DFRWS
DIGITAL FORENSIC RESEARCH CONFERENCE

Using ShellBag Information to Reconstruct User Activities

By

Vrizlynn L. L. Yuandong Zhu, Pavel Gladyshev and Joshua James

From the proceedings of

The Digital Forensic Research Conference

DFRWS 2009 USA

Montreal, Canada (Aug 17th - 19th)

DFRWS is dedicated to the sharing of knowledge and ideas about digital forensics research. Ever since it organized the first open workshop devoted to digital forensics in 2001, DFRWS continues to bring academics and practitioners together in an informal environment.

As a non-profit, volunteer organization, DFRWS sponsors technical working groups, annual conferences and challenges to help drive the direction of research and development.

http:/dfrws.org

DIGITAL INVESTIGATION 6 (2009) S69–S77

available at www.sciencedirect.com

ScienceDirect

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

Digital Investigation

Using shellbag information to reconstruct user activities☆

Yuandong Zhu, Pavel Gladyshev, Joshua James*

Centre for Cybercrime Investigation, University College Dublin, Belfield, Dublin 4, Ireland

Keywords:
Digital forensics
Event reconstruction
Windows XP
ShellBag information analysis
Registry snapshots analysis

ABSTRACT

Built into Microsoft Windows is the ability for the operating system to track user window viewing preferences specific to Windows Explorer. This information, which is called "ShellBag" information, is stored in several locations within the Windows Registry in the Windows Operating System. This paper introduces a novel method to examine ShellBag information within Registry snapshots to reconstruct user activities. It compares different states of ShellBag information within consecutive Registry snapshots in order to detect ShellBag-related user actions. Nine detection rules are proposed on the basis of analyzing the causality between user actions and updated ShellBag information. This approach can be used to prove that certain interactions between the user and system must have, or must not have happened during a certain time period.
© 2009 Digital Forensic Research Workshop. Published by Elsevier Ltd. All rights reserved.

1. Introduction

Built into Microsoft Windows is the ability for the operating system to track user window viewing preferences. These are settings specific to the default Windows shell, Windows Explorer, such as folder window size, position on the screen and others. They allow consistent user experiences between sessions. This information is stored in several locations within the Windows Registry in Windows XP, including two locations in each user's profile:

HKEY_USERS\<USERID>\Software\Microsoft\Windows\Shell
HKEY_USERS\<USERID>\Software\Microsoft\Windows\ShellNoRoam

This information will be referred to as "ShellBag" information throughout the remainder of this paper.

Previously, examining Shellbag information was used to find traces of folders that were once stored in the system (Carvey, 2007). This task is now easily achieved using state of the art forensics tools (AccessData, 2008). This paper will look into another characteristic of ShellBag information; that the information must be created and updated because of specific *user actions*. The method of analysis proposed in this paper reconstructs user activities by reasoning from the causes of different states of ShellBag information within consecutive Registry snapshots. These Registry snapshots are created by default in most versions of Microsoft Windows for the purpose of system recovery (Harder, 2001). The analysis technique described in this paper allows the forensic examiner to prove that certain user action must have, or must not have happened during the time of interest. This information can be useful for the investigation.

1.1. *Contribution*

The contribution of this paper is highlighted as follows:

- A method is proposed to analyze user activity traces in ShellBag information within Registry snapshots. The result of this method is practically applicable in the sense that it proves that certain interactions between the user and system must have, or must not have happened during a certain time period.

☆ This research was funded by the Science Foundation Ireland under Research Frontiers Programme 2007 grant CMSF575.
* *Corresponding author*.
E-mail address: yuandong.zhu@ucd.ie (Y. Zhu).
1742-2876/$ – see front matter © 2009 Digital Forensic Research Workshop. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.diin.2009.06.009

- The method proposed in this paper is a complement to established basic ShellBag analysis techniques. This method could be used with traditional forensics analysis to provide more comprehensive user activity information.
- The proposed method was developed by observing and analyzing causal links between user actions and ShellBag information update patterns. This analysis method will apply to any Windows Operation System that uses the same ShellBag information updating mechanism.

1.2. *Organization*

This paper is organized as follows: Section 2 gives a brief description of ShellBag information. Section 3 describes the results of monitoring ShellBag information updates during experiments in which some of the common user actions were simulated. Section 4 analyzes the causality between user actions and ShellBag information updating. The forensic analysis method including nine user action detection rules are then proposed in Section 5. Finally, a demonstration of the method's practical usability is given in Section 6.

2. Overview of ShellBag information

While ShellBag information exists in several Registry hives, the focus of this paper is on the ShellBag information related to user profiles (HKEY_USERS\<USERID>). This is because ShellBag information stored in an individual user profile is only affected by user actions specific to that account. Hence, any investigative results will only be applicable to this specific account. As mentioned earlier, there are two locations that contain ShellBag information,

HKEY_USERS\<USERID>\Software\Microsoft\Windows\Shell
HKEY_USERS\<USERID>\Software\Microsoft\Windows\ShellNoRoam

From our experiments, the former is used to store information related to remote folders and the latter is about the local folders' information. The structures of these two keys are identical. For simplicity, this paper only discusses the "ShellNoRoam" key.

2.1. *BagMRU key*

The ShellNoRoam key includes two sub-keys, the BagMRU key and Bags key. As illustrated in Fig. 1, the BagMRU key comprises of sub-keys named by numeric value, each of these sub-keys corresponds to a folder in the file system and the BagMRU key itself points to the Desktop folder. All of the sub-folders' keys are organized in a hierarchical fashion as they appear in the file system.

The BagMRU key and its sub-keys store information about folders which were most recently browsed by the user. These keys are a special type of Most Recently Used (MRU) key. As shown in Fig. 2, each of these keys contains

- A list of MRU items denoted by numerical values. Each MRU item is associated with a sub-folder under this key's corresponding folder.

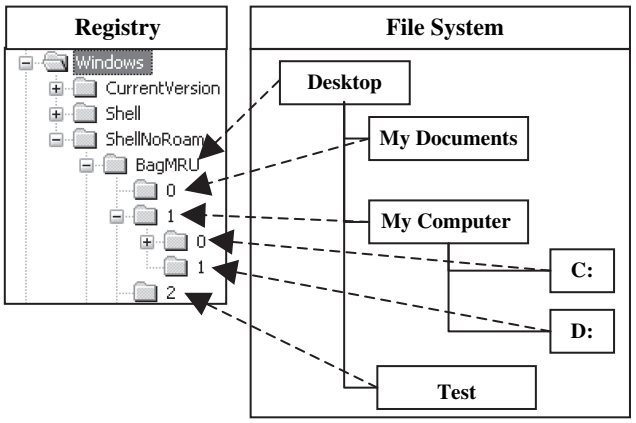

Fig. 1 – The link between the BagMRU key and its sub-keys and file system folders.

- A special value called "MRUListEx" is used to record the sequence of these MRU items. The leftmost four bytes in the "MRUListEx" value list correspond to the most recently updated MRU item in the list. The second four bytes correspond to the second MRU item and so forth.
- A "NodeSlot" value indicates the MRU key's corresponding key under the Bags key.

Each MRU item within the MRU key is related to a particular sub-folder, which could be recognized by the binary data stored in these MRU items. For example as shown in Fig. 3, the MRU item ("2") within the BagMRU key corresponding to the folder "Test" located immediately under the Desktop directory. Apart from the folder's name, the creation, modification and last access time of this folder at the time the MRU item was created were also saved into the MRU item.

The numerical name of the MRU item is also the name of BagMRU's sub-key which is associated with that sub-folder. For instance, in the previous example, the sub-folder "Test" is known to be associated with the MRU item "2" within the BagMRU key, therefore further information about the "Test" folder could be found in the sub-key "BagMRU\2".

2.2. *Bags key*

The structure of the Bags key is different. As shown in Fig. 4, it comprises of several numerically named sub-keys. Under each of these sub-keys, there is a "Shell" key that stores the display settings related to a folder, such as window position, view mode and sort method for items within the window. The maximum number of sub-keys under Bags key is defined in

Name	Type	Data
(Default)	REG_SZ	(value not set)
0	REG_BINARY	44 00 31 00 00 00 00 00 8c 39 0d 79
1	REG_BINARY	2e 00 31 00 00 00 00 00 8e 39 f5 6a
MRUListEx	REG_BINARY	01 00 00 00 00 00 00 00 ff ff ff ff
NodeSlot	REG_DWORD	0x0000035a (858)

Fig. 2 – The structure of the BagMRU key and sub-keys.

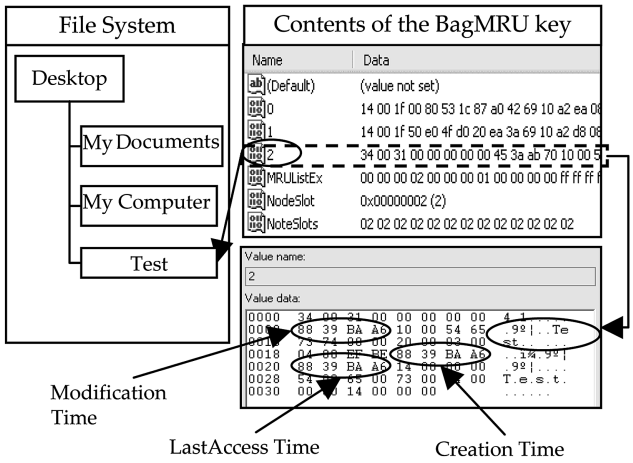

Fig. 3 – The link between the MRU item and associated folder.

the "BagMRU Size" value under the "ShellNoRoam" key. The default number for Window XP operating system is 5000.

The link between the folder and its display settings is established via the "NodeSlot" value within the sub-key under the BagMRU key associated with this folder (Fig. 5).

2.3. *Summary*

BagMRU and Bags keys contain ShellBag information related to folders which were recently accessed by the user. The information related to a particular folder is logically divided into three components,

- Folder's Display key – a key under the Bags key that stores a particular folder's display settings in its sub-key, called "Shell".
- Folder's MRU key – a key that contains the folder's most recently used sequence of sub-folders and the name of the corresponding Display key.
- Folder's MRU item – a value in the parent folder's MRU key that contains the information of this folder and its related MRU key.

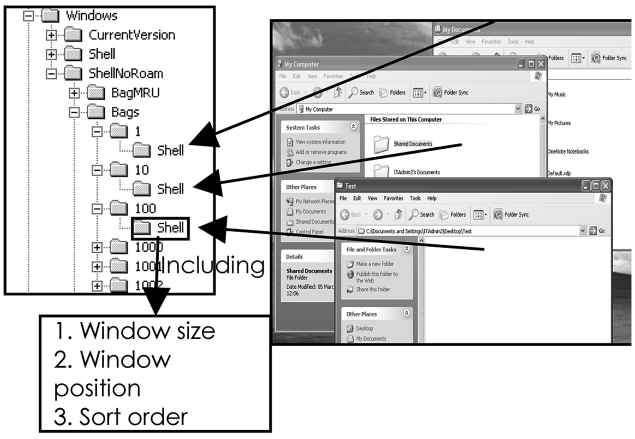

Fig. 4 – The structure of the Bags key.

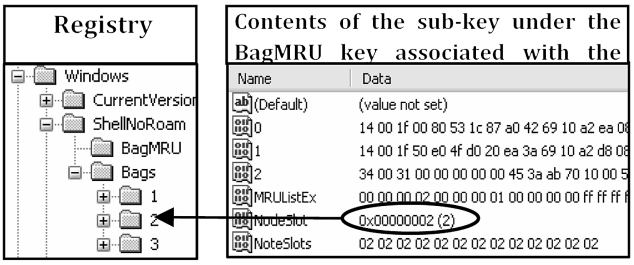

Fig. 5 – The link between folder and associated sub-key under the Bags key.

This ShellBag information is available only for folders that have been opened and closed in Window Explorer at least once.

The data links ShellBag information and file system folder is the MRU item information. To find where a folder associated with a particular MRU item is located, start from examining the BagMRU key. Each MRU item included in the BagMRU key is associated with a child folder of the Desktop directory. Using these MRU items, the sub-key's corresponding folder can be recognized. This process is repeated for the sub-keys in the BagMRU hierarchy until the target MRU item is reached. The process of walking through BagMRU and its sub-keys will give us the path of the folder by the folder name saved in the related MRU items.

3. Experimental analysis of ShellBag information updating

The main objective of this section is to understand how ShellBag information is updated when different user actions occur. In order to achieve this goal a virtual machine running Microsoft Windows XP was used. A number of experiments were preformed. The system behavior was tested depending on:

- presence or absence of ShellBag information about the target folder in the Registry;
- location of the folder with respect to the Desktop;
- type of simulated user actions.

In each experiment a Registry monitoring tool, RegMon (Russinovich and Cogswell, 2006), was used to monitor changes to the Registry.

Experiment 1: Opening a folder that currently has associated ShellBag information and is located on the Desktop.

In the first experiment the simple operation of opening an existing folder was performed. The target folder was selected and opened from the Desktop.

Log analysis: The first step observed in the log was enumerating all existing MRU items within the BagMRU key. The items were checked by the system from the most recently used item to the least recently used item, to find an MRU item's data that matched the target folder's name. The system then updated the BagMRU key's "MRUListEx" value with the new sequence data. So the target folder's MRU item was

marked as the most recently used item within the BagMRU key. In the final step the system found the target folder's Display key and read values from the Shell key under this Display key.

Experiment 2: Closing a folder that currently has associated ShellBag information and is located on the Desktop.

The second experiment involves closing a previously opened folder, which is a direct sub-folder of the Desktop, while monitoring changes to the Registry.

Log analysis: The process began by enumerating the MRU items within the BagMRU key and updating the "MRUListEx" value to ensure the target folder's MRU item is the most recent item in this key. The remaining Registry actions wrote the folder's display settings to the Shell key under target folder's Display Key.

Experiment 3: Opening a folder that currently has associated ShellBag information and is not located on the Desktop.

As mentioned earlier, the Desktop is the hierarchical staring point of the BagMRU key. The third experiment examines the changes to the Registry when the target folder is located in the sub-folder of the Desktop.

Log analysis: By opening an existing folder, which is not directly located on the Desktop, the system updated the BagMRU key first. The position of the MRU item that is associated with the target folder's ancestor folder was updated in the sequence value "MRUListEx" of the BagMRU key. The system then updated this ancestor folder's MRU key, down the hierarchy, until it reached the target folder's parent's MRU key. The whole updating process made each ancestor folder and the target folder become the most recently used item in their parent's MRU key.

Experiment 4: Closing a folder that currently has associated ShellBag information and is not located on the Desktop.

Like the previous experiment, a test for determining user action associated with non-Desktop folders was performed. This time focusing on Registry changes when the folder was closed.

Log Analysis: Here the result shows that when closing a folder, which is not directly under the Desktop folder, the MRU key updating process also started from the BagMRU and ended when it reached this target folder's parent's MRU key. The Shell sub-key under the folder's Display key was then updated.

Experiment 5: Opening a folder that currently does not have associated ShellBag information.

This experiment is designed to observe whether the new MRU key, MRU item and Display key will be created when the user performs an opening operation on folder that does not yet have associated ShellBag information. This experiment was performed twice: one time for the folder that was located directly on the Desktop, and the other time for the folder which was located in a sub-directory of the Desktop folder.

Log analysis: When opening a folder on the Desktop, the Registry operation logs show that the process began by enumerating current items in the BagMRU key. At this point, because the system did not find any existing items matching this folder's information, it did not proceed to the MRU key updating operation, and the whole process terminated. No new ShellBag information was created because of this action; the Registry maintained the same ShellBag information as before the folder was opened.

The same experiment was performed with a folder that is not located on the Desktop. It is observed that the user action did not create any new ShellBag information as the previous experiment, but it updated the target folder's ancestors' MRU items' position. The updating process began from the BagMRU key and ended when it did not find any existing items within the MRU key matching the target folder.

Experiment 6: Closing a folder that currently does not have associated ShellBag information.

The system will not create any new ShellBag information for a new folder when it is first opened. Because of this it is expected that this information is created when the folder is closed. This experiment was conducted to test this theory. It was performed twice: one time for the folder that was located directly on the Desktop, and the other time for the folder which was located in a sub-directory of the Desktop folder.

Log analysis: When closing a folder on the Desktop, the system enumerated the BagMRU key, to find an existing item matching the target folder. After the system did not find a matching item, a request was sent to the Registry to create a new item in the BagMRU key and a new sub-key associated with this folder under the BagMRU key. The newly created MRU item contains the target folder's name and timestamp information. The system also updated the BagMRU key's "MRUListEx" value which moved the newly created item to the most recently used position. The remaining log information shows the process of creating the new Display key for the target folder and writing the display settings values to its Shell sub-key.

When the same experiment is performed to the folder that is not located on the Desktop, the ShellBag information is also created when the folder is closed. In addition, not only this folder's MRU item's position was changed, all its ancestors' MRU items' position was marked as the most recent item in the corresponding "MRUListEx" value.

Experiment 7: Deleting a folder that currently has associated ShellBag information.

The objective of this experiment is to monitor whether the ShellBag information of the target folder will be deleted when the folder is deleted.

Log analysis: The resulting log shows that the target folder and its ancestors' MRU items' position was updated as other actions, but there was no registry deleting operation related to ShellBag information executed in the process of deleting the folder. The deleted folder's ShellBag information remained in the Registry.

Experiment 8: Opening a newly created folder with the same name as a previously deleted folder that has associated ShellBag information.

A new folder is created with the same name and location as a previously deleted folder. The deleted folder has associated ShellBag information stored in the Registry. Because of this, the goal of this experiment is to understand how the ShellBag information will update for this newly created folder.

Log Analysis: The result shows that when this folder was opened it inherited the previously removed folder's ShellBag information. Repeating the tested operations on this new folder will update the already existing ShellBag information, which is now associated with the new folder. Additionally, the binary data stored within the associated MRU item was not

updated. The timestamp information within the MRU item still remains the same.

Experiment 9: Closing a folder that currently does not have associated ShellBag information when the Registry contains the maximum number of Display keys.

Log Analysis: When the folder was closed, the system first created an MRU item and an MRU key associated with the target folder. Unlike experiment 6, the system did not create a new Display key for the target folder but assigned the NodeSlot value under the folder's MRU key to "1", and updated the Shell Key "Bags\1\Shell". When we performed user actions on another folder assigned a NodeSlot value "1", it updated the same "Bags\1\Shell" key. In other words, both of these two folders are associated with the Display key "Bags\1".

3.1. *Other experiments*

ShellBag information updating caused by common user actions such as open and close a folder have been presented above. There are also other actions that will possibly update a folder's ShellBag information. Experiments to monitor some of other user actions have been conducted, and the results are summarized in the next section with above experiments together.

4. Analysis of causality between user actions and ShellBag information updating

The results of conducted experiments are summarized in the form of the following observations. The goal is to consider the question: what traces are left when different ShellBag-related user actions are performed.

Observation 1: According to the experiments, there are three types of updates caused by user actions. User actions therefore can be divided into three groups based on the different updates they cause (Fig. 6).

As shown in Fig. 6, the first type user action will only update the target folder and its ancestor's MRU items' position if these MRU items exist. If some of these relevant MRU items do not exist within the current Registry, system will not create a new item but only update existing MRU items' position. For example in Fig. 7, the "Target" folder does not have any ShellBag information associated with it, so when the user executes the first type action on this folder, the system will only update the "Target" folder's ancestor's MRU item's position within the BagMRU key, and stop when it can not match the "Target" folder with any items in the parent folder "Test" MRU key. The "Test" folder's MRU key still maintains the same MRU sequence as before the action.

The second type of user action will update both the relevant MRU items' position and the contents of the Shell sub-key under the folder's Display key. If the ShellBag information associated with this folder does not exist, system will create it. The last type of action will not influence any ShellBag information within the Registry.

After we generalized the characteristics of different types of user actions, those actions tested in the earlier experiments are summarized in Table 1. Other user actions that are not listed in this table could be tested and analyzed through the same method.

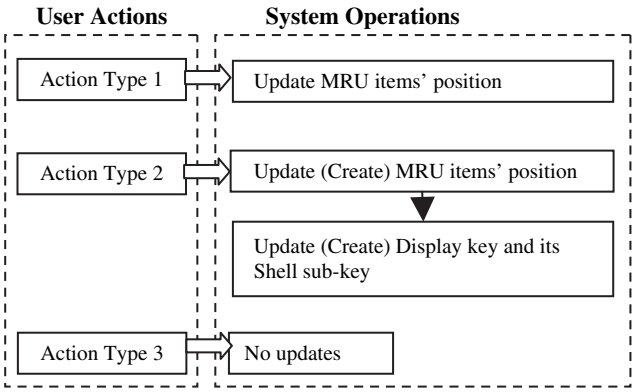

Fig. 6 – The link between user actions and system operations.

From a forensics perspective, the first and second type of user actions may be detected since they will change the ShellBag information within the Registry when performed. However, distinguishing exactly which action changed the ShellBag information will be more difficult due to some of these actions updating ShellBag information in the same manner.

Observation 2: When a user action triggers the system to update the MRU item's position of the target folder in the Registry key, the updating process begins by adjusting the position of the MRU items that associated with the folder's ancestors. The modification process updates the ancestor folder's position within the BagMRU key until reaching the target folder's MRU item. Fig. 8 illustrates this updating mechanism. Assuming that the current MRU item sequence depicted in the BagMRU key is "My Computer", "Test" and "My Documents", where "My Computer" was the most recently used item. The "Test" folder has two sub-folders where the "Target" folder is the least recently used folder. If the open operation is performed on the "Target" folder, the system first updates the BagMRU key by marking the "Target" folder's ancestor, named "Test", as the most recently used folder. The system then updates the "Test" folder's MRU key

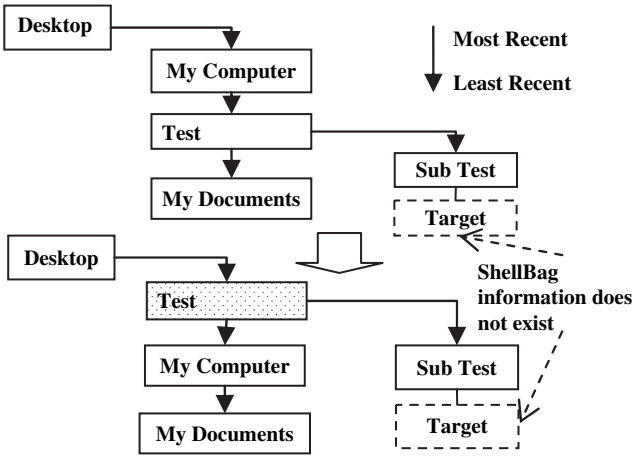

Fig. 7 – Updating example 1.

Table 1 – Summary of user actions.

User action	Action type		
	Action Type 1	Action Type 2	Action Type 3
Open a folder	√		
Close a folder		√	
Create a folder			√
Delete a folder	√		
Copy a folder	√		
View target folder as thumbnail within parent folder	√		
Create a new file			√
Delete a file			√

by marking the "Target" folder's item as the most recently used, after that the whole process completes. At this point the 'Target" folder and all its ancestor folders have became the most recently used items within their MRU keys. So to summarize this observation, if the position of the MRU item associated with a target folder is updated, all the target folder's ancestor folders' MRU items' position must have been updated prior to the target folder's MRU item.

Observation 3: In the above discussion the updating operation refers to the Registry operation "SetValue" observed in the RegMon log. This registry operation sets a value to a specific Registry key, but it does not necessarily change the target key's LastWrite timestamp. It will only change when the key is updated with a value that is new or different from the current value. Therefore, although as described earlier the system will update the folder's relevant MRU items' position within its parent key and the Shell sub-key under the folder's Display key when certain user actions are performed, these keys' LastWrite timestamps are only changed when its contents have changed.

Observation 4: In the process of matching an existing MRU item with the target folder, the only decisive factor is folder's name. When a new folder is created with the same name as the folder which once existed under the same directory, the new folder will inherit the previous folder's MRU key, MRU item and Display key if they had already been created.

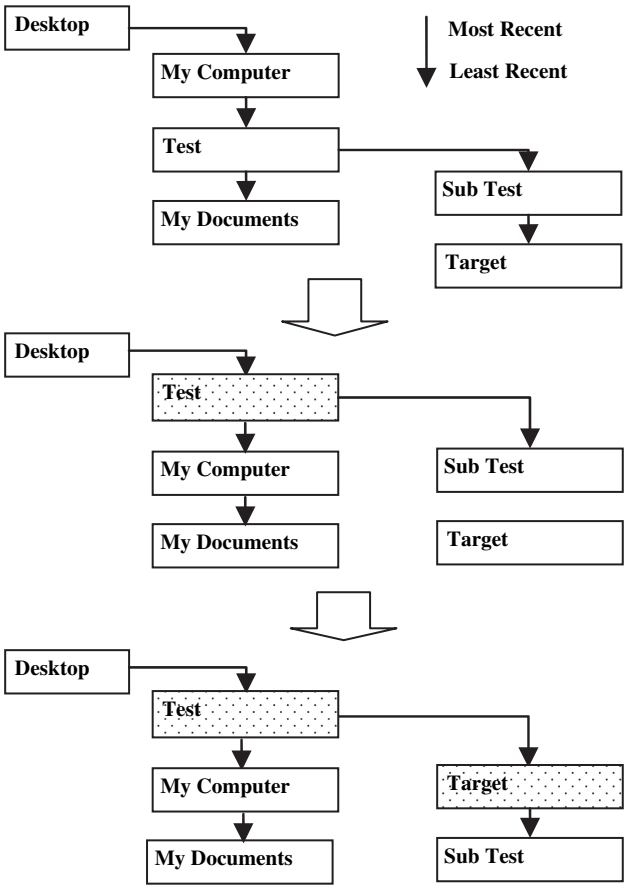

Fig. 8 – Updating example 2.

5. ShellBag information analysis method

The ShellBag information analysis method is extended from the Registry snapshots comparison method described in Zhu et al. (2009b). The Registry snapshots are, by default, created within System Restore Points to back up the Windows Registry every 24 calendar hours and possibly more frequently when certain events occur such as the installation of new software (Harder, 2001). So if the current Windows Registry can be considered as the most recent snapshot of itself with all the Windows Restore Points containing earlier snapshots. Two consecutive Registry snapshots can be compared value by value to identify values that have changed between two snapshots. Each snapshot has a creation timestamp associated with it. If a data value has changed in a registry key, then it must have happened between the creation time of the preceding snapshot and the LastWrite timestamp of this registry key. The user action that caused this data value to change must have also occurred within the same time period (see Fig. 9).

This approach can be adopted for the ShellBag information analysis. Given below are nine detection rules for detecting the presence or absence of user actions between two snapshots. Most of these rules can be derived by contraposition with the observations described in the previous section.

Detection Rule 1: If an MRU item's position is updated, it must be either the first type action or the second type action was performed on this MRU item's associated folder or its subfolders. This rule is derived from the Observation 1.

Detection Rule 2: Due to the fact that only the second type user action is observed to create and update display settings (Observation 1), if a Shell key under a Display key was created or updated, the second type action (e.g. Close a folder) must have definitely been performed on this Display key's associated folder (when the number of Display keys does not exceed the defined maximum number) or one of these associated folders (when the number of Display keys is equal to maximum number) at the time of the Shell key's LastWrite timestamp. This rule may need to be refined if other type actions are observed have the same characteristic in further research. Additionally, the second type action could also occur without changing the Display key. The unchanged state of the Display key cannot prove these actions did not occur.

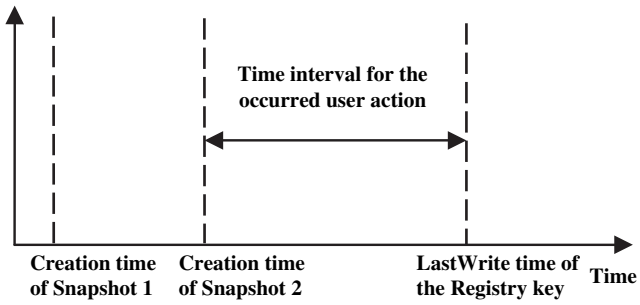

Fig. 9 – Time interval for occurred user action.

Detection Rule 3: If a folder is not associated with an MRU key, MRU item and Display key, the second type action was never occurred on this folder in the past (Observation 1).

Detection Rule 4: If an MRU item's position has been updated then its associated folder's ancestors' MRU items' position must have also been updated prior to this MRU item. This rule is developed from observing that a folder's MRU item is always updated after its parent folder's MRU item (Observation 2). So the **Detection Rule 5**, if an MRU item's position has not been updated then its associated folder's sub-folders' MRU items' position has definitely not been updated during the same period, also holds.

Detection Rule 6: An MRU key's LastWrite timestamps are the same within two consecutive Registry snapshots if and only if: no items position have been updated or only the first (the most recent) item's position was updated between the earlier Registry snapshot's creation time and the later Registry snapshot's creation time. This rule could be proved by contradiction where it is assumed that the item, which is not the most recent, was updated its position and the MRU key's LastWrite timestamp had not changed. This is a contradiction to Observation 3, so the rule is proved. This rule could also be presented as the **Detection Rule 7** that the LastWrite timestamp has not changed if and only if the items, except the most recently used item, did not update between the earlier Registry snapshot's creation time and the later Registry snapshot's creation time.

Detection Rule 8: This rule considers the process of updating an MRU key. An MRU key's timestamp was changed but all of this key's data values are identical, if and only if at least the first (the most recent) item's position and the second (the second most recent) item's position was definitely updated during the period between the previous Registry snapshot's creation time and this key's LastWrite time. Other items could also be possibly updated, which can not be determined by the comparison. This rule is derived from the Observation 3. If only the most recent item was updated, or even no item was updated, the MRU key's LastWrite timestamp would not change, therefore the two most recent items were definitely updated in order to maintain the identical sequence and change the key's LastWrite timestamp.

Detection Rule 9: The last rule is derived from Observation 4. A folder can inherit the ShellBag information of a previously deleted folder if the new folder has the same name and path as the previously deleted folder. Because of that, the detected events, which relate to particular ShellBag information, cannot be directly attributed to the associated folder that currently exists in the file system. These events could have been applied to the folder that previously existed at this path and was deleted. So it is necessary to compare the folder creation time stored in the MRU item's binary data with the creation time of the folder currently located at this path. If the two times are matching, then the events detected from ShellBag information can be attributed to the folder that is currently present in the file system.

6. Case study

In order to demonstrate the practical value of these proposed detection rules, a case study is shown. A disk image has been given to an investigator for analysis. Some remnants of child pornography (CP) have been found. The suspect is claiming to have no knowledge of these items, and claims it must have been a virus that downloaded the pornographic material. A copy of the image was mounted, and a virus scan was run. Several Trojan type viruses were found. The onus is now on the investigator to prove that the suspect did, in fact, access the illegal content.

The first step of the investigation was to extract all the Registry snapshots from the system. 16 Registry snapshots were found in the Windows Restore Points, as well as the most recent Registry hive from the Windows system folder.

After finding these Registry snapshots, the proposed ShellBag analysis method was applied. Each snapshot was compared with the preceding snapshot to reconstruct user activities. For example, the Display key ''Bags\8'' was found associated with the directory ''C:\My Documents\My Pictures'' (by using the method described in Section 2). This Display key firstly appeared in the Registry Snapshot 9 (Fig. 10). According to Detection Rule 2, a folder at path ''C:\My Documents\My Pictures'' used the first time by the second type user action at 22/04/2008 23:24:42 UTC. After comparing the creation time stored in the associated MRU item's binary data with the creation time of the folder currently located at this path, it indicates that the folder that is currently presented in the file system is associated with this particular event (Detection Rule 9).

Other ShellBag information was examined in the same manner and the results of this examination are presented below.

Under the directory ''C:\My Documents\My Pictures'', a sub-folder called ''Great Stuff'' was found to be first accessed at 01/05/2008 19:48:54 UTC and its sub-folder ''better'' was also accessed at 05:48:26 UTC next day. The detection rules clearly show that these folders were accessed several times since they were first used. Some other folders under the directory ''My Documents\fpics'', called ''toons'', ''candispic'', ''Hentai Anime Monster Forced Hardcore.avi'', and ''Sandra#312'' which contain illegal material, all first accessed between 23/04/2008 18:02:46 and 22:56:10 UTC. Meanwhile, some folders with the same name as above were also first browsed within the ''My Downloads'' folder. Because of this, it is likely that the contents of these folders were downloaded from the Internet.

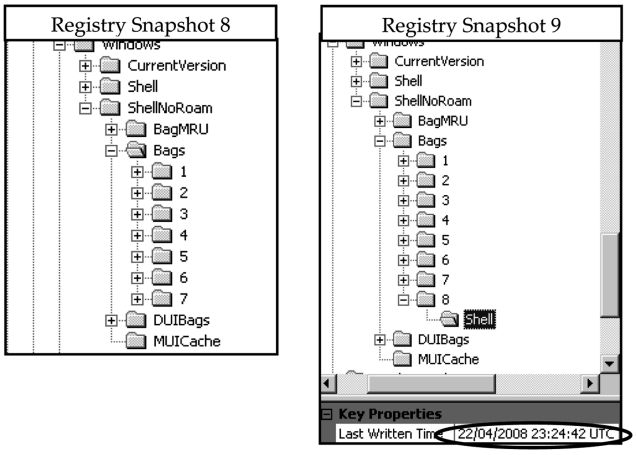

Fig. 10 – Screenshot of ShellBag information within different Registry snapshots via Registry Viewer (AccessData, 2008).

Apart from the direct evidence, the ShellBag information shows that the file "PGPDesktop982_Win32.zip" was first used at 23/04/2008 07:19:03 UTC. An application directory "C:\Program Files\PGP Corporation\ PGP Desktop\" was accessed 10 min after. This indicates that the PGP encryption tool was probably used to hide traces from investigators. The folder for tool "XnView-beta" was also first time detected on the same day at 18:44:08 UTC.

The other interesting information derived from the ShellBag detection is of a folder at "G:\DCIM\100KC763" which was first accessed at 17:23:28 23/04/2008. The path "G:" suggests that this folder is probably an external hard drive. USB device Registry entries were examined, and show that a "WD 1600BEV External USB Device" was connected to the computer three hours earlier. So this external device may be relevant to this case, and deserves further examination.

In summary, there are two benefits of using the ShellBag information analysis method. First of all, it will significantly increase the efficiency of an investigation. There are approximately 2500 folders in this case. Only 72 have associated ShellBag information. According to the detection rule 2, the user must have accessed these 72 folders at least once. The investigator could begin the investigation by focusing on these folders, which greatly narrow the scope in which user-created traces may exist. Second, as shown in Fig. 11, the detection methods provide more comprehensive information of user activities to the investigator. With this information, the investigator could understand what directory has been accessed at what time. It would also allow the investigator to determine whether a suspect had knowledge of the content found within a certain directory.

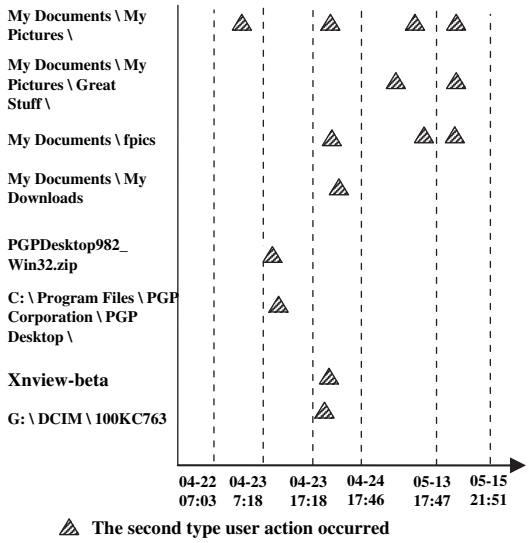

Fig. 11 – Timeline of suspect user actions (Partial).

7. Conclusion

This paper introduced a novel method to examine ShellBag information within Registry snapshots. It compared different states of ShellBag information within consecutive Registry snapshots in order to examine ShellBag-related user actions. Nine detection rules were proposed on the basis of analyzing the causality between user actions and ShellBag information update patterns. These detection rules can also be used in cooperation with other Registry snapshots analysis method, such as the MRU key comparison algorithm proposed in Zhu et al. (2009a), to provide increasingly comprehensive information. Finally, a case study was given to illustrate the practical value of this proposed ShellBag analysis method.

The method developed in this paper is targeted at Microsoft Windows XP. A forensic tool, named TraceHunter, was developed based on this proposed method. Other versions of Windows Operating Systems will be studied via the same method and more action detection methods will be developed in the course of further research.

REFERENCES

AccessData. Registry viewer, http://www.accessdata.com/products.html; 2008.

Carvey H. Registry analysis, windows forensic analysis, syngress. Rockland, Massachusetts, U.S.A.; 2007.

Harder B. Microsoft Windows XP system restore, Windows XP technical articles, http://technet.microsoft.com/en-us/library/ms 997627.aspx; 2001.

Russinovich M, Cogswell B. RegMon, http://technet.microsoft.com/en-us/sysinternals/bb896652.aspx; 2006.

Zhu Y, Gladyshev P, James J. Identifying newly updated data values of MRU Keys between registry snapshots. In: Fifth annual IFIP WG 11.9 international conference on digital forensics; 2009a.

Zhu Y, James J, Gladyshev P. A comparative methodology for the reconstruction of digital events using Windows Restore Points. Digital Investigation 2009b; doi:10.1016/j.diin.2009.02.004.

Mr. Yuandong Zhu is a PhD student at the School of Computer Science and Informatics at University College Dublin, Ireland. His research interests include user activity analysis and forensic tool development. His current work is focusing on the analysis of different states of Window Registry snapshots within Windows Restore Points. He has developed a forensic tool (TraceHunter) to extract comprehensive user activity information from a computer running Microsoft Windows by utilizing novel Registry snapshot comparison techniques.

Dr. Pavel Gladyshev is a lecturer at the University College Dublin's School of Computer Science and Informatics, where he is directing the GDip/MSc programme in Forensic Computing and Cybercrime Investigation – an international distance learning programme for the law enforcement officers specializing in cybercrime investigations. Dr. Gladyshev's research interests are in the area of Information Security and Digital Forensics. His current work is focusing on the logical foundation of digital forensic analysis and its applications to investigations of cybercrimes. Dr. Gladyshev serves on the editorial boards of the International Journal of Digital Evidence, the International Journal of Digital Crime and Forensics, and on the Board of Referees of the Digital Investigation Journal. Dr. Gladyshev also serves as an invited expert to the Interpol working party on IT Crime (Europe).

Mr. Joshua James is a research student in the Centre for Cybercrime Investigation at University College Dublin, Ireland. Coming from a background in Network Security and Administration, his focus is now on automatic digital evidence identification and correlation. He is specifically working in the area of rigorous automated investigation and analysis techniques for digital investigations with emphasis of ease of use. To this end, he has started the open source project titled Rapid Evidence Acquisition Project for Event Reconstruction (http://CybercrimeTech.com).