THE JOURNAL OF
DIGITAL FORENSICS,
SECURITY AND LAW

Journal of Digital Forensics, Security and Law

Volume 3 | Number 4 Article 2

2008

Extraction and Categorisation of User Activity from Windows Restore Points

Damir Kahvedžić
University College Dublin

Tahar Kechadi
University College Dublin

Follow this and additional works at: https://commons.erau.edu/jdfsl

Part of the Computer Engineering Commons, Computer Law Commons, Electrical and Computer Engineering Commons, Forensic Science and Technology Commons, and the Information Security Commons

Recommended Citation

Kahvedžić, Damir and Kechadi, Tahar (2008) "Extraction and Categorisation of User Activity from Windows Restore Points," *Journal of Digital Forensics, Security and Law*: Vol. 3 : No. 4 , Article 2.
DOI: https://doi.org/10.15394/jdfsl.2008.1049
Available at: https://commons.erau.edu/jdfsl/vol3/iss4/2

This Article is brought to you for free and open access by the Journals at Scholarly Commons. It has been accepted for inclusion in Journal of Digital Forensics, Security and Law by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.

(c)ADFSL

Extraction and Categorisation of User Activity from Windows Restore Points

Damir Kahvedžić
damir.kahvedzic@ucd.ie
Computer Science and Informatics
University College Dublin,
Ireland

Dr.Tahar Kechadi
tahar.kechadi@ucd.ie
Computer Science and Informatics
University College Dublin,
Ireland

ABSTRACT

The extraction of the user activity is one of the main goals in the analysis of digital evidence. In this paper we present a methodology for extracting this activity by comparing multiple Restore Points found in the Windows XP operating system. The registry copies represent a snapshot of the state of the system at a certain point in time. Differences between them can reveal user activity from one instant to another. The algorithms for comparing the hives and interpreting the results are of high complexity. We develop an approach that takes into account the nature of the investigation and the characteristics of the hives to reduce the complexity of the comparison and result interpretation processes. The approach concentrates on hives that present higher activity and highlights only those differences that are relevant to the investigation. The approach is implemented as a software tool that is able to compare any set of offline hives and categorise the results according to the user needs. The categorisation of the results, in terms of activity will help the investigator in interpreting the results. In this paper we present a general concept of result categorisation to prove its efficiency on Windows XP, but these can be adapted to any Windows versions including the latest versions.

Keywords: Windows Registry, Registry Restore-Points, User Activity, Forensic Registry

1. INTRODUCTION

System Restore is a process that monitors key system changes on a computer running Microsoft Windows. Whenever a change that could endanger the system's stability is detected, System Restore copies the core system files and stores them in a hidden directory ("C:\System Volume

Information_restore{GUID}”) in the files system before allowing the change to take place. If the subsequent change results in an unstable system the user can simply reload the last known good configuration and undo the damaging changes. Typically a large number of these time points, called Restore Points, are found in the system. They present a snapshot of the state of the system at that point in time. Differences between them can highlight significant user activity that could be useful to a criminal investigation.

A number of different files are monitored by System Restore and are archived in the restore point (Microsoft (2007)). The user can specify which file types they want to monitor by modifying a particular system parameter. However, left as default, the system restore archives the Registry, COM+ database, the IIS metabase and other specific file extensions. A log of the changes is also stored.

The most important of these are the registry hive files. The registry is a central hierarchical database used in Microsoft Windows operating systems to store information that is necessary to configure the system for the users, applications, and hardware devices. It provides a single location where installed programs, user profiles and settings can be stored and managed. Analysing different values in the registry not only reveals currently installed programs and the state of the operating system but would also give clues to recent opened files, folders, network connections, and other user activities.

This paper presents a new methodology for extracting user activity from Windows Restore points. We design and implement a forensic registry analysis tool that is able to compare different Restore Points and hives. The results are presented to the investigators to help them in determining how the system was used between the snapshots. The approach consists of, firstly, comparing Restore Points selectively and, secondly, categorising the results in terms of evidence. Even thought we focus on extracting the differences between the registry hives of two or more Restore Points, the same technique can be applied to other sets of files found within those points.

Restore Point Creation

The number of registry Restore Points stored in the file system varies from computer to computer. The factors that influence their creation include how often new drivers and large programs are installed, if the computer is powered on constantly and whether the user has turned off restore point creation process. Regardless of these factors, Restore Points are likely to be found in the majority of the file systems. Once created, they store an exact copy of the active hives of the system registry. Honeycutt, J. (2002) described when Restore Points are created.

- On Schedule: The default is 24 hours.
- On Program Installation: The system may be backed up when a user

installs a software that uses a particular type of installer.

- On Update: The system is backed up just before an update to the operating system takes place.
- On System Restore: The system is backed up before a system is restored using one of the Restore Points.
- On Driver Installation: Device drivers affect system stability so the system is backed up for security.
- On User Request: Users can create manual restore points.

The Restore Points are kept on disk for up to 90 days and are deleted after this time. As a result, it is not uncommon to find many different copies of the state of the system in a typical forensic investigation. Although these Restore Points are extremely useful to roll back unwanted system wide changes, they are also an invaluable forensic resource to provide insight about the state of the system at a given time.

Test Setup

Throughout the paper we will use a test file system to illustrate the investigative techniques on a practical example. The system is a typical independent computer using the Windows XP operating system. The computer configuration and other relevant data are shown below. All settings dictating the frequency of the restore point creation were left in their default values. Nevertheless, the creation of the Restore Points was not done at regular intervals. The differences between the Restore Point creations varied from a single day to 10 days. In this case study the Restore Points found in the system had a total time range of 2½ months. In the rest of the paper, the Restore Points will be referred to with respect to its creation name (i.e., RP11) and the number of days after the oldest hive was created (i.e. RP11 (25 days)). In this way a perspective is maintained on the time range between two Restore Points.

Computer Manufacturer	Dell
Operating System	Windows XP SP2
Number of Restore Points	17
Time Range of Restore Points	2 months
Frequency of Use	Light to Medium Use

Table 1: Test system configuration

The above computer system is used to demonstrate how user activity can be recreated by comparing the Restore Points. Initially we focus on the registry hives within these points. The validity of the activity found was confirmed by interviewing the owner of the system. The demonstrations use the “SOFTWARE” and “ntuser.dat” hives respectively, since they hold most of the interesting evidence. Other hives can be used in a similar manner.

RPCompare

RPCompare, (Restore Point Compare), is a tool for extracting and studying the user activity through Restore Points. It is designed to compare the Restore Points in an offline forensic environment and present the differences to the investigator. It does not use the WMI interface or any inbuilt Windows functions. The tool is primarily designed to extract and compare the various registry hive files and highlight any of the keys and values that have been deleted, added or modified in the interim. The tool contains two phases:

- Comparison phase: The registry entries are compared to identify which keys have changed between time points.
- Interpretation phase: The differences are analysed to interpret their meaning with respect to user activity.

Registry Comparison Phase

RPCompare extracts the registry entries from any number of Restore Points and compares either their keys or their values. The keys that present differences are extracted and tagged with “Added”, “Modified” or “Removed” with respect to the more recent registry. The comparison is done as follows.

RPCompare uses the naming conventions of the Restore Points to order the points. Each Restore Point is named RP*xx* with *xx* being the rank of the restore point (Bunting, S. (2008)). RPCompare also utilizes the “Last Written Time” values present in all of the registry keys including the root key. The value is updated by Windows whenever an operation “write” or “modify” is carried out on the data of the key. The time written to the value depends on the system clock, which can be manipulated by the user. For the purposes of this study, we assume that the time is an accurate reflection on the real time, and that the time is consistent throughout the Restore Points.

RPCompare recursively traverses the length of each hive tree comparing every node's time values. All the relations are with respect to the earlier node. If a node has a different time value to its corresponding node in the next hive then that node is tagged as “Modified”. If it does not exist in the next hive then it has been “Removed”. Finally, if a new node has been found in the new hive then it has been “Added”. Values are compared only for those keys that have been tagged as modified. Once the set of differences has been found, it is presented to the user for interpretation.

Interpretation Phase

The hive keys have a wide number of purposes and they may or may not change as a direct result of user actions. Some changes are due to automatic system processes such as program updates or system checks while others are directly attributed to the user conducting a specific activity. The roles of a large number of registry keys have been documented in both forensic (ForensicMatter (2008)) and non forensic fields and can easily be referenced if they are found to have been changed between time points. A forensic investigation is case specific and, therefore, not all information presented by RPCompare may be useful. It is up to the investigator to filter out irrelevant changes and extract only the information that is relevant to the case.

Performance

RPCompare has a number of important performance issues, which may adversely affect the efficiency of an investigation. Since RPCompare compares every key with its corresponding key in another hive, the complexity of its execution is *O(n*m)*, where n and m are the number of keys in the hive. In the worst-case scenario, if the key is at the root of the hive, the whole hive will be compared. In our tests, comparison of a mature hive, such as a SOFTWARE hive, is very time consuming; as such we present a number of techniques to concentrate on specific branches of the hive or to limit the time range of the comparisons.

Moreover, the interpretation of the results is currently done by inspecting each set of differences by hand. In the case of a complete comparison across all the Restore Points, the time needed to perform the interpretation will be very long and in some situations make this tool impractical. Therefore, we come up with two main improvements for both phases to make RPCompare more efficient.

Firstly, we developed a progressively detailed a view of the data to guide the investigator in concentrating on the hives with high user activity. We classify the techniques into two categories; those that attempt to find large-scale differences in the system such as installations/uninstallations of programs and those techniques that attempt to recreate the minute steps of the user. The latter involves the processing of Most Recently Used (MRU) lists and other private attribute information that can highlight how the user used the system. A special processing needs to be carried out on these types of registry entries to understand their meaning and their relationships with the user. The Large Scale Comparison and the User Activity Sections detail this methodology.

Secondly, we developed and implemented a technique that attempts to discard differences that are irrelevant to the investigation at hand. Specific evidence types can be analysed if they are deemed to be relevant to the investigation. Other non-relevant types may be ignored. Therefore, we employ rules to classify the differences into a set of evidence types. The Categorisation Section details a sample of the rules employed to categorise the differences in these evidence types.

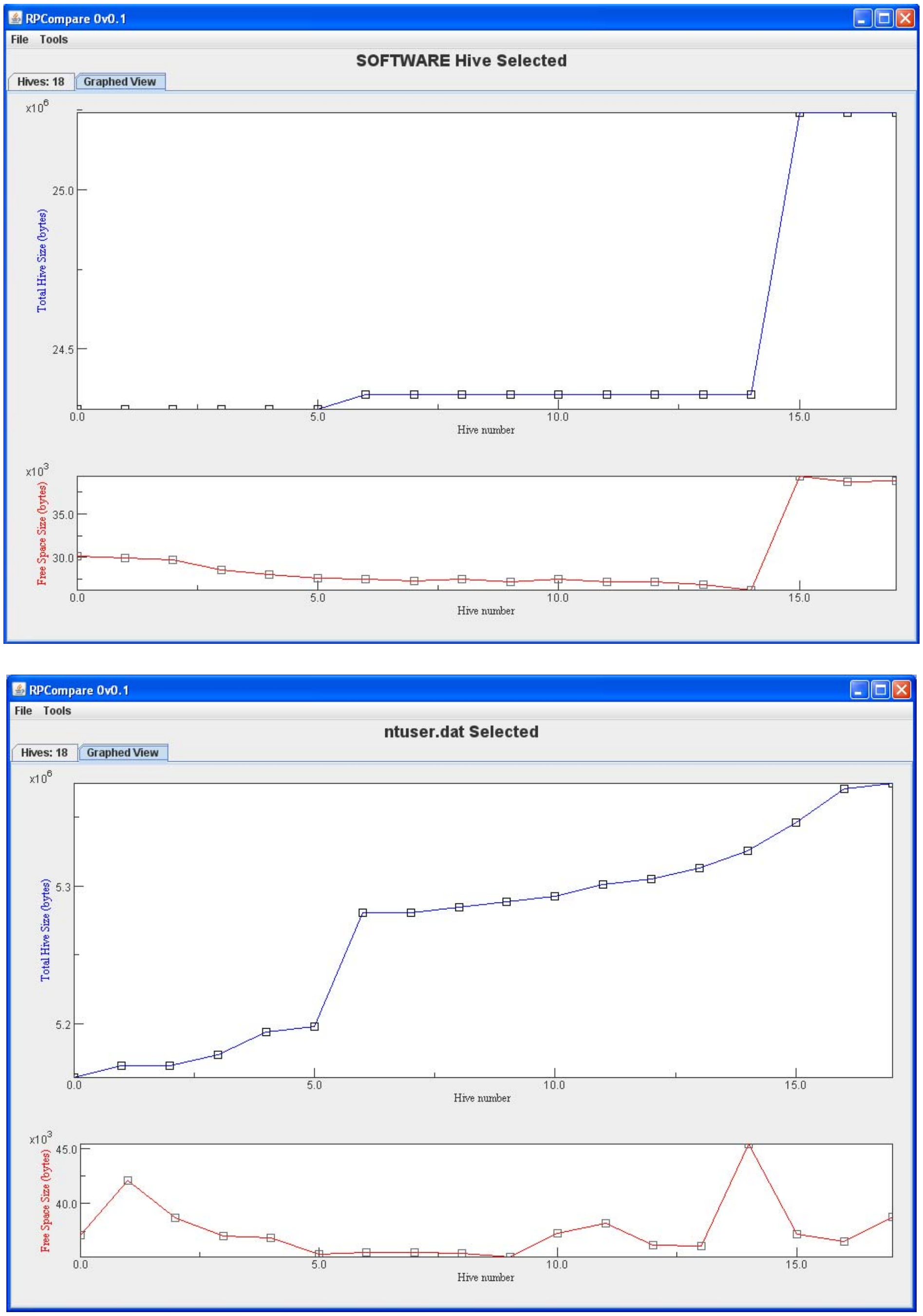


Figure 1: Sizes of the SOFTWARE and ntuser.dat Hives

Blue line: Total Hive Space, Red Line: Total Unallocated Hive Space

2. LARGE SCALE HIVE COMPARISON

The investigator may need to expose large spots of activity and illustrate what was the general use of the system over a long period of time. User activity such as installation and uninstallation of programs and the addition or deletion of user accounts can be detected by comparing the registries of the system taken before and after the activity. The following technique illustrates a method on how the registries are compared in a progressively detailed manner. The process involves highlighting the areas of large activity by comparing file sizes first, selectively comparing the entire hives, then branches and finally values. The progressively detailed comparisons allow the investigators to streamline the comparison process and avoid spending too much time on complete hive comparisons.

Registry Size Comparisons

The first procedure entails comparing the sizes of the hives against each other. The differences can highlight some important changes that have occurred between time points in a relatively quick manner. Although small changes may not be noticeable, a large change to the hive, such as a program installation, can be easily spotted by comparing the hive sizes. The results can highlight a time point of high activity and bring it to the attention of the user for further investigation.

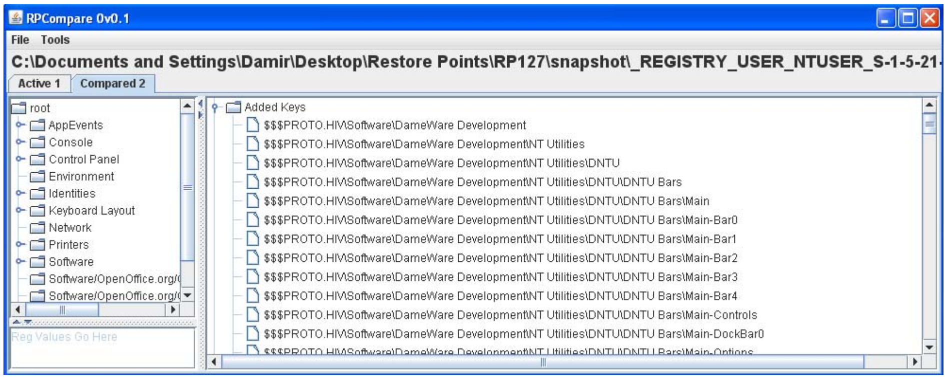
Variations in sizes highlight additions to the registry but it cannot show activity that *removes* keys from the hives. The extraction of unallocated space in the hive can be used to accomplish this aim. Whenever a key or a set of keys is deleted, due to an uninstallation or registry cleaning for example, the space left by the removed keys is marked as empty and is kept for future use (Russinovich (2008)). The hives never shrink to compress this space and therefore do not reveal the uninstallation in its file size. In order to highlight this fact, RPCompare calculates the amount of free space in the registry alongside the total amount of used space. Sharp increases in the total amount of unallocated space imply large-scale removal of keys.

Figure 1 shows the sizes of the hives in the case study. The size of the SOFTWARE hive increased at the latter part of the graph; between RP14 and RP15 which relate to 37th day and 49th day after the oldest restore point. Similarly, in the “ntuser.dat” file, the total space rises sharply between RP5 and RP6 or the 9th and 10th day after the oldest hive. Deallocated space has largely remained constant except between RP0 to RP1 and RP13 to RP14. The investigator can therefore narrow the range of the comparison for closer analysis.

RPCompare was executed on the SOFTWARE and “ntuser.dat” hives to recover the keys that were added at the above times. In the case of the

SOFTWARE hive, RPCompare found that keys relating to the installation of the .NetFramework were responsible for the size increase in the 10 day time range between RP14 and RP15.

In the case of the "ntuser.dat" hive, the size of the hive began to increase at RP3 (7 days after the first restore point) with a huge size increase at RP6 and a steady increase thereafter. Comparison of RP5 and RP6 resulted in the identification of 143 Added, 111 Modified, and 4 Removed keys. This result is shown in the Registry Compare window, as shown in Figure 2. The majority of added keys are related to a DameWare (DameWare 2008) program. Upon further investigation, this program was found to be a PC remote control utility. Although in this case the installation was for innocent use, if the investigator is looking for a particular type of criminal activity this can be seen as vital evidence. The progressive increase in hive size from RP12 was attributed to new keys being added in the "ShellNoRoam" branch of the hive. These keys store window positioning preferences for each folder in the file system and are discussed further on in this paper. New "ShellNoRoam" keys indicate creation of new folders in the file system.



```
C:\Documents and Settings\Damir\Desktop\Restore Points\RP127\snapshot\_REGISTRY_USER_NTUSER_S-1-5-21-
```

```
$$$PROTO.HIV\Software\DameWare Development
$$$PROTO.HIV\Software\DameWare Development\NT Utilities
$$$PROTO.HIV\Software\DameWare Development\NT Utilities\DNTU
$$$PROTO.HIV\Software\DameWare Development\NT Utilities\DNTU\DNTU Bars
$$$PROTO.HIV\Software\DameWare Development\NT Utilities\DNTU\DNTU Bars\Main
$$$PROTO.HIV\Software\DameWare Development\NT Utilities\DNTU\DNTU Bars\Main-Bar0
$$$PROTO.HIV\Software\DameWare Development\NT Utilities\DNTU\DNTU Bars\Main-Bar1
$$$PROTO.HIV\Software\DameWare Development\NT Utilities\DNTU\DNTU Bars\Main-Bar2
$$$PROTO.HIV\Software\DameWare Development\NT Utilities\DNTU\DNTU Bars\Main-Bar3
$$$PROTO.HIV\Software\DameWare Development\NT Utilities\DNTU\DNTU Bars\Main-Bar4
$$$PROTO.HIV\Software\DameWare Development\NT Utilities\DNTU\DNTU Bars\Main-Controls
$$$PROTO.HIV\Software\DameWare Development\NT Utilities\DNTU\DNTU Bars\Main-DockBar0
$$$PROTO.HIV\Software\DameWare Development\NT Utilities\DNTU\DNTU Bars\Main-Options
```

Figure 2: 143 Added, 111 Modified, 4 Removed Keys between Time Points 5 and 6

Registry Branch Comparisons

Once a branch of a hive has been found and suspected to contain evidence, RPCompare can take the root of that branch and compare it with similar branches in other hives. Because comparing a single branch is much faster than comparing the whole tree hierarchy, the investigator can concentrate on particular aspects of the hive at any given time relatively efficiently.

Returning to the DameWare example above, the DameWare Development key was compared to all the hives that were created after its installation. None of the hives reported any differences. This suggests that the program was rarely used. Upon further investigation it was found that the program was installed as a trial and not actively utilized. Progressively detailed key or branch comparisons can also be carried out if the investigator deemed it to be necessary.

3. USER ACTIVITY EXTRACTION

The registry contains many important locations that can be directly associated to the user and the way that the system was used at the time of the registry snapshot. The "Most Recently Used" (MRU) lists store evidence of files names, programs and other information that has been opened by the user in the recent past. They have been particularly highlighted as highly valuable pieces of user activity (Honeycutt 2002), ForensicMatter.com (2008)). These locations are widely known and are actively analysed in most investigations. The investigator may look manually at the MRUs in the Restore Points but this can be extremely laborious. RPCompare can extract an MRU key, compare it across different Restore Points and extract the user activity that the MRUs held. This section elaborates on the MRUs and how they can be processed to gain understanding of user activity.

Registry MRU Management

The MRU key is a standard in Windows that store the most recently used items in the system. Each MRU 'listens' for a particular user activity and updates its content if this activity occurs. They store two types of values; a value for each of the entries and an index value, the MRU value, which stores a list of the entries in order of most recent.

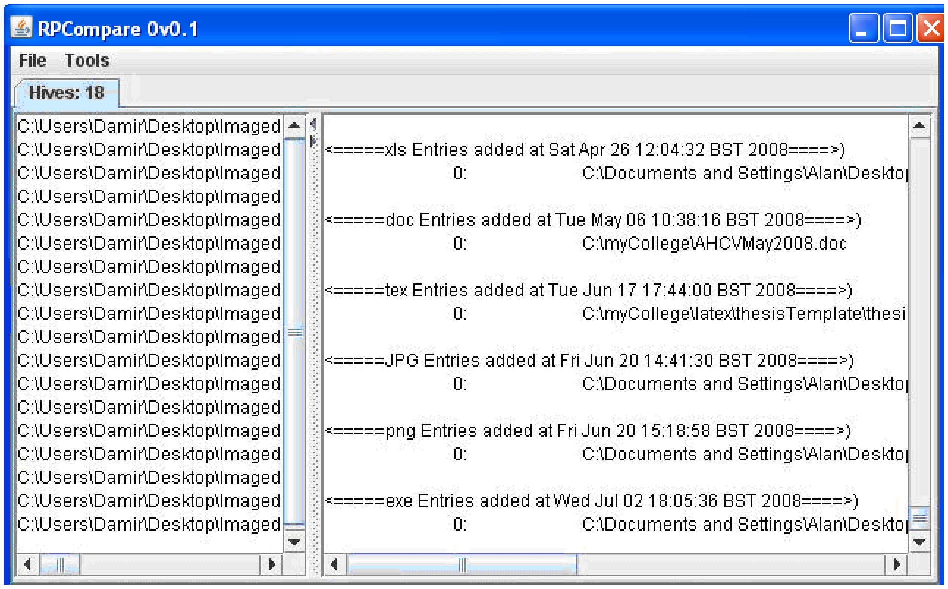
For example, the "HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU" key stores the most recent files opened in the Windows open dialogue. The subkeys of the key store entries for specific file extensions. Any new open file is captured as a value in the extension's MRU with an unused index as its name and the filename as its data. The index is placed at the head of the MRU list signifying that it is the most recent. Only a limited number of indexes exist, so if there is no free one, the oldest entry in the list is removed and its index is given to the new entry. If a command has been executed and is already present in the MRU key, then its index is simply upgraded in the MRU list. No new values are created.

RPCompare and the MRU Timeline

In order to compare the MRU keys correctly, RPCompare contains an algorithm to combine the MRU lists and disregards any reoccurring differences. Therefore, if only one new entry is found, only the new command will be highlighted in the report. In this way the analyst can get a clear history of the list without being confused by the other repetitive values.

RPCompare was executed on the ‘OpenSaveMRU’ key to extract the user activity held in this MRU. A timeline of the different MRU’s timestamps can be created to illustrate this more easily. Figures 3 and 4 show the different perspectives. Figures 3 show a textual representation (obscured for privacy) while Figure 4 shows a timeline where peaks indicate higher amounts of new MRU entries and therefore more user activity.

Very few new MRU entries are created even for an extended time range. This low user activity indicates that the computer system was used very lightly. This corroborates the stated system specifications in the Test Setup Section. Although only ‘OpenSaveMRU’ was analysed, RPCompare can aggregate other MRUs from other locations in the registry in a similar manner. The more timestamps that are collected the more accurate the MRU time line becomes.



```
C:\Users\Damir\Desktop\Imaged
C:\Users\Damir\Desktop\Imaged
C:\Users\Damir\Desktop\Imaged
C:\Users\Damir\Desktop\Imaged
C:\Users\Damir\Desktop\Imaged
C:\Users\Damir\Desktop\Imaged
C:\Users\Damir\Desktop\Imaged
C:\Users\Damir\Desktop\Imaged
C:\Users\Damir\Desktop\Imaged
C:\Users\Damir\Desktop\Imaged
C:\Users\Damir\Desktop\Imaged
C:\Users\Damir\Desktop\Imaged
C:\Users\Damir\Desktop\Imaged
C:\Users\Damir\Desktop\Imaged
C:\Users\Damir\Desktop\Imaged
C:\Users\Damir\Desktop\Imaged
C:\Users\Damir\Desktop\Imaged
C:\Users\Damir\Desktop\Imaged
```

```
<=====xls Entries added at Sat Apr 26 12:04:32 BST 2008====>)
        0:          C:\Documents and Settings\Alan\Deskto

<=====doc Entries added at Tue May 06 10:38:16 BST 2008====>)
        0:          C:\myCollege\AHCVMay2008.doc

<=====tex Entries added at Tue Jun 17 17:44:00 BST 2008====>)
        0:          C:\myCollege\latex\thesisTemplate\thesi

<=====JPG Entries added at Fri Jun 20 14:41:30 BST 2008====>)
        0:          C:\Documents and Settings\Alan\Deskto

<=====png Entries added at Fri Jun 20 15:18:58 BST 2008====>)
        0:          C:\Documents and Settings\Alan\Deskto

<=====exe Entries added at Wed Jul 02 18:05:36 BST 2008====>)
        0:          C:\Documents and Settings\Alan\Deskto
```

Figure 3: OpenSaveMRU textual Time line

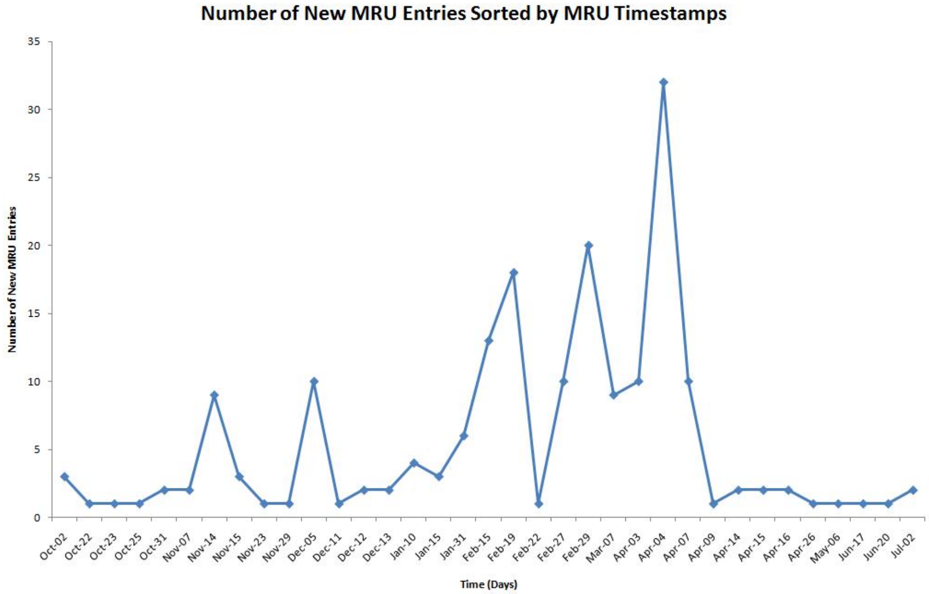
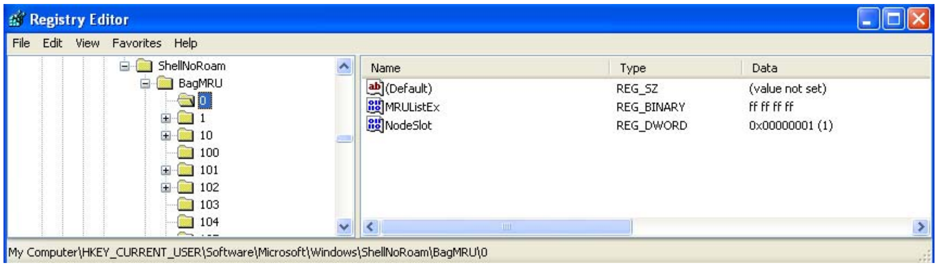


Figure 4: OpenSaveMRU Graphed Timeline



Name	Type	Data
(Default)	REG_SZ	(value not set)
MRUListEx	REG_BINARY	ff ff ff ff
NodeSlot	REG_DWORD	0x00000001 (1)

My Computer\HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\BagMRU\0

Figure 5: BagMRU key. Each bag is numerically named and represents a folder in the file system

Shell Bag MRU

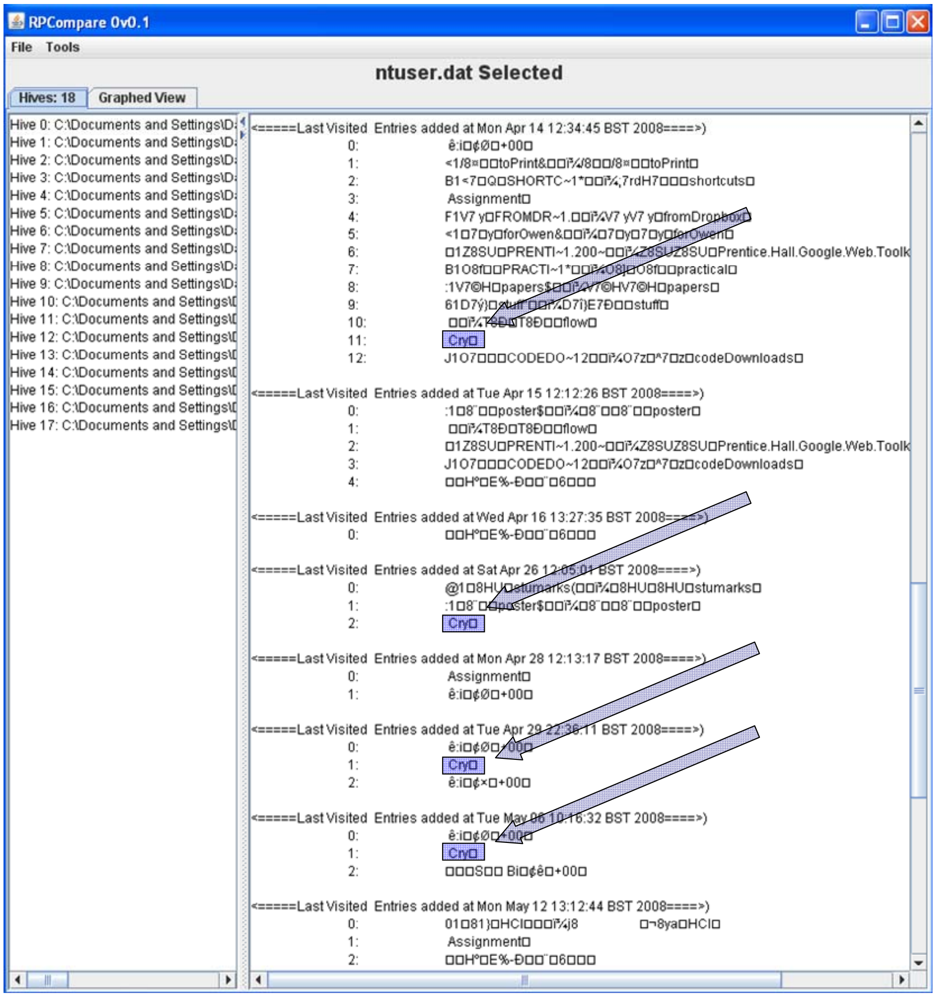
As mentioned above, a large number of 'ShellNoRoam' keys are continuously created and are responsible for the size increase in "ntuser.dat" hive. These keys store positional information of windows for each folder in the file system. Each folder (bag) has its own MRU storing information about its subfolders that were accessed most recently. If a folder is opened and the position of its window changed, then those coordinates are saved in the folders' bag. The parent MRU bag will be updated to reflect that the sub-folder has now been accessed the most recently. Not all folders have a bag. If a folder is never opened, it will not contain a representative bag. However, if the folder does have a bag and the folder is subsequently deleted, the bag will not be automatically deleted. The bags are therefore an important source of information for the folders in particular and user activity in general.

As shown in Figure 5, all bags are stored under the HKCU\Software\Microsoft\CurrentVersion\ShellNoRoam\ BagMRU key and do not contain the same name as the folders they represent. Rather, the bags are numerically named with the actual folder names being stored in the bag's parent key. In addition to storing all bags, the BagMRU key stores an MRU of the most recently accessed folders.

RPCompare is able to process this information and extract the user activity with respect to folders in a similar manner to the 'OpenSaveMRU'. However, since every folder of the system has its own MRU, these keys contain much more information than the OpenSaveMRUs. It allows the investigator to highlight which folders where opened by the user and which were the most widely used over a period of time.

RPCompare was executed on the BagMRU key to extract the timeline of the most accessed folders in the case study. Figure 6 shows a sample of the activity over a 4-week period. The folder 'Cry' is highlighted since it is present in 4 out of 8 Restore Points. This indicates that the screen position of the folder 'Cry' was changed at least 4 times in that time period. The user must have opened this folder and repositioned its window at least 4 times. The name of the folder is present more frequently than others and we can conclude that this folder is accessed more frequently than others. This is a significant user activity, if an investigator is looking for an activity relating to a suspiciously named folder, and this can be seen as vital incriminating evidence. Figure 6 also shows a folder called 'Assignment' being frequently accessed. Although it is not accessed as frequently as 'Cry', it provides a number of avenues for investigation. In this case, 'Cry' is a video games folder, while 'Assignment' is office related.

MRUs, for specific folders, which are found in the individual bags, are in a similar format. Using RPCompare, the investigator can extract any user activity relating to any folder in the system. In particular, deleted folders can be identified by correlating the bag names with the existing folders in the file system. The investigator can combine the evidence found in the Bag MRUs with those MRUs relating to the files to get more details about the user activity.



```
Hive 0: C:\Documents and Settings\D:
Hive 1: C:\Documents and Settings\D:
Hive 2: C:\Documents and Settings\D:
Hive 3: C:\Documents and Settings\D:
Hive 4: C:\Documents and Settings\D:
Hive 5: C:\Documents and Settings\D:
Hive 6: C:\Documents and Settings\D:
Hive 7: C:\Documents and Settings\D:
Hive 8: C:\Documents and Settings\D:
Hive 9: C:\Documents and Settings\D:
Hive 10: C:\Documents and Settings\[
Hive 11: C:\Documents and Settings\[
Hive 12: C:\Documents and Settings\[
Hive 13: C:\Documents and Settings\[
Hive 14: C:\Documents and Settings\[
Hive 15: C:\Documents and Settings\[
Hive 16: C:\Documents and Settings\[
Hive 17: C:\Documents and Settings\[
```

```
<=====Last Visited  Entries added at Mon Apr 14 12:34:45 BST 2008====>)
        0:        ê:i□¢Ø□+00□
        1:        <1/8¤□□toPrint&□□ï¾/8□□/8¤□□toPrint□
        2:        B1<7□Q□SHORTC~1*□□ï¾;7rdH7□□□shortcuts□
        3:        Assignment□
        4:        F1V7 y□FROMDR~1.□□ï¾V7 yV7 y□fromDropbox□
        5:        <1□7□y□forOwen&□□ï¾□7□y□7□y□forOwen□
        6:        □1Z8SU□PRENTI~1.200~□□ï¾Z8SUZ8SU□Prentice.Hall.Google.Web.Toolk
        7:        B1O8f□□PRACTI~1*□□ï¾O8]□O8f□□practical□
        8:        :1V7©H□papers$□□ï¾V7©HV7©H□papers□
        9:        61D7ý)□stuff□□ï¾D7î)E7Ð□□stuff□
        10:       □□ï¾T8Ð□T8Ð□□flow□
        11:       Cry□
        12:       J1O7□□□CODEDO~12□□ï¾O7z□^7□z□codeDownloads□

<=====Last Visited  Entries added at Tue Apr 15 12:12:26 BST 2008====>)
        0:        :1□8¨□□poster$□□ï¾□8¨□□8¨□□poster□
        1:        □□ï¾T8Ð□T8Ð□□flow□
        2:        □1Z8SU□PRENTI~1.200~□□ï¾Z8SUZ8SU□Prentice.Hall.Google.Web.Toolk
        3:        J1O7□□□CODEDO~12□□ï¾O7z□^7□z□codeDownloads□
        4:        □□H°□E%-Ð□□¨□6□□□

<=====Last Visited  Entries added at Wed Apr 16 13:27:35 BST 2008====>)
        0:        □□H°□E%-Ð□□¨□6□□□

<=====Last Visited  Entries added at Sat Apr 26 12:05:01 BST 2008====>)
        0:        @1□8HU□stumarks(□□ï¾□8HU□8HU□stumarks□
        1:        :1□8¨□□poster$□□ï¾□8¨□□8¨□□poster□
        2:        Cry□

<=====Last Visited  Entries added at Mon Apr 28 12:13:17 BST 2008====>)
        0:        Assignment□
        1:        ê:i□¢Ø□+00□

<=====Last Visited  Entries added at Tue Apr 29 22:36:11 BST 2008====>)
        0:        ê:i□¢Ø□+00□
        1:        Cry□
        2:        ê:i□¢×□+00□

<=====Last Visited  Entries added at Tue May 06 10:16:32 BST 2008====>)
        0:        ê:i□¢Ø□+00□
        1:        Cry□
        2:        □□□S□□ Bi□¢ê□+00□

<=====Last Visited  Entries added at Mon May 12 13:12:44 BST 2008====>)
        0:        01□81}□HCl□□□ï¾j8        □¬8ya□HCl□
        1:        Assignment□
        2:        □□H°□E%-Ð□□¨□6□□□
```

Figure 6: MRU list of the most recently accessed folders. Highlighting 'Cry' folder Access is done manually

4. ACTIVITY CATEGORISATION

The previous section detailed a methodology to make the comparison phase of RPCompare more efficient. In the same manner, this section will detail the RPCompare components used to increase the efficiency of the result interpretation phase.

To reduce the complexity of the interpretation, RPCompare classifies all the differences into categories. In particular, it classifies the differences into categories of evidence. Each evidence category relates to a specific type of activity that can be individually analysed. In conducting a case, the user can leave out any categories that are judged not to be relevant. To achieve this, RPCompare has been enhanced to recognise keys and automatically attribute them to one category or another. An informal categorisation was applied throughout the previous sections. Here we will detail only the rules used to reason about the test system.

1. Software Installation / Uninstallation

   Software vendors typically create their registry keys using the 'HKLM\Software\Manufacturer\Product\ Version' standard. All keys relating to the software are placed under this key and other vendors do not use keys that are found under a different vendor name. For instance, Figure 2 shows that the key "$$$PROTO.HIV\Software\DameWare Development\NT Utilities"[1] was added. Many of the 114 added keys fall under this main key. Therefore, it was correctly inferred that software called "NT Utilities" created by "DameWare Development" was installed in the system.

   Rule 1: *New software is installed in the system, if a new key is added to the "HKLM\Software" or the "HKCU\Software".*

   Similarly, if the "NT Utilities" key is removed, then one can conclude that the software is uninstalled.

2. Software Execution

   When a program is executed Windows normally changes/updates the registry keys that correspond to that program. Non-malicious software would not modify any key that belongs to another software vendor. In the DameWare Development example, it was correctly stated that DameWare was not executed much because no key was found to have been modified under its registry branch.

   Rule 2: *Software has been executed if one of its keys has been modified.*

   Some software programs use the registry very lightly and may not modify any of its keys if it is executed. However, this pattern can be used in conjunction with any other avenues of investigation.

3. New MRU Entries

[1] In registry hives, $$$PROTO.HIV is the binary version of the root key name and is replaced with the hive name by RegEdit when it is opened. In this case the hive is the "nutser.dat" and would be replaced by HKCU.

> In Windows XP each MRU contains a ‘list’ value called “MRUList” or “MRUListEx” listing file names in order of last access. This identifies the “OpenSaveMRU” and the “ShellNoRoam” keys discussed in Extraction of User Activity Section. The Registry MRU Management Section describes how MRUs are used.
>
> ***Rule 3:*** *If a key contains ‘MRU’ in its name or a value called “MRUList” or “MRUListEx”, then it is an MRU key.*
>
> Non Operating System MRUs do not follow the list naming convention or its list value is empty. Their order is maintained by numbering each of their elements. Currently, RPCompare identifies only MRU keys that follow the naming convention.

In addition, RPCompare can report if new USB or external hard drives have been connected to the computer or if new file extensions have been created for the system. The categories of evidence have been limited to a small number but can be expanded by adding more rules to RPCompare. Many of the rules are based on existing registry guides and documented roles of keys. The rules allow the results to be divided into evidence types and this operation reduces considerably the interpretation of the results.

5. RELATED WORK

Registry comparison softwares have existed for a number of years in the registry analysis fields (RegDiff (Ver3.3), WinDiff (Ver5.1) (2001)). Exact registry specifications have not been published by Microsoft. Therefore, one of the techniques to see what the function is of any key is to take snapshots of the registry before and after known activity and analyse the difference (Microsoft (2008), Honeycutt (2002)). However, these programs are limited in either the focus of the comparison and on what hives they operate on. Most of the registry comparison softwares are limited to comparing .REG files (ASCII versions of the binary registry hives). Each time a snapshot is taken, the relevant hive is exported with the inbuilt Microsoft RegEdit32 tool. This added step is undesirable in the forensic community where the goal is to avoid any modification of the original file. Other tools (RegDiff (Ver3.3)) can compare only the active registries and rely on commands provided by Microsoft Win32 API to extract relevant registry keys. Therefore they are unsuitable for offline registry analysis.

The tools mentioned above can only compare two hives at any one time and are not designed for digital forensic investigations. RPCompare differs to these programs since it has a digital forensic focus and aims to extract meaning out of the differences with respect to the user activity. It can parse any offline registry hive even when it is extracted from a live system independently from any API.

The investigators have for a long time acknowledged the value of analysing the registry for evidence (Carvey (2005), Carvey (2007)). Guides have been published explaining which keys are the most relevant to a particular investigation (ForensicMatter.com (2008)). Most current forensic suites, EnCase (Encase (Ver6.8) (2008)) or Forensic Toolkit (FTK (Ver1.62.1) (2008)), contain registry parsers that can parse any registry hive files and present the contents to the investigator for analysis. However, the forensic analysis of the restore points has been treated the same as the analysis of the active registry. Namely, the investigator must open the hive files manually and access the different registry keys.

Research into analysis of the registry with respect to retrieval of deleted data has been done by Morgan (2008) and Kim *et al.* (2008). The latter concentrated on retrieving still active keys not deleted by uninstalled programs. These clues are highly dependent on the uninstallation process of the software and may not reveal much information.

Research into Restore Points with respect to forensics has only been tackled recently (Bunting (2008), Carvey (2006), Harms (2006)). Harms has illustrated how the information stored in the Restore Points can be used to uncover evidence of a system intrusion. However, the author concentrates on the analysis of the “change.log” file only. This file is created at every Restore Point and tracks all files saved throughout the restore process. The registry hive files are not analysed.

6. CONCLUSION

This paper presented a new approach for extracting user activity in a digital forensic investigation. Namely, it focuses on comparisons of the Restore Points in general and the registry hives stored within them in particular. Differences between these hives can highlight changes in user activity that can be useful for the investigation. We introduce a tool, RPCompare; an offline, self contained and integrated environment that can compare Restore Points and registry hives and present the differences in a clear and logical interface. We also present a methodology using this tool to streamline the investigative process. Two techniques were presented in particular. The first focuses on the registry in its entirety and attempts to ascertain time points of high user activity. This activity includes what software was installed and removed and which keys were added or deleted relating to this activity. The technique, based on comparisons of hive size as well as content, is structured in a series of progressively detailed comparisons, which highlight areas of a user activity with progressively higher levels of accuracy. The technique guides the investigator away from time consuming wholesale hive comparison and into much more efficient selective hive and branch comparison.

The second technique focuses on the user trail and recovers and analyses the Most Recent Used (MRU) keys of the hives. The analysis of the MRUs requires specific processing to be investigated properly. The user activity is extracted with respect to ‘file open’ MRUs as well as ‘folder access’ MRUs to get a complete user trail. In particular we showed how RPCompare can reveal which folders and files the user accessed more frequently. Using both the timestamps of the MRUs and the hives, RPCompare presents an informative account of how the system was used in a clear and useful manner to the investigator.

7. FUTURE WORK

RPCompare will be further enhanced to streamline the techniques presented above and to add new functionality in both the comparison and interpretation phases. In this paper, we have concentrated mainly on the registry hives found within the Restore Points. Further work needs to be done to allow the software to utilise other similar files in the Restore Points and gather more information on what has changed between one point and another.

As described in the Large Scale Registry Comparison section, the investigator progressively narrows down their analysis of the registry entries by focusing on a smaller number of branches. At the start of this process, a large number of differences may be returned that may be irrelevant to the investigation. In the DameWare scenario for example, a large number of “HKLM\Software\Microsoft\Windows\ShellNoRoam\Bags\” keys were present. These keys store positional information on windows that the user has opened. Although it may be relevant to the investigator to parse these and extract useful information from them, in finding traces of added / removed programs, these keys are not relevant. Future development of RPCompare will include filters to remove unwanted keys from the results or mark them as being irrelevant to the case.

Future work will also be carried out to fully automate the interpretation phase. There is a number of registry guides that document specific registry keys relevant to forensics (ForensicMatter (2008)). These keys, as well as the roles they play in Windows XP will be encoded in RPCompare to allow automatic interpretation of the differences. The general categories discussed in the previous sections are not exhaustive and contain many sub categories that can further divide activities. Future development of RPCompare will include the addition of new categories organised in a hierarchical and logical manner.

The rules discussed in the Categorisation Section lack formality and cannot be used for automatic reasoning and inference. The rules will be formally specified and will relate to user activity to specific registry keys and locations. This will result in an ontology of concepts and rules which can be used to reason and infer new knowledge from existing data. Ontology of activity, relating different concepts of the criminal investigation to each other with formal attributes, will be developed to create a model of the investigation and will be used as a unifying, application independent source of information for all registry software, including RPCompare.

8. ACKNOWLEDGMENTS

The authors would like to thank the "Irish Research Council for Science, Engineering and Technology" (IRCSET) and Science Foundation Ireland (SFI) for providing funding that made this research possible. We also express our thanks to Dermot Kirby for his help with this research.

9. REFERENCES

Bunting, S. (2008) University of Delaware Police Computer Forensics Lab. *Restore Point Forensics*. URL http://128.175.24.251/forensics/restorepoints.htm, Accessed Mar 2009.

Carvey, H. (2005). "The Windows Registry as a Forensic Resource". *Digital Investigation*, 2(3), pp201-205

Carvey, H. (2006). "Restore Point Forensics". URL http://windowsir.blogspot.com/2006/10/restore-point-forensics.html, Accessed Mar 2009.

Carvey, H. (2007). "Registry Analysis", in *Windows Forensic Analysis DVD Toolkit*. Syngress Press, pp125-189

DameWare (Ver6.0). (2008) Dame Ware Development. URL http://www.dameware.com. Accessed: Mar 2009.

Encase (Ver6.8) (2008) Guidance Software Digital Investigations URL http://www.guidancesoftware.com/. Accessed Mar 2009.

ForensicMatter (2008). *Forensicmatter.com: Registry Hives*. Available at URL http://www.forensicsmatter.com/registry_hives.php. Accessed Mar 2009.

FTK (Ver1.62.1) (2008) Access Data, URL http://www.accessdata.com/. Accessed Mar 2009.

Harms, K. (2006). "Forensic Analysis of System Restore Points in Microsoft Windows XP". *Digital Investigation*, 3(3), pp151-158

Honeycutt, J. (2002) *Microsoft Windows XP Registry Guide*. Microsoft Press

Microsoft (2007). "Monitored File Extensions". URL http://msdn.microsoft.com/en-us/library/aa378870(VS.85).aspx. Accessed Mar 2009.

Microsoft (2008). "How to use WinDiff to Compare Registry Files". URL http://support.microsoft.com/kb/171780. Accessed Mar 2009.

Morgan, T.D. (2008) "Recovering Deleted Data from the Windows Registry". *Proceedings of Digital Forensic Research Workshop 2008*, pp33-42

RegDiff (Ver3.3). Available at URL http://p-nand-q.com/download/regdiff.html. Accessed Mar 2009.

Russinovich, M. (2008) “Inside the registry”. URL http://technet.microsoft.com/en-gb/library/cc750583.aspx. Accessed Mar 2009

Y. Kim et al (2008) “Suspects' Data Hiding at Remaining Registry Values of Uninstalled Programs”. *Proc. Of The 1st Int. Conference on Forensic Applications And Techniques In Telecommunications, Information, And Multimedia And Workshop*, 2008.

WinDiff (Ver5.1) (2001). Microsoft, Available at URL http://www.grigsoft.com/download-windiff.htm. Accessed Mar 2009