionally hidden in slack space using freeware such as slacker.exe for NTFS or bmap for Linux.

3. Quantifying OS File Slack Space

SleuthKit's fiwalk [5] tool was used to analyze file slack in NTFS for eighteen versions of Microsoft Windows. Each system used the NTFS default cluster size of 4 KiB with the underlying 512 byte sector size, and was installed using the default settings. Each installed system was then patched with the available system updates and service packs, including optional updates such as the .NET framework and additional language packs. After each change, fiwalk was executed on the acquired disk images. Table 1 lists the complete set of operating systems and the total numbers of system updates and service packs that were installed during the evaluation.

Scripts were used to parse the fiwalk XML outputs and calculate the slack space for each disk image. The calculations omitted NTFS filesystem metadata objects (e.g., $MFT and $Bitmap) and files less than a full cluster in size to avoid files that were possibly resident in the NTFS master file table ($MFT) and, thus, were not directly usable as file slack. The file slack persistence calculation compared the file SHA-1 hash values and timestamps of the last write accesses before and after the updates, assuming that modifying a file could destroy file slack (by increasing or decreasing the file size above the sector boundaries in the last cluster). This was done to prevent overestimation and provide conservative measurements. The data set is available at www.sba-research.org/team/researchers/martin-mulazzani.

4. Experimental Evaluation

This section quantifies the file slack available in Windows systems and analyzes the stability of file slack space during system updates.

4.1 Quantification of Slack Space

The amount of available file slack space depends on the number of files and the complexity of the underlying operating system. The experimental results show that more recent operating systems have a larger file base and, therefore, a larger quantity of file slack space. The base installation of Windows XP, the oldest system in the evaluation, can be used to hide about 22 MB of data in file slack while the newest versions of Windows, Windows 8 and Server 2012 RC, can hide 86 MB and 70 MB, respectively. Windows Vista (Business and Enterprise versions) allows approximately 62 MB in file slack while Windows 7 (Professional and Enterprise) allows a little more than 69 MB. A similar trend is observed for Windows server: Windows Server 2003 R2 has about 20 MB of file slack capacity while Server 2008 R2 has about 73 MB of capacity.

One of the key observations is that the amount of file slack increases with system updates, especially with service packs. This is because old system files are retained in the event that something goes wrong during the update process. Table 1 shows the number of system updates that were installed. Many system updates affected only a small number of files, while service packs, which are major updates, can affect many files. An increase in file slack capacity of more than 100% during the lifetime of an operating system is not uncommon; on the average, the slack space doubles in capacity. At the end of the evaluation process, Windows XP had more than 30 MB, Vista 105 MB and Windows 7 Professional 100 MB. Windows 7 Enterprise showed an exceptional increase of more than 500%, from around 72 MB to more than 400 MB. Windows 8 and Server 2012 RC were stable as there were few updates available. Table 2 presents the detailed results.

Figure 1 presents a box plot of the cumulative slack space over all the collected states in the evaluation grouped by product line. This graph can be used by forensic practitioners to assess the amount of possible file slack for a given operating system. Depending on the install date of the operating system, 100 MB to 200 MB of file slack space is not uncommon for newer operating systems such as Windows 7 and Server 2008, even with the default cluster size of 4 KiB in NTFS. The numbers of samples (respectively update steps) for the different product lines were: 15 for Windows XP, 32 for Vista, 19 for Windows 7, and four for 8 RC. In the

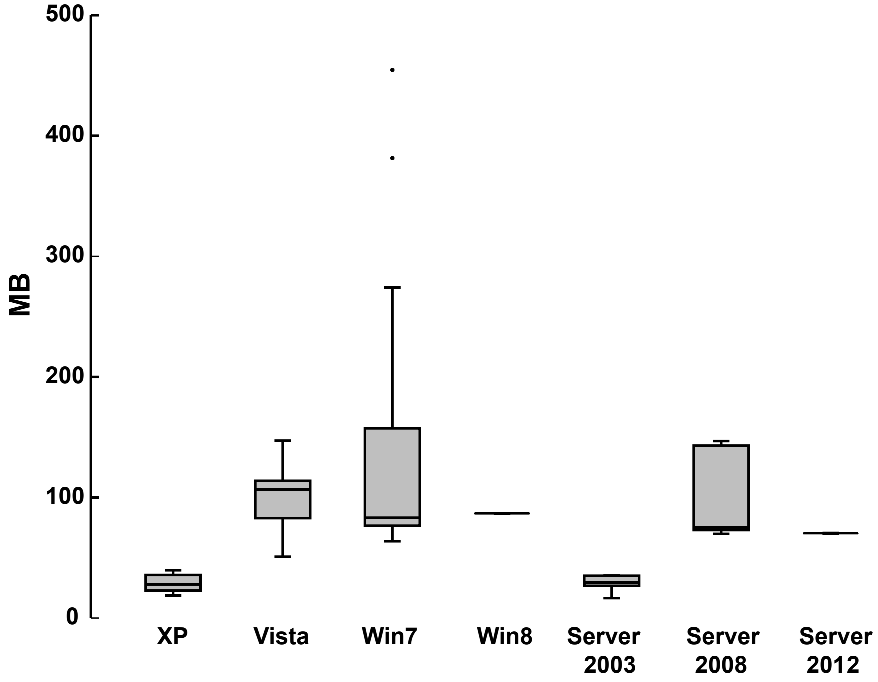

Figure 1. Box plot of quantified file slack space.

case of the Windows servers, Server 2003 R2 had ten, Server 2008 R2 had nine and Server 2012 RC had four.

4.2 Stability of Slack Space

The file slack stability evaluation compared the SHA-1 hash values and timestamps for each file in the images before and after updates. While the numbers of files increased substantially (especially with service packs), the stability of the initial file slack was high – in some cases, more than 90% of the slack space from the initial install was not modified. Also, while the older versions of Windows XP and Server 2003 had 20 MB or less that was persistent with respect to all available system updates, Vista, Windows 7 and Server 2008 had 50 MB or more slack space that was persistent. On the average and across the different Windows versions, 44 MB and 78% of the initial file slack were available at the end of the evaluation process. This means that up to two-thirds of the file slack created during installation was not modified by the end of the analysis. For Server 2003 R2, up to 80% was available. Vista SP2 from 2009 and Windows 7 SP1 from 2011 had more than 90% of their files intact. Further data from real systems in use is needed to confirm these

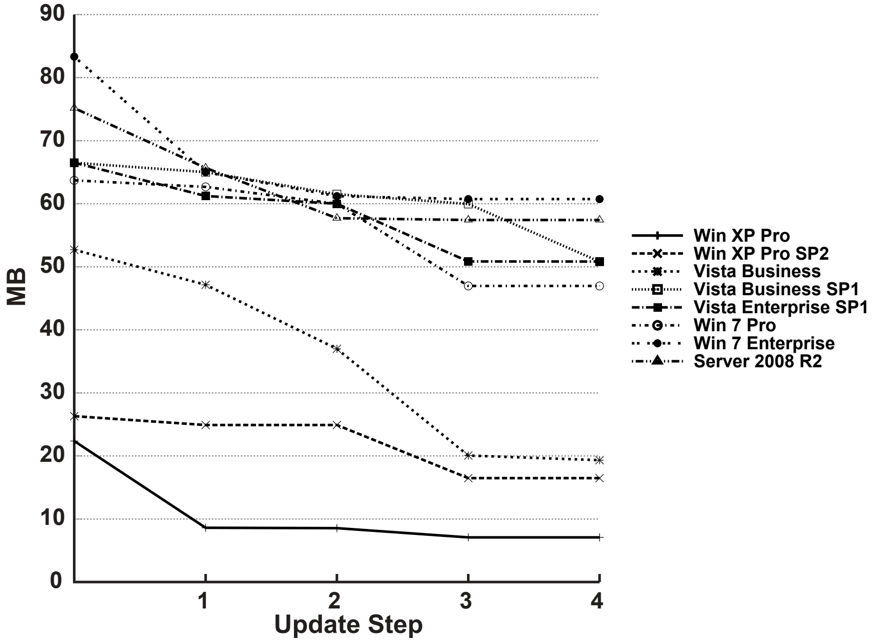

Figure 2. Stability of file slack space across update steps.

results. Figure 2 shows the file slack stability for a selection of Windows versions and the corresponding numbers of update steps. Table 2 shows the detailed results.

Note that, although the total file slack capacity increases with system updates and update steps, the stability of file slack from the initial installation is represented as a monotonically decreasing function. This is because updates can change existing files without adding new files.

Table 3 shows the top file extensions and the most stable file types with respect to slack space for three Windows versions. The number per file extension is the number of files of the type that persisted through all system updates, keeping their slack space untouched.

The most stable files were program files (`.dll`, `.exe`, `.sys`); pictures (`.png`, `.jpg`); and text files (`.chm`, `.htm`). For the three representative images, the first operating system, Windows XP Pro SP2, had a total of 11,723 persistent files, Windows 7 Pro had 29,561 and Server 2008 R2 had 36,394. The slack space per file type was not calculated, but the formula for slack space capacity estimation, presented in the next section, can be used to compute this value.

Table 2. Detailed results of file slack quantification.

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 3. Top file types with respect to slack space stability.

XP Pro SP2		**Win7 Pro**		**Server 2008 R2**	
`.dll`	4,414	`.dll`	6,302	`.dll`	7,303
`.exe`	1,106	`.mui`	3,906	`.cat`	6,752
`.sys`	793	`.est`	3,190	`.est`	4,204
`.inf`	692	`.inf`	1,352	`.mui`	3,907
`.pnf`	674	`.gpd`	1,303	`.exe`	1,364
`.chm`	317	`.exe`	1,067	`.gpd`	1,303
`.htm`	233	`.png`	1,051	`.inf`	1,160
`.nls`	192	`.cat`	945	`.mum`	909
`.jpg`	168	`.pnf`	914	`.ppd`	806
Total	8,607	Total	20,030	Total	27,708

4.3 Discussion

While the earliest versions of Windows XP had little more than 10,000 files that were larger then the default cluster size of 4 KiB, Windows 7 had more than 40,000 such files and Windows 8 had more than 55,000. The number of files increases significantly with service packs: Windows Vista, which started with 35,000 files, had more than 90,000 files after installing two service packs. For forensic practitioners, this is particularly important because it appears that a considerable amount of slack space is available to hide data.

An adversary that actively uses file slack could further increase the cluster size of NTFS – up to 64 KiB is supported by NTFS. A quick estimate obtained by running our scripts on an `.xml` file chosen at random showed that increasing the cluster size to 32 KiB on Windows Vista with more than 90,000 files could increase the file slack to 1.25 GB. This is more than eight times larger than the 150 MB available for the default cluster size of 4 KiB. Therefore, forensic practitioners should pay close attention to artificially large cluster sizes, especially because they are uncommon and can be instrumented to hide a large amount of data in the file slack.

The approach for measuring file slack has some limitations. First, the tests ignored the fact that modern hard drives have a default sector size of 4 KiB. This was necessary in order to evaluate commonly used operating systems such as Windows XP that do not support hard drives with larger sectors [12]. Second, user activity was excluded because there is no adequate method that simulates user activity in a realistic fashion. Third, additional files that are commonly found in Windows systems due to software installed by users were excluded. Because of these limitations, the results represent a conservative upper bound for operating-system-specific filesystem slack and a lower bound when including user data.

5. File Slack Space Estimation

This section presents a formula that generalizes the evaluation results and provides a means for a digital forensic practitioner to determine the expected size of the slack space.

Let n be the number of files and k be the cluster size (i.e., number of sectors in a cluster). Furthermore, let s denote the number of bytes per sector. Since only the last cluster of a file may not be filled completely (i.e., all the clusters of a file except for one cluster do not provide slack space), the slack space that can be expected from a single file is completely independent of the actual size of the file. Thus, the expected value S_E of the average slack space of a file is given by:

$$S_E = s\mathbb{E}(X)$$

where $\mathbb{E}(X)$ is the expected value with respect to the underlying statistical distribution of the fill-grade of the last cluster.

In general, it can be assumed 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 NTFS, the number of free sectors that can be used is an element of the set $\{1, \ldots, k-1\}$. This yields:

$$\begin{aligned} S_{(n,s,k)} &= ns\mathbb{E}(X) \\ &= ns\sum_{x=1}^{k-1} x\frac{1}{k} \\ &= ns\frac{1}{k}\frac{(k-1)k}{2} \\ &= ns\frac{k-1}{2}. \end{aligned}$$

Using a typical sector size $s = 512$ bytes and $k = 8$ sectors per cluster, the formula above reduces to:

$$S_n = 1792n.$$

6. Related Work

Recent research related to slack space has examined the use of file fragmentation to hide data in FAT partitions [10]. FragFS [14] uses slack space in $MFT entries to hide data, leveraging the fact that $MFT entries commonly have a fixed length of 1,024 bytes per entry. According to estimates [14], a total of about 36 MB of data can be hidden in $MFT slack of Windows XP systems because each $MFT entry uses less than 450 bytes on the average.

Researchers have studied slack space in cloud environments and in hard drives purchased in the second-hand market. In particular, it is possible to retrieve possibly sensitive data such as deleted SSH keys [1] and to hide data without leaving traces in a cloud client or in the log files of a cloud service provider [13]. Also, Garfinkel and Shelat [6] have shown that slack space is rarely deleted from pre-used hard drives from the second-hand market.

7. Conclusions

Slack space cannot be ignored in digital forensic investigations because it can be used to conceal large amounts of data. The principal contributions of this paper are the quantification of file slack space in eighteen versions of Microsoft Windows, an evaluation of the durability of file slack space under system updates, and a simple model for estimating the total storage capacity of file slack space. All eighteen operating systems that were tested offered initial slack space in the tens of megabytes and this space was largely immune to system updates. On the average, 44 MB or 78% of the initial file slack was available after installing all the available system updates, including complete service packs. In the case of newer versions of Windows, such as Windows 7 and Server 2008, around 100 MB to 200 MB of slack space was available with the default cluster size of 4 KiB.

Our future work will address the limitations discussed in the paper and will conduct a large-scale survey to measure the slack space in real-world systems. Additionally, it will evaluate the amount of slack space available in user files and their stability. We also plan to study modern operating systems and filesystems that use sector clustering, especially HFS+ in OS X systems and ext4 in Linux systems.

Acknowledgement

This research was funded by the Austrian Research Promotion Agency under COMET K1 and by 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, *Proceedings of the Twenty-Seventh Annual ACM Symposium on Applied Computing*, pp. 1427–1434, 2012.

[2] H. Berghel, Hiding data, forensics and anti-forensics, *Communications of the ACM*, vol. 50(4), pp. 15–20, 2007.

[3] B. Carrier, *File System Forensic Analysis*, Pearson Education, Upper Saddle River, New Jersey, 2005.

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

[5] S. Garfinkel, Automating disk forensic processing with Sleuthkit, XML and Python, *Proceedings of the Fourth International IEEE Workshop on Systematic Approaches to Digital Forensic Engineering*, pp. 73–84, 2009.

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

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

[8] S. Katzenbeisser and F. Petitolas (Eds.), *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House, Norwood, Massachusetts, 2000.

[9] K. Kent, S. Chevalier, T. Grance and H. Dang, Guide to Integrating Forensic Techniques into Incident Response, NIST Special Publication 800-86, National Institute of Standards and Technology, Gaithersburg, Maryland, 2006.

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

[11] Microsoft, Default cluster size for NTFS, FAT and exFAT, Redmond, Washington (`support.microsoft.com/kb/140365`), 2002.

[12] Microsoft, Microsoft support policy for 4K sector hard drives in Windows, Redmond, Washington (`support.microsoft.com/kb/2510009`), 2013.

[13] 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, *Proceedings of the Twentieth USENIX Conference on Security*, 2011.

[14] I. Thompson and M. Monroe, FragFS: An advanced data hiding technique, presented at the *BlackHat Federal Conference*, 2006.