Automating Windows Registry correlation and interpretation

By Jacky Fox

A minor thesis submitted in part fulfilment of the degree of M.Sc. in Digital Investigation and Forensic Computing with the supervision of Dr. Fergus Toolan

School of Computer Science and Informatics

University College Dublin

24 August 2012

Abstract

The original idea for this dissertation developed from the observation that forensic triage solutions could benefit from automated registry artefact reporting. While researching this area, it became apparent that existing tools tend to report artefacts serially as found on a hive by hive basis and that the correlation and further interpretation of this data often has to be done manually. This posed a further question. Would large scale correlation of registry data highlight any information that is usually too difficult to access? To answer this question and to test if better cross hive correlation and interpretation could be automated; it was decided to develop some UNIX-based open source tools that would automate the reporting and correlation of registry artefacts. The development process involved assessing current registry reporting tools and researching other related peripheral areas such as process automation and admissibility. This scope grew as the project progressed, in order to provide ease of use and to incorporate some closely related non-registry artefacts. The end result was the production of five open source tools that collect and correlate registry artefacts across multiple operating systems.

It is hoped that these tools may be used for triage, live investigations and integrated into existing forensic toolkits. The tools will work with registry files extracted by any means, but to provide ease of use, processes have been outlined and scripted to extract files from commonly used digital evidence formats. The tools aspire to meet ACPO and Daubert standards and full documentation is provided on the operation and inner machinations of the tools to facilitate this. On completion of the tools, tests and experiments brought some items of interest to light as a direct result of the automated correlation.

Contents

1 Introduction

The main focus of this project was to explore how much automation could be applied to the correlation and interpretation of Windows registry artefacts and to establish if it is worthwhile? In order to answer both of these questions several areas needed exploration. An analysis of the currently available applications and tools offering registry parsing and interpretation was completed. An investigation into what areas of the registry hold potentially useful information was undertaken. This involved identifying commonly reported areas and searching the registry for related artefacts. Consideration was also given to the features required in a tool that would help ensure any generated report, has the best chance of being admissible in court.

1.1 Motivation

There were two motivating factors when selecting the subject of registry correlation. Firstly it was important that whatever was researched or produced, would be of potential use to the community. Secondly it needed to strike a balance between being an area of existing interest and having potential for learning through research and experimentation.

In order to fulfil the first criteria it was decided to identify something that was already being done and try to improve on it. Discussions both online and in person in the digital forensics community highlighted the issue of case backlogs. Several approaches to tackling these backlogs are being proposed, one of the technology based solutions is automated triage to identify and prioritise fruitful evidence which should in turn speed up case analysis. Automating more dataprocessing, by assisting with tasks that are currently either carried out manually or not at all, should theoretically help to highlight suspect systems and in turn reduce backlogs.

For the second criteria, assignments and cases to date identified the Windows registry as an area of personal interest. Designed as a common database for holding hardware, software and user details the Windows registry is a treasure trove of digital footprints. As it was not designed with forensic analysis in mind related artefacts are often dispersed in unrelated areas. The extraction and correlation of this data requires a combination of expertise and patience to perform the repetitive tasks involved, an ideal combination for providing an automated solution.

Further reading and discussions with other researchers cemented the view that this area had scope for enhancement. This was particularly evident in the open source field which is relevant to fulfilling the usefulness objective.

1.2 Idea of the problem and adopted approach

Commercial forensic software generally has some kind of registry reporting feature built in and several open source tools have been written to parse the registry. Typically these applications report registry key values sequentially and do little or no interpretation, organisation or correlation.

Most available open source tools work on extracted registry files meaning that the examiner has to manually extract the registry and associated files prior to usage. This was considered to be a barrier to usage from both a technical skills and a time consumption perspective.

The evidence from commercial closed source tools such as EnCase is currently usually accepted in court. However a serious challenge on disclosure of source code could affect this position and weight the argument even further in favour of open source tools.

Manual correlation of evidence such as USB device usage on a heavily used system is very time consuming. As many as twenty five different relevant pieces of information may be held on each USB device in seventeen different locations, all with differing indices such as serial numbers or guids (globally unique identifiers). On a system with as few as ten devices two hundred and fifty pieces of data may need to be interpreted and correlated to provide a profile of device usage.

The objective was to produce a set of open source tools that would;

- Automate the extraction of files for analysis from commonly used evidence storage formats.
- Correlate the evidence extracted and present the data sorted in a related manner rather than sorted by the source of its extraction.
- Support Windows XP, Vista and 7 without user intervention
- Require an understanding of the registry theory for usage but not memorised knowledge of the methodology to interpret the output.
- Produce reports that are admissible in court.
- Operate independently or as part of a triage suite.

To achieve this;

- The current state of the art will be reviewed.
- Any existing "facts" of relevance will be verified.
- Relevant artefacts and the manner in which they relate to each other will be identified.
- A body of test data will be gathered prior to the tool design; this will be used in both the design and the test phases.
- The tools should support hives supplied from any source, not just the built in extractor.
- The tools will be coded with usability and readability as a priority
- An attempt will be made to distribute the tools for feedback

1.3 Summary of achievements

The production of four registry tools which report related registry artefacts in a readable format. These tools have been tested with registries from Windows XP, Vista and 7 and focus on the areas of system, user, network and USB artefacts. All the tools have integrated help functions for added usability. Reports are output to the screen and logged. Two types of log are maintained, a copy of the screen dump and an additional "expert log" recording the data extracted prior to interpretation. A subsidiary set of scripts has also been created to assist with the extraction of registry files from common digital evidence formats such as EWF, AFF, dd or directly from a PC. These subsidiary tools also assist with the digital chain of custody by hashing files prior to collection.

The system tool generates a report that lists all the basic system information such as name, owner, time zone, system wide autoruns and installed applications.

The user tool generates a report that lists all users, their group membership, number of logons etc; it also lists the users by group. It then generates a report for each user hive in the target directory which details user profiles, userassist, user specific autoruns, network connections, Most recently used lists (MRUs) such as recentdocs, mediaplayer, opensave, typed URLs, terminal server clients etc.

The network tool generates a report that lists all network interface cards and their most recent settings. It also reports any greater network contact recorded in the registry such as Wireless access points (WAPs), network printers, shares etc.

The USB tool generates a report that lists each device’s serial number, vendor, drive letters, associated users and .lnk files, and several timestamps

Using these tools across multiple registries or in some cases the same registries at different time intervals has facilitated the observations below:

- When a USB device is inserted, all currently logged in users register the device.

- Automated searching and association of .lnk files with specific USB devices.

- Effects of the “global Enum” event on timestamps that are still commonly used for USB device time-lining.

- Counts in Userassist Windows 7, are for the current month.

- The users group always reports a count of two more users, than there are recorded within it.

It is also envisaged that the tools could prove effective when assessing theories and norms as part of general investigations.

1.4 Document organisation

The document has five main parts. Section 2 details the background information and literature survey. Section 3 defines the problem being tackled. Section 4 details the adopted approach to the problem and sections 5 details the specific tools and results. Section 6 discusses the outcome and suggests areas for further research.

2 Background Information and Literature Survey

The research topics have been divided into five specific areas, the Windows registry, process automation, existing tools, admissibility of forensics tools and more general information useful for formulating a specification. Various sources have been utilised to assess current wisdom including forums, journals, conferences, papers, books and search engines. As Digital Forensics is a relatively new discipline it was found that some areas of interest did not have a wide variety of academic sources. Non-academic sources of information have been acknowledged even if the work has not been widely published. Where sources have not been peer reviewed the validity of the information has been tested.

2.1 The Windows Registry

2.1.1 Registry specifications

The Windows registry is a hierarchical database that stores settings and general information relating to hardware, users, application and operating system software. This data can be related to configurations, preferences and also user experience. There are already excellent papers (1) (2) that discuss the architecture of Windows Registry in great detail, so this explanation is deliberately brief.

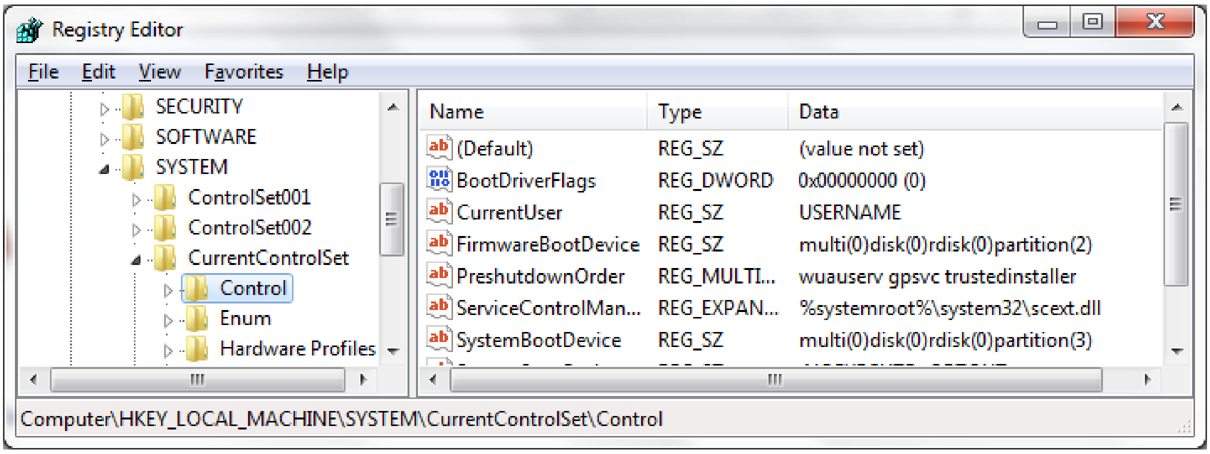
In early versions of Windows the operating system and each application, stored their settings and parameters in separate ".ini" files. These were all formatted differently and made interoperability challenging. In Windows 3.1, Microsoft introduced the registry as a common database that is used by most applications today. The registry is made up of several files referred to as hives, where information is stored and grouped broadly by function. There are various other backup and peripheral registry files but for the purposes of this exercise the registry files of interest are:

SYSTEM	Stores hardware and system related settings
SOFTWARE	Stores operating system and application settings
SAM	Security Account Manager - stores details about users and groups
SECURITY	Security policy
NTUSER.DAT	Stores user specific settings, created for each user at first logon

Fig 1 Registry hives of interest

With the exception of NTUSER.DAT, these files are typically found in the *%SystemRoot%\System32\Config* directory. NTUSER.DAT is stored in a users profile folder which is typically *%SystemRoot%\Documents and Settings\Username* for Windows XP and *%SystemRoot%\Users\Username* from Vista onwards.

Information is stored in tree format within the registry hives. The branches are called Keys which may have additional Subkeys or Values beneath them, much like a subdirectory and file structure. Keys are referred to by a handle, which states whether the key is system or user specific, the hive involved and the path to the values, see Fig 2. In the example HKLM\SYSTEM\CurrentControlSet\Control, HKLM means Handle to Key Local Machine, SYSTEM is the hive, followed by the path to the Values. Each data value such as "CurrentUser" within the key are named and have a specific data type. For example data type REG_SZ means that the data is a string and type REG_BINARY holds binary data.



Name	Type	Data
(Default)	REG_SZ	(value not set)
BootDriverFlags	REG_DWORD	0x00000000 (0)
CurrentUser	REG_SZ	USERNAME
FirmwareBootDevice	REG_SZ	multi(0)disk(0)rdisk(0)partition(2)
PreshutdownOrder	REG_MULTI...	wuauserv gpsvc trustedinstaller
ServiceControlMan...	REG_EXPAN...	%systemroot%\system32\scext.dll
SystemBootDevice	REG_SZ	multi(0)disk(0)rdisk(0)partition(3)

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control

Fig 2 Sample of Windows Registry structure viewed with Regedit

2.1.2 Identifying keys of interest

Research was started with Harlan Carvey's authoritative Windows Forensic Analysis (3) book which highlights and discusses important keys in the registry. Microsoft also provides information about registry structure and keys in their knowledge base. (4) As not all keys are Microsoft specific it would be impossible for them to provide an exhaustive reference. Microsoft's Regedt32 application was used to search the registry for interesting artefacts. Other registry reporting software was reviewed, to help compile a list of commonly used artefacts within each predefined group.

2.1.3 Why is the registry useful to a forensic examiner?

Common questions raised as part of a forensic examination on any system, are to analyse whether material was used on, introduced to or extracted from a system. Sample questions could be did *User X* gain unauthorised access to design blueprints and sell them or did *User Y* introduce illegal material to the company infrastructure. While not designed with digital forensics in mind the Windows Registry often stores evidence that can help answer these questions. Registry forensics can help to trace the history of user actions on a system for example by reporting the usage of applications, networks or USB devices. In accordance with Locard's exchange principle (5) of "every contact leaves a trace", some of the most important artefacts in the Windows registry are timestamps, which can tell a forensic examiner at what time an event occurred. An examiner needs to determine how to assemble something meaningful from these often disparate, pieces of information.

2.1.4 The extraction, reporting and interpretation of registry artefacts

The task of extracting registry artefacts can be a repetitive one. Alongside some case specific areas, the same general keys often need to be examined for each case. There are several applications and tools that will parse registry keys and their values. The current focus of registry reporting is on sequential reporting of the values as stored, on a hive by hive basis as opposed to relating or interpreting the data.

While this is useful, the examiner is still left to do a lot of manual correlation work. For example key pieces of information about USB devices are kept in three separate hives, SYSTEM, SOFTWARE and NTUSER.DAT. Few tools appear to do any correlation of the values; this is typically performed manually by the forensic examiner. In a system that has used twenty devices a considerable amount of manual correlation is required to link those artefacts.

Current tools vary considerably in their interpretation of registry artefacts. For example if the registry records that DHCP is enabled by assigning the number “00000001” to a value then this could be reported as “1” or as “yes”. The latter removes the need for manual interpretation or possible misinterpretation by an examiner. If the original data prior to interpretation is logged and the interpretation process is valid and fully documented, then it is reasonable to report data at the highest level of interpretation. This practice is widely accepted for timestamps but is often overlooked or not considered due to purism for other areas, leaving data in unnecessarily cryptic forms.

2.2 Process Automation

The goal of automation is to free humans from performing repetitive tasks. To be suitable for automation, a task must be capable of being broken down into subtasks where specific input produces prescribed results at each stage. A process that is suitable for automation can be documented by way of a logic flowchart. Not all tasks are suitable for automation. Some decisions have too many parameters or subtleties and require human input.

An automated process will be a reflection of the expertise of the designers and testers. To maintain standards new processes must be subject to peer review.

While not a strictly a scientific issue, automating tasks traditionally performed by humans, is not usually a popular suggestion, it is often viewed as a threat to employment (6).This attitude needs to be considered when reviewing arguments for and against automation of a specific process.

Automating a repetitive task to the highest level reduces the possibility of human error. For example why would you report a Unicode string “48 00 69 00”, when you could automate further interpretation and just say “Hi”? The amount of input required to produce a result, will have a proportionate relationship to the amount of mistakes that a human will make when initiating processes. This in turn influences the reliability of the output, so the less input an operator has to make, the less incorrect output, should be produced. No research reflecting the error rate of human correlation versus automated correlation could be found. However a study was found evaluating the methods of identifying deliberate errors in clinical laboratory results, comparing automated versus human quality control. This study showed that automation spotted 71.8-77.9% of errors versus humans spotting 23.9%-71.2% (7). This shows a huge variance in human ability to complete an automatable task. If this variance was transferred to registry correlation you could extrapolate that the same digital evidence could produce different reports based on the examiner and their effectiveness on the day. It is possible to put consistent error rates on automated processes. However it could prove challenging to assess the manual work of individual examiners. Note that in this study the worst automated result outperformed the best manual result.

Using automated tools to perform repetitive tasks would mean that forensic examiners should have more time to apply their skills to more complex or specialist areas.

In reality there are multiple considerations that a forensic examiner should review before creating or using an automated tool. To present evidence in court it is important for experts to understand their tools but they do not have a verbatim recollection of the theory involved every day on every investigation. Regardless of time saving, a question that should be satisfied is; does this automated process produce an accurate and understandable reflection of the evidence provided?

2.3 Identification and evaluation of existing tools

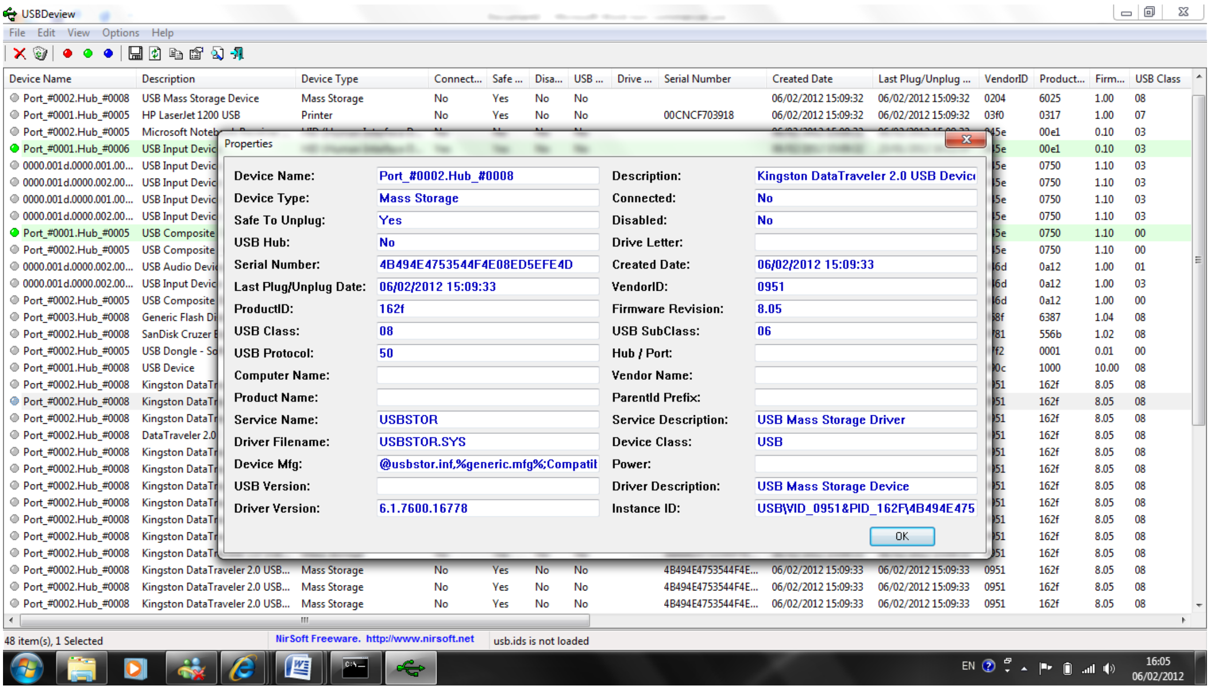
By far the majority of registry tools available focus on registry cleaning to speed up system operation, a few do some manner of registry reporting and fewer still claim do this in a forensically sound manner. There are several tools in the latter category both commercial and non-commercial, open and closed source. In order to discover if similar tools to those proposed had been previously published a survey was performed to identify and evaluate existing tools. This identified nine tools that appear to be commonly used by the forensic community. Four are closed source commercial products and five have non-commercial licensing arrangements but only two are open source. These tools were identified by using search engines and browsing forensic blogs and websites. All the tools identified that had a significant following and the potential to possess correlation features were evaluated.

The evaluation assessed the accuracy, comprehensiveness, level of correlation, license type, and forensic soundness of each tool. Particular attention was paid to any correlation performed. All the tools assessed were written with different objectives so perhaps it is somewhat unfair to judge them against this project's criteria. It should be noted that just because a tool does not perform correlation etc. does not mean that the tool does not fulfil its own objectives perfectly. With this in mind any comments made should not be viewed as criticisms. This is merely an attempt to discover the state of the art with regard to registry correlation. As the scope of the tools surveyed is large it was decided to focus on one area of interest namely the correlation of USB history for comparative purposes.

2.3.1 USBDeview

Version: 2.0 Size: 53760 bytes Modified time: 15/01/12 22:44:28
Source: www.nirsoft.net (06/02/12)
Licence type: Closed source freeware
Platform: Windows 2000 =>

USBDeview works on a live system, not post mortem registry files so any findings may not be reproducible. It reports from the system hive only so it does not retrieve the information stored in the software or user hives. It does do some correlation work between values rather than just parsing keys. The creation and last plug-in dates reported are from HKLM\System\CurrentControlSet\Enum\USB\VID&PID. USBDeview accurately reports what is stored in the registry in this location but this key is not always an accurate representation of the last plug-in time. Only devices accessed since the global enum event’s broad timestamp change are accurate. (See section 5.2.5.3 for more details on this) User information, guids etc. are not reported.



Device Name	Description	Device Type	Connect...	Safe ...	Disa...	USB ...	Drive ...	Serial Number	Created Date	Last Plug/Unplug ...	VendorID	Product...	Firm...	USB Class
Port_#0002.Hub_#0008	USB Mass Storage Device	Mass Storage	No	Yes	No	No			06/02/2012 15:09:32	06/02/2012 15:09:32	0204	6025	1.00	08
Port_#0001.Hub_#0005	HP LaserJet 1200 USB	Printer	No	Yes	No	No		00CNCF703918	06/02/2012 15:09:32	06/02/2012 15:09:32	03f0	0317	1.00	07
Port_#0002.Hub_#0005	Microsoft Noteb										45e	00e1	0.10	03
Port_#0001.Hub_#0006	USB Input Devic										5e	00e1	0.10	03
0000.001d.0000.001.00...	USB Input Devic										5e	0750	1.10	03
0000.001d.0000.002.00...	USB Input Devic										5e	0750	1.10	03
0000.001d.0000.001.00...	USB Input Devic										5e	0750	1.10	03
0000.001d.0000.002.00...	USB Input Devic										5e	0750	1.10	03
Port_#0001.Hub_#0005	USB Composite										5e	0750	1.10	00
Port_#0002.Hub_#0005	USB Composite										5e	0750	1.10	00
0000.001d.0000.002.00...	USB Audio Devi										6d	0a12	1.00	01
0000.001d.0000.002.00...	USB Input Devic										6d	0a12	1.00	03
Port_#0002.Hub_#0005	USB Composite										6d	0a12	1.00	00
Port_#0003.Hub_#0008	Generic Flash Di										8f	6387	1.04	08
Port_#0002.Hub_#0008	SanDisk Cruzer										81	556b	1.02	08
Port_#0002.Hub_#0005	USB Dongle - So										f2	0001	0.01	00
Port_#0001.Hub_#0008	USB Device										0c	1000	10.00	08
Port_#0002.Hub_#0008	Kingston DataTr										51	162f	8.05	08
Port_#0002.Hub_#0008	Kingston DataTr										51	162f	8.05	08
Port_#0002.Hub_#0008	Kingston DataTr										51	162f	8.05	08
Port_#0002.Hub_#0008	DataTraveler 2.0										51	162f	8.05	08
Port_#0002.Hub_#0008	Kingston DataTr										51	162f	8.05	08
Port_#0002.Hub_#0008	Kingston DataTr										51	162f	8.05	08
Port_#0002.Hub_#0008	Kingston DataTr										51	162f	8.05	08
Port_#0002.Hub_#0008	Kingston DataTr										51	162f	8.05	08
Port_#0002.Hub_#0008	Kingston DataTr										51	162f	8.05	08
Port_#0002.Hub_#0008	Kingston DataTr										51	162f	8.05	08
Port_#0002.Hub_#0008	Kingston DataTr										51	162f	8.05	08
Port_#0002.Hub_#0008	Kingston DataTraveler 2.0 USB...	Mass Storage	No	Yes	No	No		4B494E4753544F4E...	06/02/2012 15:09:33	06/02/2012 15:09:33	0951	162f	8.05	08
Port_#0002.Hub_#0008	Kingston DataTraveler 2.0 USB...	Mass Storage	No	Yes	No	No		4B494E4753544F4E...	06/02/2012 15:09:33	06/02/2012 15:09:33	0951	162f	8.05	08
Port_#0002.Hub_#0008	Kingston DataTraveler 2.0 USB...	Mass Storage	No	Yes	No	No		4B494E4753544F4E...	06/02/2012 15:09:33	06/02/2012 15:09:33	0951	162f	8.05	08

Properties

Device Name:	Port_#0002.Hub_#0008	Description:	Kingston DataTraveler 2.0 USB Devic
Device Type:	Mass Storage	Connected:	No
Safe To Unplug:	Yes	Disabled:	No
USB Hub:	No	Drive Letter:	
Serial Number:	4B494E4753544F4E08ED5EFE4D	Created Date:	06/02/2012 15:09:33
Last Plug/Unplug Date:	06/02/2012 15:09:33	VendorID:	0951
ProductID:	162f	Firmware Revision:	8.05
USB Class:	08	USB SubClass:	06
USB Protocol:	50	Hub / Port:	
Computer Name:		Vendor Name:	
Product Name:		ParentId Prefix:	
Service Name:	USBSTOR	Service Description:	USB Mass Storage Driver
Driver Filename:	USBSTOR.SYS	Device Class:	USB
Device Mfg:	@usbstor.inf,%generic.mfg%;Compatil	Power:	
USB Version:		Driver Description:	USB Mass Storage Device
Driver Version:	6.1.7600.16778	Instance ID:	USB\VID_0951&PID_162F\4B494E475

48 item(s), 1 Selected NirSoft Freeware. http://www.nirsoft.net usb.ids is not loaded

Fig 3 USBDeview screenshot showing fields displayed

2.3.2 Registry Decoder

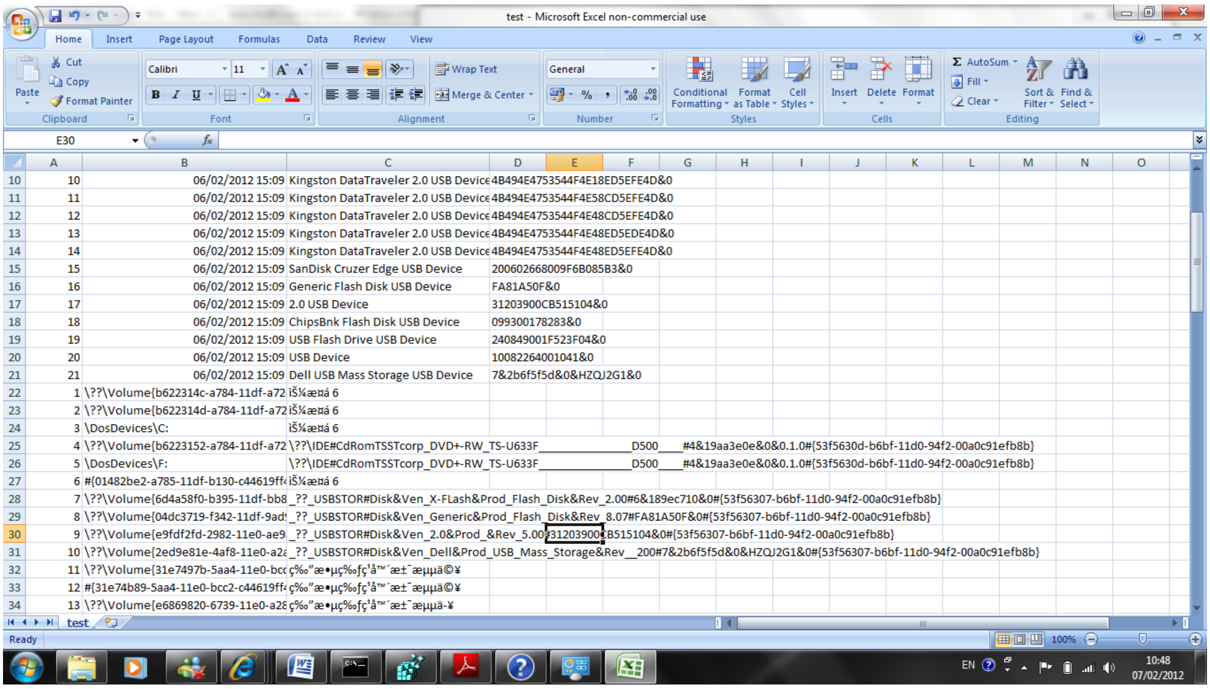
Version: 1.2 Size: 14,888 kbytes Modified time: 02/02/12 2:08

Source: http://code.google.com/p/registrydecoder/downloads (06/02/12)

Licence type: Open source GNU GPL v2

Platform: Windows

Registry decoder has two elements, registry acquisition and offline analysis. Registry acquisition works by triggering a restore point and retrieving a copy of the latest registry from there. This action could destroy forensic artefacts by deleting the oldest restore point or writing what could be a considerable amount of information over slackspace. Some stated limitations are that it doesn't work if system protection is disabled, the default on Windows 7, or if SP is unsupported like in Windows server 2003. Registry analysis consists of reporting key contents by way of plugins. The output is in CSV and no correlation work is done.



	A	B	C	D
10	10	06/02/2012 15:09	Kingston DataTraveler 2.0 USB Device	4B494E4753544F4E18ED5EFE4D&0
11	11	06/02/2012 15:09	Kingston DataTraveler 2.0 USB Device	4B494E4753544F4E58CD5EFE4D&0
12	12	06/02/2012 15:09	Kingston DataTraveler 2.0 USB Device	4B494E4753544F4E48CD5EFE4D&0
13	13	06/02/2012 15:09	Kingston DataTraveler 2.0 USB Device	4B494E4753544F4E48ED5EDE4D&0
14	14	06/02/2012 15:09	Kingston DataTraveler 2.0 USB Device	4B494E4753544F4E48ED5EFE4D&0
15	15	06/02/2012 15:09	SanDisk Cruzer Edge USB Device	200602668009F6B085B3&0
16	16	06/02/2012 15:09	Generic Flash Disk USB Device	FA81A50F&0
17	17	06/02/2012 15:09	2.0 USB Device	31203900CB515104&0
18	18	06/02/2012 15:09	ChipsBnk Flash Disk USB Device	099300178283&0
19	19	06/02/2012 15:09	USB Flash Drive USB Device	240849001F523F04&0
20	20	06/02/2012 15:09	USB Device	10082264001041&0
21	21	06/02/2012 15:09	Dell USB Mass Storage USB Device	7&2b6f5f5d&0&HZQJ2G1&0
22	1	\??\Volume{b622314c-a784-11df-a72	iŠ¼æ¤á 6	
23	2	\??\Volume{b622314d-a784-11df-a72	iŠ¼æ¤á 6	
24	3	\DosDevices\C:	iŠ¼æ¤á 6	
25	4	\??\Volume{b6223152-a784-11df-a72	\??\IDE#CdRomTSSTcorp_DVD+-RW_TS-U633F________________D500____#4&19aa3e0e&0&0.1.0#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}	
26	5	\DosDevices\F:	\??\IDE#CdRomTSSTcorp_DVD+-RW_TS-U633F________________D500____#4&19aa3e0e&0&0.1.0#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}	
27	6	#{01482be2-a785-11df-b130-c44619ff	iŠ¼æ¤á 6	
28	7	\??\Volume{6d4a58f0-b395-11df-bb8	_??_USBSTOR#Disk&Ven_X-FLash&Prod_Flash_Disk&Rev_2.00#6&189ec710&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}	
29	8	\??\Volume{04dc3719-f342-11df-9ad	_??_USBSTOR#Disk&Ven_Generic&Prod_Flash_Disk&Rev_8.07#FA81A50F&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}	
30	9	\??\Volume{e9fdf2fd-2982-11e0-ae9	_??_USBSTOR#Disk&Ven_2.0&Prod_&Rev_5.00#31203900CB515104&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}	
31	10	\??\Volume{2ed9e81e-4af8-11e0-a2a	_??_USBSTOR#Disk&Ven_Dell&Prod_USB_Mass_Storage&Rev__200#7&2b6f5f5d&0&HZQJ2G1&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}	
32	11	\??\Volume{31e7497b-5aa4-11e0-bcc	ç‰"æ•µç‰ƒç¹å™´æ±¯æµµä©¥	
33	12	#{31e74b89-5aa4-11e0-bcc2-c44619ff	ç‰"æ•µç‰ƒç¹å™´æ±¯æµµä©¥	
34	13	\??\Volume{e6869820-6739-11e0-a28	ç‰"æ•µç‰ƒç¹å™´æ±¯æµµä-¥	

Fig 4 Registry decoder sample report

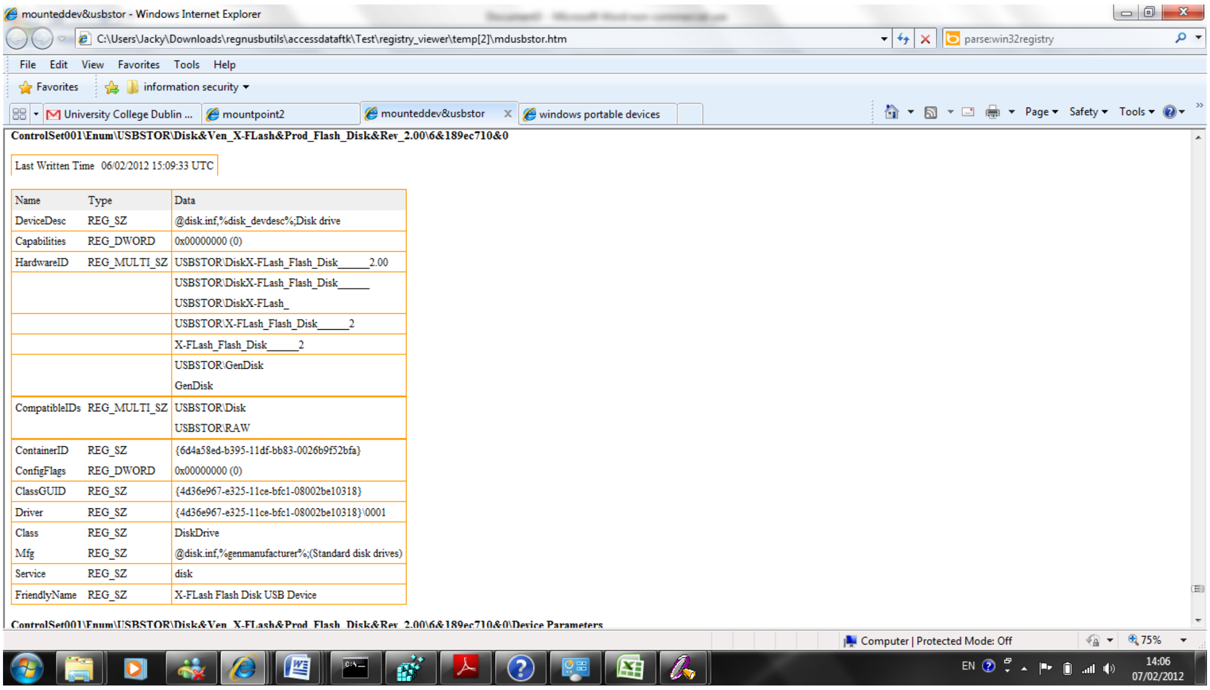
A small experiment was performed on a Windows 7 system to see the impact of generating a restore point. Firstly system protection was not enabled on the system so this was enabled and a restore point – RP1 was created. The system was used for a couple of days and then two more restore points RP2 and RP3 were created in close succession with no user activity in-between. The free space on the disk pre and post restore points was recorded. No other activities or processes that would change disk contents were reported by task manager during the experiment. The intention here is to document rather than explain the findings.

Status	Free space in bytes	Difference in bytes
Pre RP2	74,869,706,752	-
Post RP2	74,903,445,504	-33,738,752
Post RP3	74,698,407,936	+205,037,568

2.3.3 Accessdata Registry Viewer

Version: 1.5.2.34 Size: 2,156 kbytes Modified time: 26/06/08 17:17
Source: Accessdata installation CD
Licence type: Closed source and licensed - protected by Dongle
Platform: Windows

Accessdata's RV can be run against a raw image or extracted registry files. Accessdata's FTK can also export registry files from an image. RV allows you to generate customised reports based on user selected registry keys. Sample HTML reports generated from USB related keys show that the tool parses the registry comprehensively and accurately. However no automated correlation is performed, this needs to be done manually. This can be an ominous task on a heavily used system. One of the sample hive sets used for testing records forty-five separate USB devices. With nineteen identified pieces of information held about a USB device over six separate key trees there is a lot of manual correlation required for one device, never mind forty-five. There is no facility to run reports across multiple hives making even manual correlation cumbersome given USB device history can be stored over three separate hives.



C:\Users\Jacky\Downloads\regnusbutils\accessdataftk\Test\registry_viewer\temp[2]\mdusbstor.htm

ControlSet001\Enum\USBSTOR\Disk&Ven_X-FLash&Prod_Flash_Disk&Rev_2.00\6&189ec710&0

Last Written Time 06/02/2012 15:09:33 UTC

Name	Type	Data
DeviceDesc	REG_SZ	@disk.inf,%disk_devdesc%;Disk drive
Capabilities	REG_DWORD	0x00000000 (0)
HardwareID	REG_MULTI_SZ	USBSTOR\DiskX-FLash_Flash_Disk______2.00
		USBSTOR\DiskX-FLash_Flash_Disk______ USBSTOR\DiskX-FLash_
		USBSTOR\X-FLash_Flash_Disk______2
		X-FLash_Flash_Disk______2
		USBSTOR\GenDisk GenDisk
CompatibleIDs	REG_MULTI_SZ	USBSTOR\Disk USBSTOR\RAW
ContainerID	REG_SZ	{6d4a58ed-b395-11df-bb83-0026b9f52bfa}
ConfigFlags	REG_DWORD	0x00000000 (0)
ClassGUID	REG_SZ	{4d36e967-e325-11ce-bfc1-08002be10318}
Driver	REG_SZ	{4d36e967-e325-11ce-bfc1-08002be10318}\0001
Class	REG_SZ	DiskDrive
Mfg	REG_SZ	@disk.inf,%genmanufacturer%;(Standard disk drives)
Service	REG_SZ	disk
FriendlyName	REG_SZ	X-FLash Flash Disk USB Device

ControlSet001\Enum\USBSTOR\Disk&Ven_X-FLash&Prod_Flash_Disk&Rev_2.00\6&189ec710&0\Device Parameters

mounteddev&usbstor - Windows Internet Explorer
parse:win32registry
File Edit View Favorites Tools Help
Favorites
information security
University College Dublin ...
mountpoint2
mounteddev&usbstor
windows portable devices
Page
Safety
Tools
Computer | Protected Mode: Off
75%
EN
14:06
07/02/2012

Fig 5 Accessdata Registry Viewer SYSTEM USBSTOR sample

2.3.4 MiTec Windows Registry Recovery

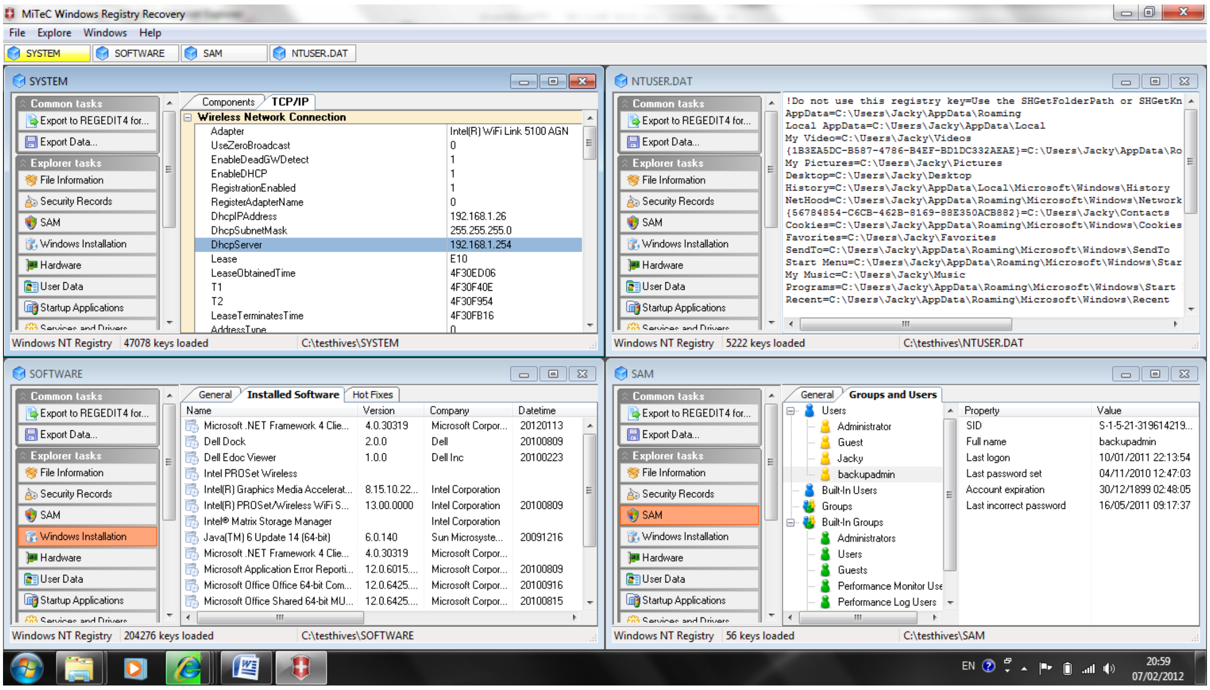
Version: 1.5.2.0 Size: 2,170 kbytes Modified time: 10/02/11 10:48

Source: http:// http://www.mitec.cz/wrr.html (07/02/12)

Licence type: Closed source freeware

Platform: Windows 2000=>

This is an offline registry parser. This utility does not report on any USB history. However it does do a good job of reporting some registry areas so it was worth including as a sample of available tools. The presentation of data is excellent but it could be more comprehensive, for example by reporting the number of times a user logged in from the SAM. Again no correlation is performed, for example by reporting when a network profile was first created.



SYSTEM

Wireless Network Connection

Adapter	Intel(R) WiFi Link 5100 AGN
UseZeroBroadcast	0
EnableDeadGWDetect	1
EnableDHCP	1
RegistrationEnabled	1
RegisterAdapterName	0
DhcpIPAddress	192.168.1.26
DhcpSubnetMask	255.255.255.0
DhcpServer	192.168.1.254
Lease	E10
LeaseObtainedTime	4F30ED06
T1	4F30F40E
T2	4F30F954
LeaseTerminatesTime	4F30FB16
AddressType	0

Windows NT Registry 47078 keys loaded C:\testhives\SYSTEM

NTUSER.DAT

```
!Do not use this registry key=Use the SHGetFolderPath or SHGetKn
AppData=C:\Users\Jacky\AppData\Roaming
Local AppData=C:\Users\Jacky\AppData\Local
My Video=C:\Users\Jacky\Videos
{1B3EA5DC-B587-4786-B4EF-BD1DC332AEAE}=C:\Users\Jacky\AppData\Ro
My Pictures=C:\Users\Jacky\Pictures
Desktop=C:\Users\Jacky\Desktop
History=C:\Users\Jacky\AppData\Local\Microsoft\Windows\History
NetHood=C:\Users\Jacky\AppData\Roaming\Microsoft\Windows\Network
{56784854-C6CB-462B-8169-88E350ACB882}=C:\Users\Jacky\Contacts
Cookies=C:\Users\Jacky\AppData\Roaming\Microsoft\Windows\Cookies
Favorites=C:\Users\Jacky\Favorites
SendTo=C:\Users\Jacky\AppData\Roaming\Microsoft\Windows\SendTo
Start Menu=C:\Users\Jacky\AppData\Roaming\Microsoft\Windows\Star
My Music=C:\Users\Jacky\Music
Programs=C:\Users\Jacky\AppData\Roaming\Microsoft\Windows\Start
Recent=C:\Users\Jacky\AppData\Roaming\Microsoft\Windows\Recent
```

Windows NT Registry 5222 keys loaded C:\testhives\NTUSER.DAT

SOFTWARE

Name	Version	Company	Datetime
Microsoft .NET Framework 4 Clie...	4.0.30319	Microsoft Corpor...	20120113
Dell Dock	2.0.0	Dell	20100809
Dell Edoc Viewer	1.0.0	Dell Inc	20100223
Intel PROSet Wireless			
Intel(R) Graphics Media Accelerat...	8.15.10.22...	Intel Corporation	
Intel(R) PROSet/Wireless WiFi S...	13.00.0000	Intel Corporation	20100809
Intel® Matrix Storage Manager		Intel Corporation	
Java(TM) 6 Update 14 (64-bit)	6.0.140	Sun Microsyste...	20091216
Microsoft .NET Framework 4 Clie...	4.0.30319	Microsoft Corpor...	
Microsoft Application Error Reporti...	12.0.6015....	Microsoft Corpor...	20100809
Microsoft Office Office 64-bit Com...	12.0.6425....	Microsoft Corpor...	20100916
Microsoft Office Shared 64-bit MU...	12.0.6425....	Microsoft Corpor...	20100815

Windows NT Registry 204276 keys loaded C:\testhives\SOFTWARE

SAM

Users
- Administrator
- Guest
- Jacky
- backupadmin

Built-In Users

Groups

Built-In Groups
- Administrators
- Users
- Guests
- Performance Monitor Use
- Performance Log Users

Property	Value
SID	S-1-5-21-319614219...
Full name	backupadmin
Last logon	10/01/2011 22:13:54
Last password set	04/11/2010 12:47:03
Account expiration	30/12/1899 02:48:05
Last incorrect password	16/05/2011 09:17:37

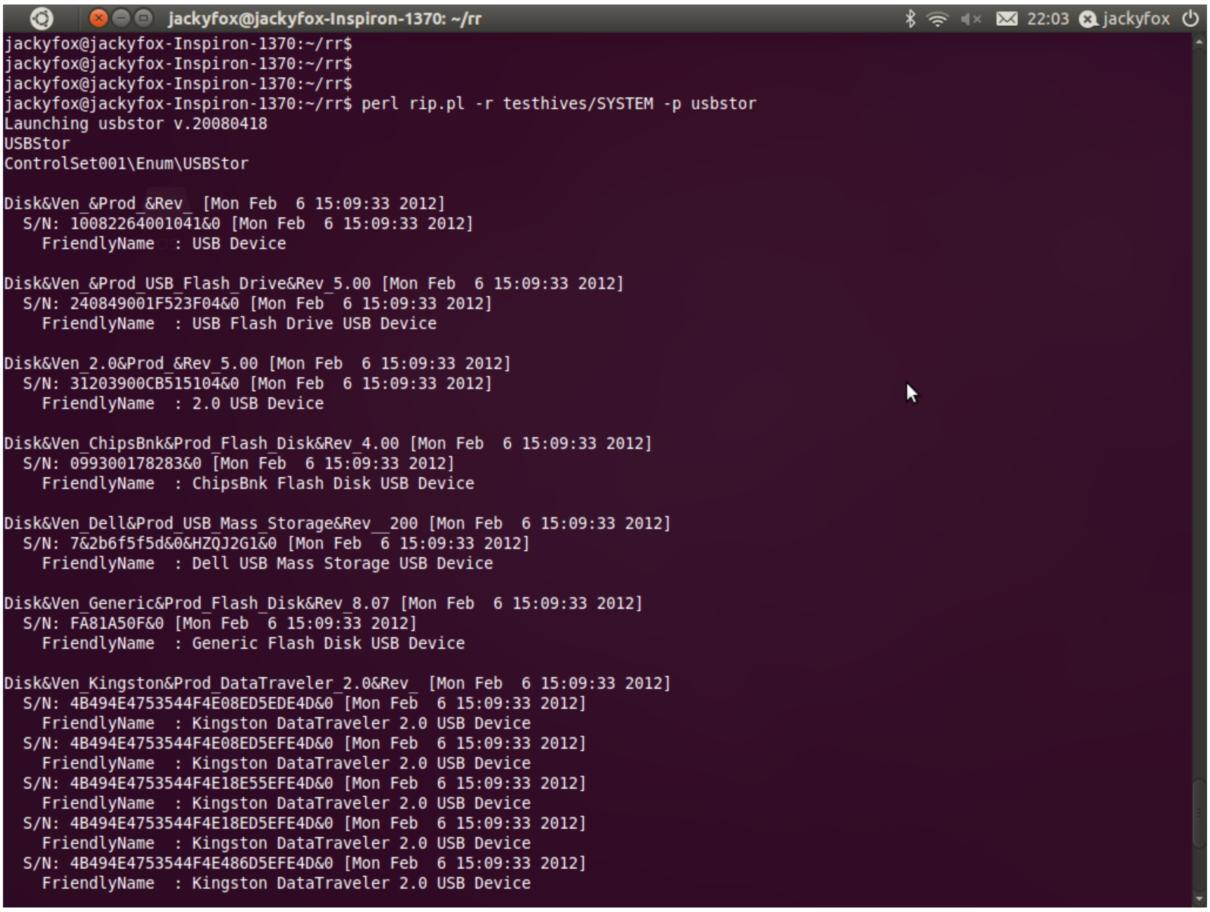
Windows NT Registry 56 keys loaded C:\testhives\SAM

Fig 6 MiTec Windows Registry Recovery sample reports by hive

2.3.5 Regripper

Version: Feb 10th 2011 Size: 7.4MB (download) Modified time: Not available
Source: www.code.google.com/p/winforensicaanalysis
Licence type: Open source
Platform: Anything that can run Perl (tests run on Ubuntu Linux)

Regripper, created by Harlan Carvey is a registry parser that works on extracted hives and utilises James Macfarlane's Parse:Win32Registry. It is written in Perl and is designed to query and report on registry keys by way of add on modules or plug-ins. A large and ever growing plug-in library is available. Ten plug-ins, relating to USB history were reviewed. Typically plug-ins extract from one hive at a time. The output from the plug-ins reviewed was comprehensive and accurate. The testing was done using rip.pl. No published plug-in has been generated that does cross-hive correlation in relation to USB history, manual correlation is still required.



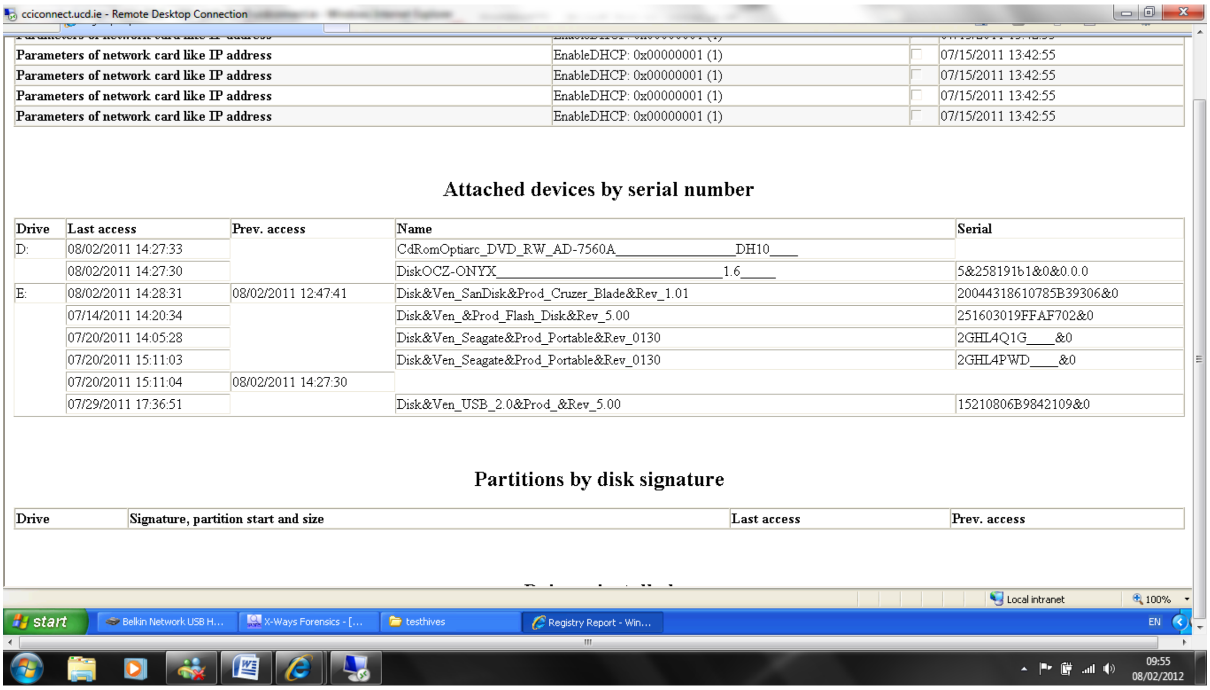
```
jackyfox@jackyfox-Inspiron-1370:~/rr$
jackyfox@jackyfox-Inspiron-1370:~/rr$
jackyfox@jackyfox-Inspiron-1370:~/rr$
jackyfox@jackyfox-Inspiron-1370:~/rr$ perl rip.pl -r testhives/SYSTEM -p usbstor
Launching usbstor v.20080418
USBStor
ControlSet001\Enum\USBStor

Disk&Ven_&Prod_&Rev_ [Mon Feb  6 15:09:33 2012]
  S/N: 100822640010410&0 [Mon Feb  6 15:09:33 2012]
    FriendlyName  : USB Device

Disk&Ven_&Prod_USB_Flash_Drive&Rev_5.00 [Mon Feb  6 15:09:33 2012]
  S/N: 240849001F523F04&0 [Mon Feb  6 15:09:33 2012]
    FriendlyName  : USB Flash Drive USB Device

Disk&Ven_2.0&Prod_&Rev_5.00 [Mon Feb  6 15:09:33 2012]
  S/N: 31203900CB515104&0 [Mon Feb  6 15:09:33 2012]
    FriendlyName  : 2.0 USB Device

Disk&Ven_ChipsBnk&Prod_Flash_Disk&Rev_4.00 [Mon Feb  6 15:09:33 2012]
  S/N: 0993001782830&0 [Mon Feb  6 15:09:33 2012]
    FriendlyName  : ChipsBnk Flash Disk USB Device

Disk&Ven_Dell&Prod_USB_Mass_Storage&Rev__200 [Mon Feb  6 15:09:33 2012]
  S/N: 7&2b6f5f5d&0&HZQJ2G1&0 [Mon Feb  6 15:09:33 2012]
    FriendlyName  : Dell USB Mass Storage USB Device

Disk&Ven_Generic&Prod_Flash_Disk&Rev_8.07 [Mon Feb  6 15:09:33 2012]
  S/N: FA81A50F&0 [Mon Feb  6 15:09:33 2012]
    FriendlyName  : Generic Flash Disk USB Device

Disk&Ven_Kingston&Prod_DataTraveler_2.0&Rev_ [Mon Feb  6 15:09:33 2012]
  S/N: 4B494E4753544F4E08ED5EDE4D&0 [Mon Feb  6 15:09:33 2012]
    FriendlyName  : Kingston DataTraveler 2.0 USB Device
  S/N: 4B494E4753544F4E08ED5EFE4D&0 [Mon Feb  6 15:09:33 2012]
    FriendlyName  : Kingston DataTraveler 2.0 USB Device
  S/N: 4B494E4753544F4E18E55EFE4D&0 [Mon Feb  6 15:09:33 2012]
    FriendlyName  : Kingston DataTraveler 2.0 USB Device
  S/N: 4B494E4753544F4E18ED5EFE4D&0 [Mon Feb  6 15:09:33 2012]
    FriendlyName  : Kingston DataTraveler 2.0 USB Device
  S/N: 4B494E4753544F4E486D5EFE4D&0 [Mon Feb  6 15:09:33 2012]
    FriendlyName  : Kingston DataTraveler 2.0 USB Device
```

Fig 7 Sample regripper usbstor script

2.3.6 X-ways

Version: 15.7 SR3 Size: 2.11 Mbytes Modified time: 19/08/10 9:47:43
Source: UCD (08/02/12)
Licence type: Closed source licensed with dongle
Platform: Windows

Registry reports can be generated in X-ways via the registry viewer component using the create report option. A template file "Reg Report.txt" is run against the selected hive and will report on any of the predetermined keys from the template that exist in that hive. The data produced is accurate. In relation to USB devices there is a limited amount of correlation done by serial number between HKLM\SYSTEM\MountedDevices and HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR. X-ways does no cross-hive correlation and does not report items such as first install date, guids, the user(s) associated with USB device etc.



Parameters of network card like IP address	EnableDHCP: 0x00000001 (1)	□	07/15/2011 13:42:55
Parameters of network card like IP address	EnableDHCP: 0x00000001 (1)	□	07/15/2011 13:42:55
Parameters of network card like IP address	EnableDHCP: 0x00000001 (1)	□	07/15/2011 13:42:55
Parameters of network card like IP address	EnableDHCP: 0x00000001 (1)	□	07/15/2011 13:42:55

Attached devices by serial number

Drive	Last access	Prev. access	Name	Serial
D:	08/02/2011 14:27:33		CdRomOptiarc_DVD_RW_AD-7560A________________DH10____	
	08/02/2011 14:27:30		DiskOCZ-ONYX______________________________1.6_____	5&258191b1&0&0.0.0
E:	08/02/2011 14:28:31	08/02/2011 12:47:41	Disk&Ven_SanDisk&Prod_Cruzer_Blade&Rev_1.01	20044318610785B39306&0
	07/14/2011 14:20:34		Disk&Ven_&Prod_Flash_Disk&Rev_5.00	251603019FFAF702&0
	07/20/2011 14:05:28		Disk&Ven_Seagate&Prod_Portable&Rev_0130	2GHL4Q1G____&0
	07/20/2011 15:11:03		Disk&Ven_Seagate&Prod_Portable&Rev_0130	2GHL4PWD____&0
	07/20/2011 15:11:04	08/02/2011 14:27:30		
	07/29/2011 17:36:51		Disk&Ven_USB_2.0&Prod_&Rev_5.00	15210806B9842109&0

Partitions by disk signature

Drive	Signature, partition start and size	Last access	Prev. access

Fig 8 X-ways showing limited correlation by serial number

2.3.7 Paraben P2 Commander

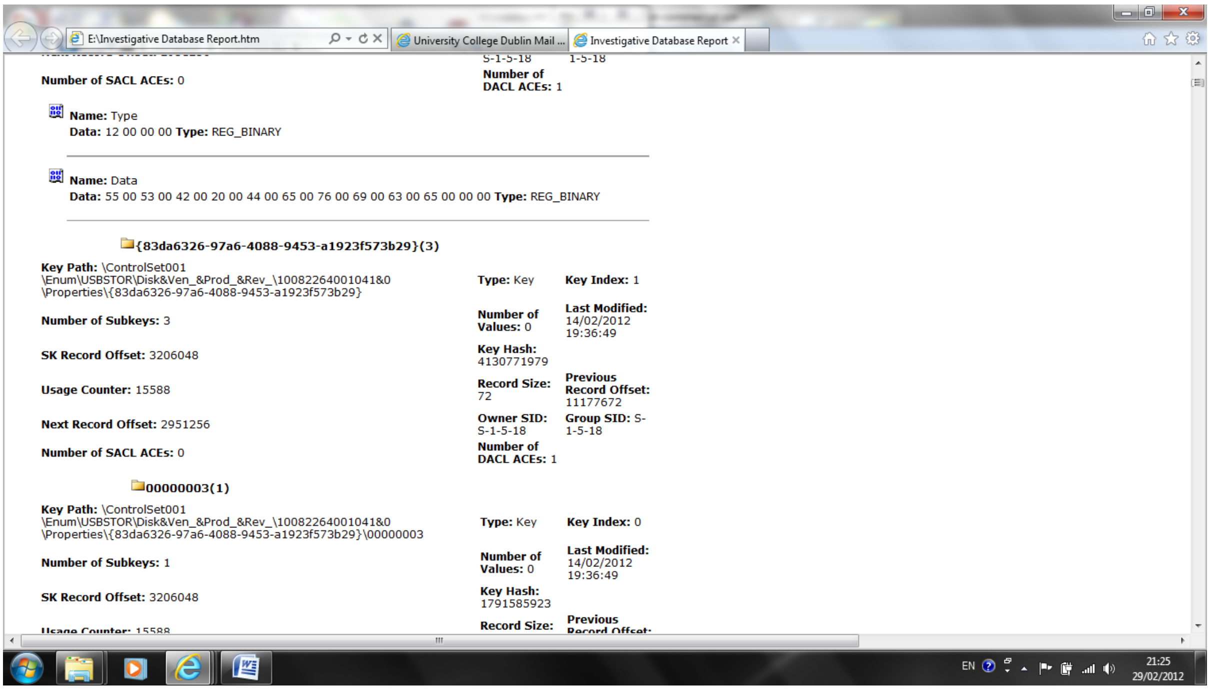
Version: 2.0 Size: 1.81 Mbytes Modified time: Not available

Source: www.paraben.com

Licence type: Closed source licensed (trial version)

Platform: Windows

Registry files can be accessed from an added image or as single evidence files. P2 Commander provides standard registry tree type viewing and reports can be generated from user selected keys. To get an overview of USB History each relevant key must be selected for inclusion in the report and the correlation must be performed manually by the examiner. Use of the software is intuitive and the report generation is fast and easy to operate.


E:\Investigative Database Report.htm
University College Dublin Mail ...
Investigative Database Report
EN
21:25
29/02/2012

```
                                                       S-1-5-18      1-5-18
Number of SACL ACEs: 0                                 Number of
                                                       DACL ACEs: 1

   Name: Type
   Data: 12 00 00 00 Type: REG_BINARY

   Name: Data
   Data: 55 00 53 00 42 00 20 00 44 00 65 00 76 00 69 00 63 00 65 00 00 00 Type: REG_BINARY

          {83da6326-97a6-4088-9453-a1923f573b29}(3)

Key Path: \ControlSet001
\Enum\USBSTOR\Disk&Ven_&Prod_&Rev_\10082264001041&0   Type: Key     Key Index: 1
\Properties\{83da6326-97a6-4088-9453-a1923f573b29}

                                                       Number of     Last Modified:
Number of Subkeys: 3                                   Values: 0     14/02/2012
                                                                     19:36:49
SK Record Offset: 3206048                              Key Hash:
                                                       4130771979
                                                                     Previous
Usage Counter: 15588                                   Record Size:  Record Offset:
                                                       72            11177672
Next Record Offset: 2951256                            Owner SID:    Group SID: S-
                                                       S-1-5-18      1-5-18
Number of SACL ACEs: 0                                 Number of
                                                       DACL ACEs: 1

          00000003(1)

Key Path: \ControlSet001
\Enum\USBSTOR\Disk&Ven_&Prod_&Rev_\10082264001041&0   Type: Key     Key Index: 0
\Properties\{83da6326-97a6-4088-9453-a1923f573b29}\00000003

                                                       Number of     Last Modified:
Number of Subkeys: 1                                   Values: 0     14/02/2012
                                                                     19:36:49
SK Record Offset: 3206048                              Key Hash:
                                                       1791585923
                                                                     Previous
Usage Counter: 15588                                   Record Size:  Record Offset:
```

Fig 9 P2 Commander showing Enum\USBSTOR\...

2.3.8 EnCase

Version: 7.01 Size: 1.81 Mbytes Modified time: Not available

Source: UCD (29/02/12)

Licence type: Closed source licensed

Platform: Windows

EnCase allows a registry hive to be mounted and viewed in tree format. EnScript, Guidance Software's proprietary scripting language facilitates user coding of scripts to automate tasks. Some scripts have been published that enumerate USB devices, notably Lance Mueller's (6) EnScript that collates USB device information from the system hive. His script is XP based and reports from Enum\USBSTOR, MountedDevices and DeviceClasses.

2.3.9 Registry Report

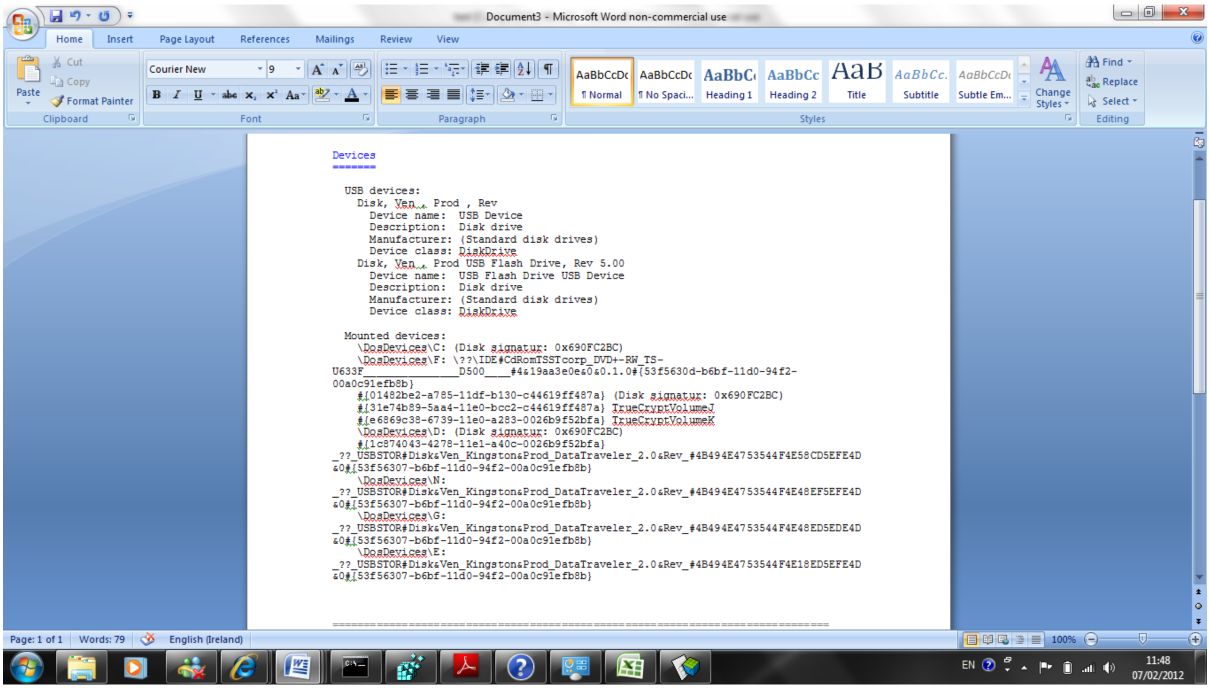
Version: 1.4.1 Size: 983 kbytes Modified time: 02/02/12 16:38

Source: http://www.gaijin.at/en/dlregreport.php (07/02/12)

Licence type: Closed source freeware

Platform: Windows 2000=>

RegistryReport runs against pre-extracted registry files. It presents selected keys in a readable format. No correlation work is done. No serial numbers, dates, users, guids etc. are reported



```
Devices
=======

  USB devices:
    Disk, Ven , Prod , Rev
      Device name:  USB Device
      Description:  Disk drive
      Manufacturer: (Standard disk drives)
      Device class: DiskDrive
    Disk, Ven , Prod USB Flash Drive, Rev 5.00
      Device name:  USB Flash Drive USB Device
      Description:  Disk drive
      Manufacturer: (Standard disk drives)
      Device class: DiskDrive

  Mounted devices:
    \DosDevices\C: (Disk signatur: 0x690FC2BC)
    \DosDevices\F: \??\IDE#CdRomTSSTcorp_DVD+-RW_TS-
U633F________________D500____#4&19aa3e0e&0&0.1.0#{53f5630d-b6bf-11d0-94f2-
00a0c91efb8b}
    #{01482be2-a785-11df-b130-c44619ff487a} (Disk signatur: 0x690FC2BC)
    #{31e74b89-5aa4-11e0-bcc2-c44619ff487a} TrueCryptVolumeJ
    #{e6869c38-6739-11e0-a283-0026b9f52bfa} TrueCryptVolumeK
    \DosDevices\D: (Disk signatur: 0x690FC2BC)
    #{1c874043-4278-11e1-a40c-0026b9f52bfa}
_??_USBSTOR#Disk&Ven_Kingston&Prod_DataTraveler_2.0&Rev_#4B494E4753544F4E58CD5EFE4D
&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}
    \DosDevices\N:
_??_USBSTOR#Disk&Ven_Kingston&Prod_DataTraveler_2.0&Rev_#4B494E4753544F4E48EF5EFE4D
&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}
    \DosDevices\G:
_??_USBSTOR#Disk&Ven_Kingston&Prod_DataTraveler_2.0&Rev_#4B494E4753544F4E48ED5EDE4D
&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}
    \DosDevices\E:
_??_USBSTOR#Disk&Ven_Kingston&Prod_DataTraveler_2.0&Rev_#4B494E4753544F4E18ED5EFE4D
&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}


================================================================================
```

Fig 10 RegistryReport excerpt

2.3.10 Comparison of existing tools

Tool	Comprehe-nsiveness	Level of Correlation	License type	Forensic Soundness / Accuracy
USBDeview	High	Low	Closed source freeware	Problems with dates& times
Registry Decoder	High	None	Open source GNU	Acquisition stage breaks ACPO/Daubert guidelines
Accessdata Registry Viewer	High	None	Closed Source Licensed	No Problems found
MiTec Windows Registry Recovery	Medium	None	Closed Source Freeware	No problems found
RegRipper	High	Low	Open Source GNU	No problems found
Xways	High	Low	Closed Source Licensed	No problems found
Registry Report	Medium	None	Closed Source freeware	No problems found
Paraben	Medium	None	Closed Source Licensed	No problems found
Encase	High	Medium (XP)	Closed Source Licensed	No problems found

Fig 11 Tool Comparison Chart

Regripper was viewed to be the most comprehensive, forensically sound open source tool. Having established that no tool appears to exist that performs cross-hive correlation of USB artefacts, it was decided to go ahead and define a specification for the proposed tools. To properly assess if significant benefits can be gained from the automated correlation of the registry artefacts it was decided to produce tools for USB, users and network information. System information was also included to make the tools a more complete set. The next step was to define the standards and specifications that the tools should adhere to. On completion of the tools some comparisons were made against Regripper, viewed as the gold standard.

2.4 Tool and process standards effecting the admissibility of digital evidence

Digital evidence is generally gathered in an attempt to support or refute a digital event. Regardless of whether evidence is collected for an internal corporate affair or a criminal trial, it is advisable to apply the highest standards for evidence collection and preservation from the outset. Failure to do so could make potential evidence inadmissible if a legal case ensued. Every jurisdiction will have minor variations but in general an investigator must ensure they have proper authority to seize evidence, they must follow any prescribed guidelines like ACPO or Daubert and ensure they have a chain of custody in place to prevent evidence tampering.

2.4.1 Authority

Evidence gathered without proper authority may be inadmissible in court. In Ireland evidence improperly gathered may be admitted at the Judge's discretion unless it has been obtained unconstitutionally in which case it must be excluded. Best practice is to ensure that the proper authority is obtained before collecting any evidence. This is typically granted by warrant, permission of the owner or by an Acceptable Usage Policy that has been acknowledged by the user.

2.4.2 ACPO Guidelines

The UK Association of Chief Police Officers provides four guidelines for the collection and handling of digital evidence. (9)

Principle 1: No action taken by law enforcement agencies or their agents should change data held on a computer or storage media which may subsequently be relied upon in court.
Principle 2: In circumstances where a person finds it necessary to access original data held on a computer or on storage media, that person must be competent to do so and be able to give evidence explaining the relevance and the implications of their actions.
Principle 3: An audit trail or other record of all processes applied to computer-based electronic evidence should be created and preserved. An independent third party should be able to examine those processes and achieve the same result.
Principle 4: The person in charge of the investigation (the case officer) has overall responsibility for ensuring that the law and these principles are adhered to.

2.4.3 Tool testing

Digital forensic tools can be used to gather evidence that may be used to support the guilt or innocence of an individual/entity. Therefore, a defendant is entitled to question the reliability of a tool that can influence the imposition of sanctions. Prior to designing any tool, consideration should be given to the Daubert test, open versus closed source tools and test data sets.

Originating in America, the **Daubert** test is a generic test applied to evaluate a process that extracts evidence. This test determines the reliability of evidence produced by applying four criteria to evaluate the methodology and techniques used**.**

- Testing: Can and has the procedure been tested?
- Error Rate: Is there a known error rate of the procedure?
- Publication: Has the procedure been published and subject to peer review?
- Acceptance: Is the procedure generally accepted in the relevant scientific community?

There is an argument that closed source commercial tools cannot meet these requirements thus the proliferation of open source forensic tools. Brian Carrier states that “open source tools may more clearly and comprehensively meet the guideline requirements than closed source tools.” (10)

One of the most important aspects of testing a tool is to provide a broad sample of test data. It has proved difficult to collect sample registry files from individuals as these can disclose personal details/passwords etc. It is also possible that a data set could be purposely or inadvertently designed that makes a tool appear more successful than is the case. One method of alleviating this problem is to include publicly available data as part of the test data. Simson Garfinkel has generated a corpus of digital images. When these images are used as test data they enhance the scientific value of any tests performed by their impartiality (11).

2.4.4 Acquisition and chain of custody

A forensically sound image has been historically collected from a Windows desktop system by acquiring a post-mortem raw disk image. This is verified as an exact copy by computing and matching the output of a hash algorithm applied to the original data set prior to collection and the raw image post collection. The ability to collect a bit mapped image has been complicated by developments in technology and is now not always possible. Data sets have become larger, mission critical systems cannot be shutdown for imaging and mobile devices may require invasive techniques to acquire evidence. It is often necessary to install an agent on a device or filter the data to a subset to facilitate collection. This is currently an area of debate in the forensic community. Erin E. Keneally has put forward a methodology for collecting partial digital images that proposes a structure for risk assessment that balances reasonability with completeness (12). Eoghan Casey (13) concludes ‘Provided the acquisition process preserves a complete and accurate representation of the original data, and its authenticity and integrity can be validated, it is generally considered forensically sound. Imposing a paradigm of “preserve everything but change nothing” is impractical and doing so can create undue doubt in the results of a digital evidence analysis, with questions that have no relation to the merits of the conclusions.’ If the decision is made to collect evidence selectively it is important that all relevant artefacts are identified.

In order to prove the integrity of digital evidence a chain of custody should be established that exhibits where a piece of evidence resides, who has come into contact with it and any processes applied to it since collection. Philip Turner has proposed a methodology for the collecting and handling of digital evidence in the form of a Digital Evidence Bag (DEB) (14). This process forms a parallel with tangible evidence handling and is suitable for application when dealing with selective imaging.

2.5 Defining the requirements – the selection of collection, preservation and presentation methods

2.5.1 Determine which version(s) of Windows to support

Microsoft Windows operating systems are installed on 92% of desktops PCs (15). This is broken down into XP 46.52%, Vista 8.44% and Windows 7 36.99%. In order to make the tool produced as useful as possible it was decided to support Windows XP, Vista and 7. According to Gartner Research in 2010 the installed base of PCs reached 1.4 billion (16). The combined data suggest that at the end of 2010 this specification would make the tools potentially useful for 1.288 billion windows PCs worldwide.

2.5.2 Platform, language and API selected

Altheide & Carvey (17) state that Linux is the most common open source forensic platform so a decision was made that any tool generated should be based on Linux.

Common languages used for writing forensic scripts include Perl, Python, Ruby and Bash. Amongst the many objectives, it was sought to make the tools portable, flexible, easy to understand and modify for a broad audience, so Bash was selected.

It was decided to use a tried and tested API for querying the registry. Two APIs were considered, Python-registry and Parse:Win32Registry. Python-Registry is an API developed by Willi Balenthin (18) for querying specific registry keys. Parse:Win32Registry, written by James Macfarlane (19) is a suite of Perl scripts that can amongst other things query, compare and report from Windows registries. It is used by Harlan Carvey for querying the registry in RegRipper. It proved accurate and reliable in tests and has been well used in the field. Parse:Win32Registry (Regdump.pl) won out as a tried and tested API due to its usage in RegRipper.

2.5.3 Common forensic data formats presented for analysis

There was a lack of statistical information about common forensic image formats. A review was performed on currently available forensic tools on *www.opensourceforensics.org* and *www.forensicswiki.org* to ascertain which formats are supported. Based on this data a decision was made to support registry file extraction from dd, EWF, AFF and direct acquisition from post mortem systems(via a forensically sound boot medium e.g. helix).

2.5.4 How the output report should be formatted

The primary focus of this project is to produce tools that correlate and interpret data to produce meaningful reports. Plain text-based output is considered to be adequate at this stage. The tools produce output to screen and also to log files. Two types of log files are maintained, an exact copy of what is sent to the screen and also an “expert log” recording the original source data prior to any interpretation. The first log file holds the processed information that can just be read or included in reports. The second log file holds the information extracted from a hive before processing, this is useful if the processing needs to be explained, in court for example.

2.6 Tool and process standards proposed

In order to provide the tools produced the best possibility of producing admissible, usable evidence the project aspired to meet the following criteria:

- Ensure authorisation is obtained prior to any data collection
- Follow ACPO and Daubert guidelines
- Document the testing process
- Automate the collection of selective identified artefacts
- Data should be collected with a peer accepted product
- Use some form of “Digital Evidence Bag” to maintain data integrity and provide a chain of custody
- Some test data should be from an impartial publically available source

- Any tools produced will be open source tools
- Windows XP, Vista and 7 will be supported
- Tools will be scripted in Bash under Linux
- dd, EWF, AFF and direct system image formats will be supported for registry file extraction
- Require minimal user input

3 Problem Statement

Manual correlation and interpretation of data can be repetitive, slow and prone to error. Windows registry analysis still requires a level of manual correlation and interpretation, so the question posed is. Is it possible to improve the process of Windows registry analysis by providing greater automation than is currently available, in a forensically sound manner?

4 Adopted Approach

The research in Section 2 shows that there is scope for increasing the level of automation by replacing some layers of manual correlation and interpretation of data with an automated solution. However, the strict criteria outlined in section 2.6 must be adhered to, in order to produce reports that can be presented in court.

The aim was to produce a suite of tools that would identify, collect, report, interpret and correlate Windows registry and related artefacts. Artefacts were grouped into four areas, USB, network, user and system related. Five tools were created, one for each area and one for the collection process. Sample registry hives were gathered for Windows XP, Vista and 7 to be used in the design, coding and testing phases.

5 Scripts, Experiments and Observations

Five sections follow detailing each tool, its output and usage where appropriate. As part of the testing process some experiments and observations of interest were made and these are detailed at the end of each section.

The first tool produced was the data extraction script to assist with the acquisition of sample test data. The extraction script identifies hashes and zips the required registry and associated files. This script can be run against any mounted volume, full instructions are provided for mounting common forensic image formats as Linux volumes. A full log is maintained of the activity.

The other four tools are used for the registry analysis. They possess a built-in help function for ease of use and also maintain detailed logs. These include an “expert” log which stores the original data before any interpretation or correlation is performed. An examiner does not need to memorise which hive file a particular key or value is stored in. All the hive files for analysis can be stored in a single directory and the tools know which hive to analyse. Multiple user hives can be placed in a single directory as long as they have different names. Details of manipulations performed on the extracted hive data are recorded separately in the Appendices.

5.1 Method for evidence collection and preservation

Before any tool was generated test data had to be acquired. Various methods of extracting the required data from the supported digital evidence formats were evaluated. This process was not as straightforward as expected. Complicated setup procedures were considered a potential barrier to the use of the tools produced, so it was decided to document this process as part of the project. This also resulted in the production of some scripts that would automate certain aspects of the procedure.

5.1.1 Acquisition of test data

In order to comprehensively test the tools generated it was decided to source a variety of registry files from the versions of Windows supported (XP, Vista, 7). As a registry contains traces of a user’s activity it is understandable that users were reluctant to give access to files that could reveal private information. Despite this a few subjects did agree to donate real registries. Other potential sources were; fresh system installs with simulated use (there is a question as to how ‘real’ that data is), image archives from the public domain and images generated by UCD for assignment purposes.

Once the registry sources had been identified the next challenge was to extract the required data. This process was to be repeated to acquire test data and any examiner choosing to use the reporting tools will also face the same challenge. For ease of use it was decided to procedurise and as far as possible automate this process. A method was developed to collect registry files from each of the formats supported (dd, AFF, EWF and direct system) paying due attention to the soundness of the collection and preservation of the files. Based loosely on Garfinkel's AFF4 (20) research it was decided to zip’n’hash the collected artefacts for storage and proof of integrity. The process of extracting registry files from a live system, dd, AFF and EWF is documented in the next section, so if you are already familiar with this it is suggested you skip to Section 5.1.4

5.1.2 Direct system selected file acquisition

The Windows architecture was designed to protect the registry files and does not simply allow you to take a copy from a live system. Even if the files were successfully copied from a live system it could result in an inconsistent version of the registry. Undoubtedly taking a forensic bit image followed by extracting artefacts from that image is the preferable method for collecting registry files as this method complies with ACPO and Daubert principles. However in reality situations can arise where this is either impractical or impossible, for example forensic triage or when dealing with very large disks or cloud storage. It was decided to support partial imaging from a live system in addition to file extraction from existing forensic images. In order to collect a consistent copy of the registry from a system it is necessary to boot the system from a source other than the partition in question and take a post mortem copy of the files. Helix was chosen to perform this task and in order to simplify the process, a Helix boot USB was created.

5.1.2.1 Creation of a multi-partitioned Helix boot USB:

- A low level format was performed on the 32GB USB pen drive to zero any existing data.
- The single partition was deleted and two new primary partitions were created on the drive. The second partition was set to active. This was done via Windows with EaseUS partition manager v9.10 (home edition free). Windows only sees the first primary partition on a pen drive so the first partition was assigned for evidence.
- To access the 2nd partition from Windows the first partition should be *temporarily* changed from primary to logical (with EaseUS), making the 2nd partition the first primary partition and accessible from Windows.
- UNetBootin-windows-563 was used to load the Helix2008R1.iso onto the second partition.
- The first Partition was reverted to primary.

Evidence (Primary Partition – non active) **26GB –NTFS**	Boot (Primary Partition – Active) **4GB – FAT32**

Fig 12 USB partition layout

- The script extractreg.sh was placed into /scripts directory on the NTFS partition.

5.1.2.2 Collection of evidence from a system with the helix usb stick

Set the system to boot from Helix USB via setup or boot options, typically F2 or F12. The MBR on the USB points to the second partition as active so it will boot into Helix. Once booted the other (first) partition can then be mounted via terminal by:

- identifying the device with $ *sudo fdisk –l* (on the test system it was /dev/sdb1)
- making a mountpoint $ *sudo mkdir /mnt/usbntfs*
- mounting $ *sudo mount –t ntfs-3g /dev/sdb1 /mnt/usbntfs*.
- Changing to the scripts dir $ *sudo cd /mnt/usbntfs/scripts*
- Run extractreg.sh $ ./extractreg.sh -o /mnt/usbntfs -i sda3 -t system -c mycase
- The script will display an fdisk of the system and autosuggest which partition to extract the evidence files from e.g. /dev/sda2. If this is not the required device the correct device should be entered here.
- Section 5.1.4 contains further details on the extractreg.sh script.

5.1.3 Instructions for bitmap image mounting and registry extraction

Files can also be collected directly from a dd, AFF or EWF image. This is achieved by running the extractreg.sh script against a mounted image. With a raw dd image a simple mount can be performed. To access compressed forensic images a few methods were reviewed. After some experimentation the methods outlined below proved to be consistent and effective. For both AFF and EWF images the forensic container is mounted via FUSE (Filesystem in Userspace). This presents the raw image which can then be mounted using the usual raw image mounting process. When the image has been successfully mounted the extractreg.sh is applied to extract the registry and associated files.

5.1.3.1 dd image

If the image has been supplied as a split image it must be concatenated it into a single file prior to mounting. Mounting is performed on volumes or partitions, which do not typically correspond to a whole disk. In order to mount a partition from a disk image the offset to the partition starting point must be calculated. The mmls utility from Carriers Sleuthkit (21) was used to calculate this.

```
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

     Slot     Start        End          Length       Description
00:   Meta      0000000000    0000000000    0000000001    Primary Table
(#0)
01:  -----    0000000000   0000000062   0000000063   Unallocated
02:  00:00    0000000063   0019502909   0019502847   NTFS (0x07)
03:  -----    0019502910   0020010815   0000507906   Unallocated
```

Fig 13 Sample mmls output - areas of interest are highlighted

First time only

```
$ sudo mkdir /mnt/ex_volume
```

For usage

```
$ mmls image.dd                    => (512*63=32256, sectorsize *starting sector=offset)
 $ sudo mount –t ntfs –o loop,ro,show_sys_files,offset=32256 image.dd /mnt/ex_volume
$ ./extractreg.sh -o /mydir -i /mnt/ex_volume -t image -c mycase
```

To umount

```
$ sudo umount /mnt/ex_volume
```

5.1.3.2 AFF image

The following prerequisites must be installed prior to using affuse, zlib development library, Open SSL development and fuse library.

First time only

```
$ sudo apt-get install libfuse-dev libssl-dev zlib-dev
$ tar –zxvf afflib-3.6.15.tar.gz (downloaded from afflib.org-02/02/12)
$ ./configure
$ make
$ sudo make install
$ sudo mkdir /mnt/aff
$ sudo mkdir /mnt/ex_volume
```

For usage

```
$ sudo affuse image.aff /mnt/aff
$ sudo ls /mnt/aff                          => image.aff.raw
$ sudo mmls /mnt/aff/image.aff.raw          => (512*2048=1048576)
$ sudo mount –t ntfs –o ro,loop,show_sys_files,offset=1048576 /mnt/aff/image.aff.raw
/mnt/ex_volume
$ ./extractreg.sh -o /mydir -i /mnt/ex_volume -t image -c mycase
```

To umount

```
$ sudo umount /mnt/ex_volume
$ sudo umount /mnt/aff
```

5.1.3.3 EWF image

Python-fuse must be installed prior to using the ewf mounter.

First time only

```
$ sudo apt-get install python-fuse
Download mount_ewf-20090113.py (http://sourceforge.net/projects/libewf/files-02/02/12)
$ mv mount_ewf-20090113.py /usr/local/bin/mount_ewf.py
$ sudo mkdir /mnt/ewf
$ sudo mkdir /mnt/ex_volume
```

For usage

```
$ sudo mount_ewf.py myimage.e01 /mnt/ewf
$ sudo ls /mnt/ewf                          =>myimage
$ sudo mmls /mnt/ewf/4Dell                  =>(512*63=32256)
$ sudo mount –t ntfs –o ro,loop,show_sys_files,offset=32256 /mnt/ewf/myimage /mnt/ex_volume
$ ./extractreg.sh -o /mydir -i /mnt/ex_volume -t image -c mycase
```

To umount

```
$ sudo umount /mnt/ex_volume
$ sudo umount /mnt/ewf
```

5.1.4 Script for extraction and preservation from a mounted volume

To automate the collection of registry files from a Linux mounted Windows volume, the script "*extractreg.sh"* was produced. Some consideration was given to whether files should be located via their 'magic' signature or by their location. To move the default location of the registry files would require technical knowledge and would probably be a deliberate obfuscation on a system. It was considered an unlikely event that would probably not be done in isolation so automated extraction may not be appropriate on this type of system. A decision was made that the scripts would search for files in their default location. The script uses the find command to locate the required log, registry and .lnk files, namely system, software, sam, security, ntuser.dat(s), setupapi(s) and .lnks. On the test systems setupapi.dev.20110205_143756.log was also found.

Operating system	Path	Filenames
XP	\WINDOWS\system32\config \Documentsand Settings\username \WINDOWS \	System, software, SAM, SECURITY NTUSER.DAT Setupapi.log .lnk
Vista	\Windows\System32\config \Users\username \Windows\inf \	SYSTEM or system, SOFTWARE or software, SAM SECURITY NTUSER.DAT or ntuser.dat Setupapi.dev.log .lnk
Windows 7	\Windows\System32\config \Users\username \Windows\inf \	SYSTEM, SOFTWARE, SAM, SECURITY NTUSER.DAT Setupapi.dev.log .lnk

Fig 14 Windows registry files recorded from test systems by operating system

Prior to copying the message digest of these files is calculated and stored in a file called md5eachfileprecopy.log stored in the log directory. These files are then zipped (which records their original location), re-hashed in the zipped format and saved to the specified output directory. The script has a help function which details the options below to assist with correct usage.

Option	function
-o	Output location eg. ~/myfiles
-i	Input volume location eg. /mnt/ex_volume
-t	Type of volume image or system
-c	Casename
-h	help

Fig 15 extractreg.sh options

5.1.5 Storage format

The storage format see Fig 16 has been loosely based on a simplified version of the AFF4 (22) methodology. The script identifies and hashes the required files which are then immediately zipped, stored and re-hashed. The hash is stored and also reported on screen. A directory structure detailed below is then created in the requested output location. The naming convention used for this directory is, case name, followed by the current date and time e.g. mycaseSun.Jan.22.18.27.23.UTC.2012. Note the time recorded is the current time on the system running the script. This is particularly valuable if the extraction is being done directly from a system via the Helix boot stick.

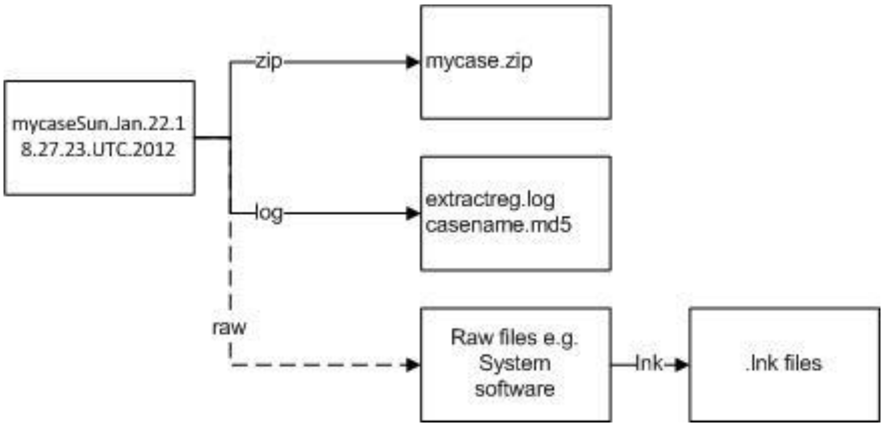


Fig 16 Directory format – raw files are unzipped prior to usage

5.1.6 Unzipping the extracted files – getraw.sh

As detailed in Fig 14 (Page 29) separate versions of Windows store the registry files differently. For example the user specific hive NTUSER.DAT can be upper or lower case and stored in different locations. In order to provide some consistency for further script processing it was decided to provide a script that would check the hash value and unzip the files with consistent naming conventions. User hives will be given an extension indicating their original location to differentiate and identify them. E.g.

C:\Documents and Settings\terry\NTUSER.DAT => NTUSER.DAT.terry
C:\Users\Student\ntuser.dat => NTUSER.DAT.Student

Fig 17 unzipped filename translation

5.1.7 Table of sample data sets

Eighteen sets of test data were collected across the supported Windows platforms from a variety of evidence formats and sources. Four of the hive sets were from publically available sources.

#	ID	Windows O/S	Image Type	Source
1	xpnist	XP	EWF	http://www.cfreds.nist.gov/Hacking 4Dell Lattitide CPi.E01 & E02
2	xpfkb	XP	EWF	http://www.forensickb.com/2008/01/forensic-practical.html (WinXP.E01 & .E02)
3	xpryan	XP	dd	UCD (Mr Ryan Scenario)
4	xpbadguy	XP	Dd	UCD (bad guy scenario)
5	xpj	XP	Extracted	UCD scenario
6	xphc	XP	Extracted	Windows Forensic Analysis accompanying CD
7	Xpknap2a	XP	E01	UCD Interpol scenario 2a
8	xpkorean	XP	Hives	A set off Korean language hives
9	vistapc	Vista	System	3 year old heavily used home based system
10	vistal	Vista	System	4 years old, well used
11	vistaaff	Vista	AFF	http://digitalcorpora.org/corpora/scenarios/m57-patents-scenario (Terry's second image)
12	win7laptop	7	System	18 month old home system
13	win7j	7	System	3 years old, well used
14	win7t	7	System	3 years old, well used
15	win7d	7	System	2 years old, well used
16	Win7knap1	7	E01	UCD Interpol scenario system 1
17	Win7Knap2b	7	E01	UCD Interpol scenario system 2b
18	Win7knap3	7	E01	UCD Interpol scenario system 3

Fig 18 Test data table

Now that the specifications have been set and the test data has been collected it is possible to start scripting the actual analysis tools.

5.2 USBDevices script

5.2.1 Goals

- To identify, correlate, interpret and report as many artefacts as possible relating to USB storage devices recorded in the registry (or logs).
- To understand the process of how the registry records USB activity.
- To document and log the process of data interpretation and manipulation to provide transparency. (Full details for this script are provided in Appendix A)

5.2.2 Sample output

Five examples of output produced from usbdevices.sh are provided from the sample registry hives. The source of each field is related to the data underlined in section 5.2.3.1.

```
Serial num/iid : 000A270018B6253F&0
Long name      : Disk&Ven_Apple&Prod_iPod&Rev_1.62
Friendly name  : Apple iPod USB
Last in dc 307 : 2010-02-10 16:34:35 (UTC)
Last in dc 30d : 2011-09-21 09:41:45 (UTC)
Last in enumusb: 2011-09-21 09:20:12 (UTC)
Last in dc a5  : 2010-02-10 16:34:34 (UTC)
Last test time :
Vendor ID       : 05AC (Apple, Inc.)
Product ID     : 1260
Drive\Volume   : LAURA'S IPO (G:) 2008-08-01 11:12:43 (UTC)
               : LAURA'S IPO (G:) 2009-06-28 19:02:47 (UTC)
Volume GUID    : {fb5d8bd9-5ceb-11dd-b48f-001d09806873}
Volume s/n     :
First install  : 2008/08/01 12:12:21.831 (Local Time)
Username       : Jacky 2010-02-10 20:24:37 (UTC)
```

Fig 19 USBdevices sample output 1

```
Serial num/iid : 099300178283&0
Long name      : Disk&Ven_ChipsBnk&Prod_Flash_Disk&Rev_4.00
Friendly name  : ChipsBnk Flash Disk USB
Last in dc 307 : 2011-05-10 10:34:03 (UTC)
Last in dc 30d : 2011-09-21 09:41:48 (UTC)
Last in enumusb: 2011-09-21 09:20:12 (UTC)
Last in dc a5  : 2011-05-10 10:34:03 (UTC)
Last test time : 2011-04-15 19:21:04 (UTC) Fri Apr 15 20:21:04 IST 2011
Vendor ID       : 0204 (Chipsbank Microelectronics Co., Ltd)
Product ID     : 6025
Drive\Volume   : I: (mountdev)
               : Removable Disk (I:) 2011-04-15 19:16:31 (UTC)
               : JACKY&KARIN (I:) 2011-05-10 10:55:01 (UTC)
Volume GUID    : {d2ab32d5-6747-11e0-8eda-001d09806873}
Volume s/n     : 2019133461 (0x78598815)
First install  : 2011/04/15 20:16:10.772 (Local Time)
Username       : Jacky 2011-05-10 10:34:05 (UTC)
```

Fig 20 USBDevices sample output 2

```
Serial num/iid : AA77071100019996&0
Long name      : Disk&Ven_LG&Prod_USB_Drive&Rev_1100
Friendly name  : LG USB Drive USB
Last in dc 307 : 2010-03-01 18:12:59 (UTC)
Last in dc 30d : 2011-09-21 09:42:00 (UTC)
Last in enumusb: 2011-09-21 09:20:13 (UTC)
Last in dc a5  : 2010-03-01 18:12:59 (UTC)
Last test time : 2009-07-19 20:35:59 (UTC) Sun Jul 19 21:35:59 IST 2009
Vendor ID      :  090C  (Silicon  Motion,  Inc.  -  Taiwan  (formerly  Feiya
Technology Corp.))
Product ID     : 1000
Drive\Volume   : USB Drive (H:) 2008-04-07 19:52:58 (UTC)
               : USB Drive (I:) 2010-03-01 20:35:46 (UTC)
Volume GUID    : {7b7aa3aa-02b5-11dd-9d1c-001d09806873}
Volume s/n     : 661216930 (0x27695EA2)
First install  : 2008/04/07 20:52:38.144 (Local Time)
Username       : Jacky 2010-03-01 20:35:44 (UTC)
```

Fig 21 USBDevices sample output 3

```
Serial num/iid : AA7ED7500C30&0
Long name      : Disk&Ven_LaCie&Prod_SAFE_drive&Rev_
Friendly name  : LaCie SAFE drive USB
Last in dc 307 : 2011-09-13 08:50:31 (UTC)
Last in dc 30d : 2011-09-21 09:41:57 (UTC)
Last in enumusb: 2011-09-21 09:20:12 (UTC)
Last in dc a5  : 2011-09-13 08:44:24 (UTC)
Last test time : 2008-04-05 10:23:54 (UTC)
Vendor ID       : 0451 (Texas Instruments, Inc.)
Product ID     : 6250
Drive\Volume   : SAFE DRIVE (G:) 2008-04-05 10:24:10 (UTC)
               : SAFE (G:) 2011-09-13 09:34:39 (UTC)
Volume GUID    : {7b7aa252-02b5-11dd-9d1c-001d09806873}
Volume s/n     : 14747223 (0xE10657)
First install  : 2008/04/05 11:23:47.682 (Local Time)
Username       : Jacky 2011-09-13 09:34:03 (UTC)
```

Fig 22 USBDevices sample output 4

```
Serial num/iid    : 4B494E4753544F4E08EF1EFE4D&0
Long name         : Disk&Ven_Kingston&Prod_DataTraveler_2.0&Rev_
Friendly name     : Kingston DataTraveler 2.0 USB
Last in dc 307    : 2012-02-17 14:39:15 (UTC)
Last in dc 30d    : 2012-02-17 14:39:17 (UTC)
Last in enumusb   : 2012-02-29 19:22:13 (UTC) Time may not be device specific
Last in dc a5     : 2012-02-17 14:39:11 (UTC)
Last test time    : 2012-02-17 14:39:17 (UTC)
Vendor ID         : 0951
Product ID        : 162F
Drive\Volume      : EVIDENCE 2012-02-17 14:39:19 (UTC)
Volume GUID       : {747e63e5-565e-11e1-b0d8-c44619ff487a}
Volume s/n        : 625134160 (0x2542CA50)
.lnk files        : bootsqm.lnk
                  : EVIDENCE (E).lnk
                  : extractreg (2).lnk
                  : extractreg.lnk
                  : Investigative Database Report.lnk
                  : lookatthisone.lnk
                  : mine.lnk
First install     : 2012/02/17 14:39:14.439 (Local Time)
Username          : Jacky 2012-02-17 14:39:18 (UTC)
```

Fig 23 USBDevices sample output 5

5.2.3 Artefact identification

The two best sources identified for artefacts relating to USB devices, were Harlan Carvey's Windows Registry Forensics (23) and Rob Lee's USB guides (24). Using these as a starting point the registry hives from the sample data set were investigated in an attempt to highlight all the pertinent registry USB artefacts. Identifiable artefacts can be traced back to a specific device either by volume guid, device serial number, parent id prefix or volume serial number. To ensure that all the relevant keys were identified, searches were performed in the registry for these identifiers in decimal, hexadecimal and unicode in big and little endian for sample known devices. Once identified the keys were inspected to gather any potentially interesting data from their values and timestamps. This procedure was repeated for several devices using XP, Vista and Windows 7 hives.

In addition to the registry, related data is stored in other areas and where available is reported. If the files are collected using the extractreg.sh script, any available setupapi log files or .lnk files from the image/system, will be gathered automatically. The setupapi.log is used to determine the first installed time of a device. ".lnk" files can be associated with USB devices by the volume serial number. From Vista onwards the volume serial number is stored in the readyboost key - EMDMgmt (detailed in section 5.2.3.1). This volume serial number is stored in little endian hexadecimal format in any associated .lnk files. For example if the volume serial number in Readyboost is 625134160 => 0x2542CA50 => 50CA4225 (25).

Section 5.2.3.1 details a sample device trace through a Windows 7 registry for USB data and highlights any forensic values. The connections between the highlighted keys were established and detailed in a diagram in section 5.2.4.

5.2.3.1 Registry device trace and interpretation

The findings from a trace of a sample device used, with serial number 200602668009F6B085B3 and volume guid {55782b83-ca94-11e0-b929-0026b9f52bfa} are detailed below. The data extracted from each key/value is named and <u>underlined</u>. These names correspond with the headings used for the reports as seen in the sample output in section 5.2.2.

A search for the serial number 200602668009F6B085B3 found the following registry entries:

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Enum\USBSTOR\Disk&Ven_SanDisk&Prod_Cruzer_Edge&Rev_1.10\200602668009F6B085B3&0
Devices are grouped by vendor/product and revision and are uniquely identified by a device instance identifier/<u>serial number</u>. Other values of importance stored here are the device <u>long name</u>, <u>friendly name</u> and if XP the <u>ParentIdPrefix.</u> This is a good starting point to identify every device that has been connected to a system.

HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\EMDMgmt_??_USBSTOR#Disk&Ven_SanDisk&Prod_Cruzer_Edge&Rev_1.10#200602668009F6B085B3&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}_3368049607 (20)
This Vista or later key relates to the Windows Readyboost service where USB devices can be used as cache on systems without SSD disks installed. When a new USB device is installed a benchmark is performed on that device to determine its suitability as a readyboost drive. The result of the test and the <u>last tested time</u> are stored here. This <u>timestamp</u> shows when a device was last considered for testing and if an actual test took place. This timestamp in Fig 20 shows the device was connected for at least 4 minutes and 56 seconds. In Fig 21 the timestamp shows a connection not recorded elsewhere. An 'unsuitable' device may only be tested on first insertion. The key and test times are reported as the absence of a test time in itself is information; it may

show for example that the device didn’t meet the minimum free space requirements to do a benchmark (26). The last number in the key 3368049607 is a decimal value for the volume serial number. This value is also displayed in hexadecimal to aid searching for the volume serial number in .lnk files stored on the system.

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows Portable Devices\Devices\ WPDBUSENUMROOT#UMB#2&37C186B&1&STORAGE#VOLUME#_??_USBSTOR#DISK&VEN_SANDISK&PROD_ CRUZER_EDGE&REV_1.10#200602668009F6B085B3&0#
A timestamped key present in Windows Vista or later with a value FriendlyName that may contain drive letters or volume names. All entries are timestamped and there may be multiple entries per device here.

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\##?#USBSTOR#Disk&Ven_SanDisk&Prod_Cruzer_Edge&Rev_1.10#200602668009F6B085B3&0 #{53f56307-b6bf-11d0-94f2-00a0c91efb8b}
The device interface class {53f56307… is used to enumerate hard disk storage devices (27). The timestamp here records the first time a device was inserted after the last reboot (24). i.e. subsequent insertions are not recorded in this key until the next reboot, this is very apparent if a system is typically hibernated rather than shutdown. Fig 21 shows an example of this where the username mountpoints2 timestamp shows the actual last insertion is later than the {53f56307… value.

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\DeviceClasses\{a5dcbf10-6530-11d2-901f-00c04fb951ed}\##?#USB#VID_0781&PID_556B#200602668009F6B085B3#{a5dcbf10-6530-11d2-901f-00c04fb951ed}
The device interface class {a5dcbf10… is used to enumerate raw usb devices (27). The timestamp here records the first time a device was inserted after the last reboot (24). i.e. subsequent insertions are not recorded in this key until the next reboot, this is very apparent if a system is typically hibernated rather than shutdown. Fig 21 shows an example of this where the username mountpoints2 timestamp showing the actual last insertion is later than this value. Fig 22 shows a timestamp more than 6 minutes earlier than the {…307..} guid traditionally used. It is worthwhile showing both timestamps as this could give show a minimum length of usage.

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Enum\STORAGE\Volume\ _??_USBSTOR#Disk&Ven_SanDisk&Prod_Cruzer_Edge&Rev_1.10#200602668009F6B085B3&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}
Present but not used for this report as this key contains no data/timestamps not repeated elsewhere.

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Enum\USB\VID_0781&PID_556B\ 200602668009F6B085B3
This key is quoted by Rob Lee to hold the first time that a key was connected after reboot. i.e. If a usb device that has not been inserted since the last system reboot (not hibernation) is inserted at 9:10:13 and it is subsequently removed and then re-inserted at 11:30:15, the timestamp will remain at 9:10:13. According to Rob Lee this should not update until after the next insertion post reboot. In experiments performed this was not found to be always the case. Of the sample hives extracted and used for testing several were found to have the same/similar “Enum tree” timestamps across all usb devices. An example of this can be seen in the samples in section 5.2.2 where all three devices have the same/similar timestamps for this value. This value is still reported as if the timestamp varies by more than a few seconds from the Enum root timestamp Rob Lees findings are correct. However caution should be used when quoting this timestamp as it may not be device specific. The values extracted from this key are Last in enumusb, the four digit vendor id and product id numbers. The name of the vendor is identified by comparing the vendor id number to the vendor id database moderated on linux-usb.org.

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Enum\WpdBusEnumRoot\UMB\2&37c186b&1&STORAGE#VOLUME#_??_USBSTOR#DISK&VEN_SANDISK&PROD_CRUZER_EDGE&REV_1.10#200602668009F6B085B3&0#
Present but not used for this report as this key contains no data/timestamps not repeated elsewhere.

HKLM\SYSTEM\MountedDevices   Value: \DosDevices\E:
The serial number (or ParentIDPrefix if XP) can be found in HKLM\SYSTEM\MountedDevices as a Unicode value, if the device was last device, assigned to a particular drive letter. MountedDevices can also be used to identify the last device assigned to a drive letter. Serial insertion and extraction of USB devices will typically result in all devices being assigned the same drive letter. Drive letters extracted from the mounted device key will have (mountdev) appended from the script to denote their source.

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\DeviceClasses\{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}\##?#STORAGE#VOLUME#_??_USBSTOR#DISK&VEN_&PROD_&REV_#200602668009F6B085B3&0#{53F56307-B6BF-11D0-94F2-00A0C91EFB8B}#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
The device interface class {53f5630d… is used to enumerate volumes (27). The timestamp here records the first time a device was inserted after the last reboot.(Proved by experimentation) i.e. subsequent insertions are not recorded in this key until the next reboot, this is very apparent if a system is typically hibernated rather than shutdown. However this key may also get updated with a time that is not specific to the device so caution should be used when gauging its validity. Fig 21 shows an example of where three separate devices hold the same/similar times. This appears to be a similar phenomena to the enum global event (see section 5.2.5.3 for more details).

The Search for volume guid of {55782b83-ca94-11e0-b929-0026b9f52bfa} in the registry found:

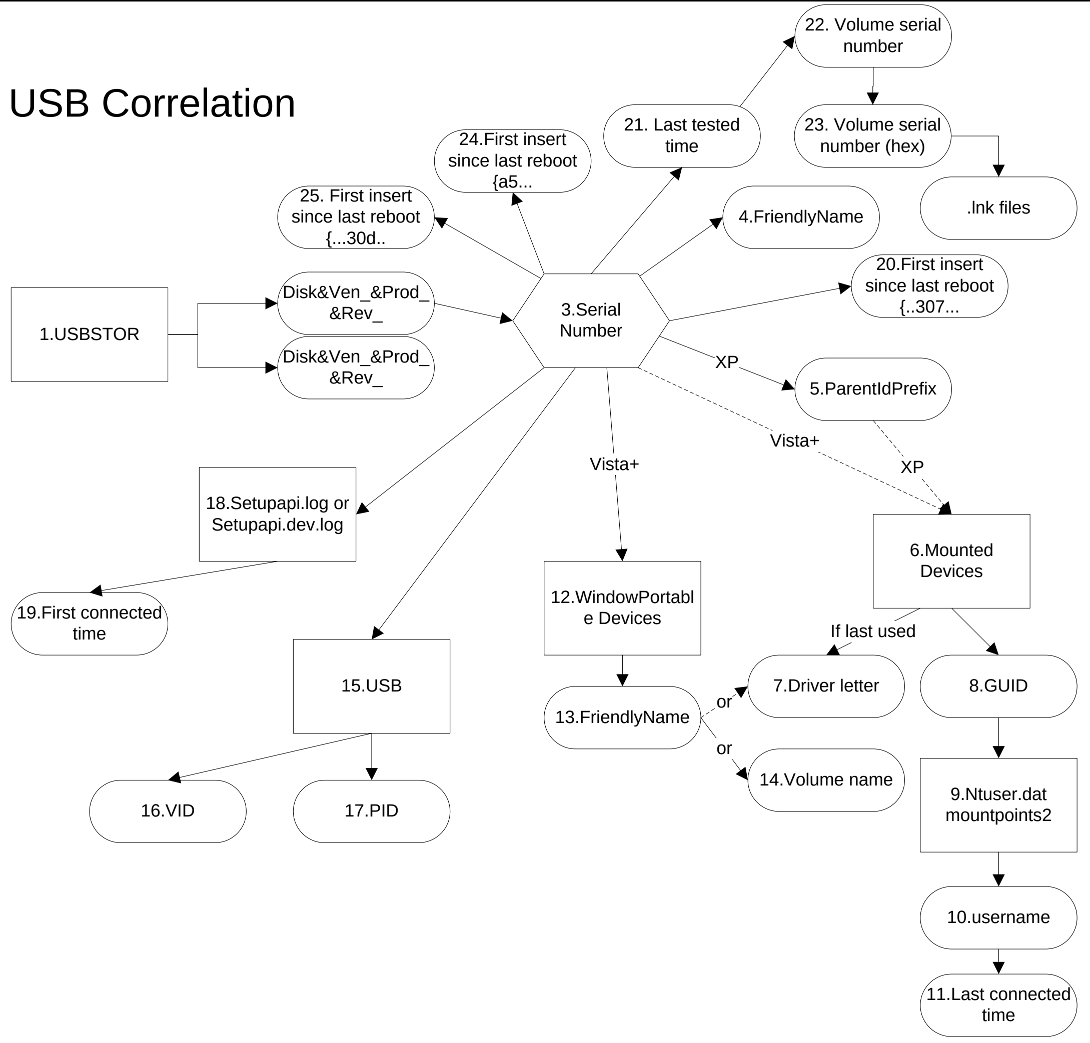
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{55782b83-ca94-11e0-b929-0026b9f52bfa}
The MountPoints2 key in all available user hives (NTUSER.DAT) can be inspected to see if any subkeys with known volume guids are present. If a guid is present the timestamp from this key is a record of the time a device was last inserted into the system while that user was logged in. i.e if the device was inserted at 5:02:03 and remained plugged in for some time, the key would be timestamped 5:02:03, not the time immediately prior to extraction. It should be noted that all user logged in when a device is connected will have their MountPoints2 timestamp updated. (See section 5.2.5.2 for more details)

HKLM\SYSTEM\MountedDevices   Value: \??\Volume{55782b83-ca94-11e0-b929-0026b9f52bfa}
The values in the MountedDevices key hold the volume guid. These can be matched with the device instance id or parent id prefix (XP), to relate the volume guid to a device.

Other devices were seen to have guid entries under other DeviceClasses keys. These were not used in the usbdevices script as they were never found to hold unique information.

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\DeviceClasses\{10497b1b-ba51-44e5-8318-a65c837b6661}\##?#WpdBusEnumRoot#UMB#2&37c186b&1&STORAGE#VOLUME #_??_USBSTOR#DISK&VEN_&PROD_&REV_#10082264001041&0##{10497b1b-ba51-44e5-8318-a65c837b6661}
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\DeviceClasses\{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}\##?#STORAGE#VOLUME#_??_USBSTOR#DISK&VEN_&PROD_&REV_#10082264001041&0#{53F56307-B6BF-11D0-94F2-00A0C91EFB8B}#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\DeviceClasses\{6ac27878-a6fa-4155-ba85-f98f491d4f33}\##?#WpdBusEnumRoot#UMB#2&37c186b&1&STORAGE#VOLUME#_??_USBSTOR#DISK&VEN_&PROD_&REV_#10082264001041&0##{6ac27878-a6fa-4155-ba85-f98f491d4f33}
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\DeviceClasses\{f33fdc04-d1ac-4e8e-9a30-19bbd4b108ae}\##?#WpdBusEnumRoot#UMB#2&37c186b&1&STORAGE#VOLUME#_??_USBSTOR#DISK&VEN_&PROD_&REV_#10082264001041&0##{f33fdc04-d1ac-4e8e-9a30-19bbd4b108ae}

5.2.4 USB device Connections diagram



1.HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR
2.HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR\Disk&Ven_&Prod_&Rev
3.HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR\Disk&Ven_&Prod_&Rev\eg999999999
4.HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR\Disk&Ven_&Prod_&Rev\eg999999999\FriendlyName
5.HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR\Disk&Ven_&Prod_&Rev\eg999999999\ParentIdPrefix (XP)
6.HKLM\SYSTEM\MountedDevices\DosDevices\egF:\search value for unicode of parentidprefix(XP) or serial number(V+)
7.Drive letter assignations if any from 6. and 13.
8.GUID assigned from search at 6.
9.HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2 – search for GUID
10.Confirmed if GUID is present at 9. in Ntuser.dat
11.HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\matched GUID – key timestamp
12.HKLM\Software\Microsoft\Windows Portable Devices\Devices\WPDBUSetc.. - search keys for s/n
13.HKLM\Software\Microsoft\Windows Portable Devices\Devices\matched s/n\FriendlyName – letter or volume?
14.HKLM\Software\Microsoft\Windows Portable Devices\Devices\matched s/n\FriendlyName – if volume
15.HKLM\SYSTEM\CurrentControlSet\Enum\USB\egVID_9999&PID_9999\ -search for serial numbers
16.HKLM\SYSTEM\CurrentControlSet\Enum\USB\egVID_***here***&PID_9999\ -matched serial number – set to vendor id
17.HKLM\SYSTEM\CurrentControlSet\Enum\USB\egVID_9999&PID_***here***\ -matched serial number – set to product id
18.Search for serial number in c:\windows\setupapi.log(XP) or c:\windows\inf\setupapi.dev.log(Vista+)
19.Timestamp from 18.
20. HKLM\SYSTEM\CurrentControlSet\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91rfb8b}\search for serial #
21. HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\EMDMgmt_??_USBSTOR...# -matched serial number
22. Volume serial number from 21.
23. Hexidecimal serial number from 22.
24. HKLM\SYSTEM\CurrentControlSet\Control\DeviceClasses\{a5dcbf10-6530-11d2-901f-00c04fb951ed}\search for serial #
25. HKLM\SYSTEM\CurrentControlSet\Control\DeviceClasses\{53f5630d-b6bf-11d0-94f2-00a0c91rfb8b}\search for serial #

Fig 24 USB artefact connections diagram

5.2.5 Experiments and observations

5.2.5.1 Order of key population on insertion of a new device

Samples of output from the script usbdevices.sh, showing devices that were inserted into a system only once. It shows the order that the keys are initially populated by their timestamps.

```
Serial num/iid : 4B494E4753544F4E58CF5EDE4D&0
Long name      : Disk&Ven_Kingston&Prod_DataTraveler_2.0&Rev_
Friendly name  : Kingston DataTraveler 2.0 USB
Last in dc 307 : 2012-02-13 09:35:44 (UTC)                  4th
Last in dc 30d : 2012-02-13 09:35:48 (UTC)                  5th
Last in enumusb: 2012-02-13 09:35:41 (UTC)                  2nd/3rd
Last in dc a5  : 2012-02-13 09:35:38 (UTC)                  1st
Last test time : 2012-02-13 09:35:49 (UTC)                  6th/7th
Vendor ID       : 0951 (Kingston Technology)
Product ID     : 162F
Drive\Volume   : F: (mountdev)
               : EVIDENCE 2012-02-13 09:36:03 (UTC)         8th
Volume GUID    : {198ecec3-5328-11e1-b3a7-001dd9e9c172}
Volume s/n     : 625134160 (0x2542CA50)
First install  : 2012/02/13 09:35:41.928 (Local Time)       2nd/3rd
Username       : Laura 2012-02-13 09:35:49 (UTC)            6th/7th
```

Fig 25 Initial device insertion sample 1

```
Serial num/iid : AA33046300039387&0
Long name      : Disk&Ven_Verbatim&Prod_STORE_N_GO&Rev_1100
Friendly name  : Verbatim STORE N GO USB
Last in dc 307 : 2012-02-13 10:45:19 (UTC)                  4th
Last in dc 30d : 2012-02-13 10:45:27 (UTC)                  5th
Last in enumusb: 2012-02-13 10:45:15 (UTC)                  1st/2nd/3rd
Last in dc a5  : 2012-02-13 10:45:15 (UTC)                  1st/2nd/3rd
Last test time : 2012-02-13 10:45:30 (UTC)                  6th
Vendor ID       : 18A5 (Verbatim, Ltd)
Product ID     : 0304
Drive\Volume   : STORE N GO 2012-02-13 10:45:50 (UTC)       8th
Volume GUID    : {13033b7b-562f-11e1-987c-001dd9e9c172}
Volume s/n     : 2827365395 (0xA8862C13)
First install  : 2012/02/13 10:45:15.585 (Local Time)       1st/2nd/3rd
Username       : Laura 2012-02-13 10:45:31 (UTC)            7th
```

Fig 26 Initial device insertion sample 2

This experiment shows that there is a consistent pattern to the way values are populated/timestamped. The devices monitored took approximately 150ms to populate the values shown. It is not unusual for devices to be listed in a report that do not have all these values populated. The unpopulated values are not just the latter ones which may have indicated a short device insertion see Fig 27. There is no obvious reason for this.

```
Serial num/iid    : 7&2b6f5f5d&0&HZQJ2G1&0
Long name         : Disk&Ven_Dell&Prod_USB_Mass_Storage&Rev__200
Friendly name     : Dell USB Mass Storage USB Device
Last in dc 307    : 2011-03-19 09:41:05 (UTC)
Drive\Volume      : E:\ 2011-03-19 09:41:11 (UTC)
Volume GUID       : {2ed9e81e-4af8-11e0-a2a4-c44619ff487a}
First install     : 2011/03/19 09:41:04.095 (Local Time)
```

Fig 27 Sample of a device with sparsely populated keys

5.2.5.2 *Sample of a USB stick used by separate users on one system.*

An unexpected phenomena was noticed while testing the script to ensure that the timestamps for inserts by multiple users, reported correctly from HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{volume guid}. Harlan Carvey referencing Rob Lee's research states that "by the presence of the key (mountpoints2) within the user's hive there is now an association with a specific user."

It was found that the Mountpoints2 key is updated for all logged on users when a device in inserted, as seen in Fig 28. So if multiple users are logged on a user hive may have an entry in mountpoints2 for a device that was never accessed by that logged in user. In the sample below the hives were extracted shortly after the USB insertion so the occurrence is still obvious. However if a subsequent access was made by one username when the other users were not logged on, the update would be hive specific and would overwrite the fact that a simultaneous update had happened in the past. The findings were that it is only correct to associate a device with a user, if that user and only that user, has accessed a device. In the case where multiple users are recorded as having accessed a device, you can only associate a device with a user if the last recorded MountPoints2 timestamp across the entire system is unique to that user. Previous access recorded in another user hive or non-unique MountPoint2 timestamps would need corroborating evidence, perhaps in the form of a .lnk file, to show actual usage.

This experiment shows that the presence of a device specific MountPoints2 key in a user hive may not be sufficient to associate the device with that user. This is a good demonstration of the need to analyse a system as a whole rather than just focusing on one aspect.

```
Serial num/iid     : 200602668009F6B085B3&0
Long name          : Disk&Ven_SanDisk&Prod_Cruzer_Edge&Rev_1.10
Friendly name      : SanDisk Cruzer Edge USB
Last in dc 307     : 2012-02-24 09:24:31 (UTC)
Last in dc 30d     : 2012-02-24 09:24:32 (UTC)
Last in enumusb    : 2012-02-24 09:24:31 (UTC)
Last in dc a5      : 2012-02-24 09:24:31 (UTC)
Last test time     : 2011-09-19 13:17:45 (UTC) Mon Sep 19 14:17:45 IST 2011
Vendor ID          : 0781 (SanDisk Corp.)
Product ID         : 556B
Drive\Volume       : E: (mountdev)
                   : E:\ 2011-09-19 13:15:02 (UTC)
Volume GUID        : {55782b83-ca94-11e0-b929-0026b9f52bfa}
Volume s/n         : 3368049607 (0xC8C05BC7)
First install      : 2011/09/19 14:14:52.609 (Local Time)
Username           : Jenny 2012-02-24 09:24:36 (UTC)
                   : Jacky 2012-02-24 09:24:36 (UTC)
                   : Stephanie 2012-02-24 09:24:36 (UTC)
```

Fig 28 Sample of multiuser mountpoints2 update

5.2.5.3 *Experiment to evaluate timestamps in DeviceClasseses and Enum*

Correlation on a large scale highlighted some artefacts that did not always behave as anticipated. It was noted that the Enum\USB\VID&PID and the System\CurrentControlSet\Control\DeviceClasses\{53f5630d…} timestamps were not as expected.

Harlan Carvey in his book Windows Forensic Analysis (Registry Analysis Chapter 4) referring to the DeviceClasses key states that. “when a device is connected to a Windows system, a subkey called “Control” is created; when the device is disconnected from the system, the Control subkey is deleted. Both of these actions cause the *LastWrite* time of the DeviceClasses key for the specific device to be modified.“ Rob Lee subsequently found that this key holds the “first time device connected after last reboot”. As this did not always appear to be the case with the sample data set, it was decided to try and verify Rob Lee’s findings.

A Windows 7 system was used to conduct experiments to verify what triggers an updated timestamp for the keys highlighted in Fig 30 and Fig 31 (Page 39). This experiment took the form of some kind of USB action(or inaction) followed by a registry snapshot taken with the helix boot USB and extractreg.sh created for the project. There were a number of findings of interest here:

It was found that the Enum\USB\VID&PID, System\CurrentControlSet\Control\ DeviceClasses\{53f5630d…} and the {53f56307…} key only updates the timestamp on the first insertion after a shutdown, as described by Lee. If a user is inclined to hibernate a system a USB device can be inserted and removed any number of times and this key will not update until the first insertion after the next full reboot. The removal of a key by any means did not update the time on these keys in the experiments performed. Several methods were tried, safe removal followed by physical removal, safe removal only and physical removal only, all followed by a shutdown or a hibernate.

Name	Key
Last insertion 307	HKLM\System\CurrentControlSet\DeviceClasses\{53f5630**7**-b6bf-11d0-94f2-00a0c91efb8b}
Last insertion 30d	HKLM\System\CurrentControlSet\DeviceClasses\{53f5630**d**-b6bf-11d0-94f2-00a0c91efb8b}
(enum)insert	HKLM\System\CurrentControlSet\Enum\USB\VID_9999&PID_9999
Volume name/time	HKLM\Software\Microsoft\Windows Portable Devices\Devices\WPDBUSENUMROOT...s/n...
First Install	setupapi log files
Username	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2

Fig 29 Registry keys monitored for evidence relating to a single USB device

Time = time of insertion	
Test1 - none	Device had never been installed in the system - shutdown
Test2 - 10:44	Device was installed for the first time - the sequence here tells us the order the keys were created in – device installed, viewed, safely removed – system shutdown
Test3 - 12:03	Point of note here the device was extracted prior to updating the users mountpoints2 - device installed, viewed, safely removed – system shutdown
Test4 - 12:18	The device was removed post shutdown - device installed, viewed, device not removed – system shutdown
Test5 - 12:38	The device was removed without doing safe eject - device installed, viewed, device removed physically only – system shutdown
Test6 - 14:06	Install device and leave unattended for auto hibernate/ power off from power management – removed post power off
Test7 - none	Booted windows, it searched for and removed device - shutdown
Test8 - none	Booted windows
Test9 - 16:55 & 17:03	Installed twice in one session only enum & mp2 update, device classes only registers first install after boot from full shutdown, i.e. not hibernation, sleep etc. - shutdown

Fig 30 Record of timed actions

Name	Test1	Test2	Test3	Test4	Test5	Test6	Test7	Test8	Test9
Last insertion 307	-	10:45:19	12:03:40	12:18:40	12:38:45	14:07:24	14:07:24	14:07:24	16:55:50
Last insertion 30d	-	10:45:27	12:03:40	12:18:40	12:38:45	14:07:24	14:07:24	14:07:24	16:55:50
(enum)insert	-	10:45:15	12:03:40	12:18:40	12:38:45	14:07:24	14:07:24	14:07:24	17:03:06
Volume name/time	-	10:45:50	10:45:50	10:45:50	10:45:50	10:45:50	10:45:50	10:45:50	10:45:50
First Install		10:45:15	10:45:15	10:45:15	10:45:15	10:45:15	10:45:15	10:45:15	10:45:15
Username		10:45:31	10:45:31	12:18:40	12:38:45	14:07:24	14:07:24	14:07:24	17:03:07

Fig 31 Times recorded after actions

It was found that Enum\USB\VID&PID and System\CurrentControlSet\Control\ DeviceClasses\{53f5630d… did not retain these dates on several systems used in testing. The sample output in section 5.2.2 page 31 shows an example of this. This particular system registry had traces of 45 usb devices inserted, all of which showed the same/very similar times for these keys. Further inspection of these hives showed that every key in the Enum tree had been updated on or around the timestamps recorded. Registry Decoder (evaluated above) was used to get all the timestamps from the Enum tree. In one registry there were 17,000 keys in the tree, all updated.

The test data for this project consists of (8 x XP),(7 x Windows 7) & (3 x Vista) registry samples. No XP set exhibits this “global enum event”, nine out of ten, of the Win7/Vista sets do. The tenth set is from a simulated use system. Nine of the hives have a timestamp across the Enum tree that is +/- 20 seconds and in some cases much smaller. However if a USB device is inserted post this global Enum timestamp update, the new timestamp is valid and is still worth reporting. With a registry that has more than five USB device entries the existence of this global timestamp is easily apparent but with fewer devices it may not be. On one of the test systems one of the DeviceClasses keys also exhibited the same sort of behaviour. The key HKLM\System\CurrentControlSet\DeviceClasses\{53f5630**d**-b6bf-11d0-94f2-00a0c91efb8b} updated across every device with a timestamp twenty minutes (see Fig 19 Fig 20 Fig 21 Fig 22) post the “Enum Event”.

Some investigations were performed to identify the cause of this global Enum event. At the time of writing, the event has not been identified however certain events have been ruled out. It is not a shutdown, hibernate, sleep, Volume Shadow snapshot, device insertion or extraction, all of which were forced while a system was being monitored. On one occasion this event happened while a system was in active use and the registry keys were being monitored (with USBDeview). A USB keyboard and mouse were in active use on the system so this ruled out USB power management as a possible cause. No updates or Antivirus scan were being run at the time and nothing unusual showed up in any system logs or taskman. The frequency of this enum update on the monitored system, was approximately every 2-4 days which could equate to 24 hours uptime but this was hard to measure precisely.

Rob Lee states that first connection since last reboot and last connection times can be determined from Enum/USB/Vid&Pid/"Device s/n" and Enum/USBSTOR/VendorProductRev/"Device s/n" (24). Both of which are affected by the "global enum event". Tests have proved this statement unreliable on the Vista and Windows 7 systems in the sample data set. Some of the software reviewed was designed using these registry keys and report the non device specific timestamps. USBDeview (see 0) is an example of this. It is also a good way of testing if a system is exhibiting this behaviour. At first glance this product was thought to be flawed but it is reporting the Enum tree keys. The author states “According to user reports, on some systems the 'Last Plug/Unplug Date' and the 'Created Date' values are initialised after reboot. This means that these columns may display the reboot time instead of the correct date/time.” This time proves to be the “Enum event” time and not the time of a reboot.

The Enum tree is not the only place where the first insertion since last reboot can be stored. However in the test data on occasion it was the only place. As this may report a valid time not recorded elsewhere in the registry it was decided to go ahead and report this value while building in a warning. The possibility of the occurrence of the Enum event in a registry can be identified by inspecting other Enum keys. If the time on HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001 \Enum\USB\VID_????&PID_????\ serial number, is +/- 20 seconds from the early branches of the tree then the there is a strong possibility that the timestamp may not be device specific. A warning has been coded into the script which checks the timestamp above against Enum\ACPI to highlight this issue. It is particularly useful if a system only has one or two devices as exhibited in **Fig 32**. This method proved accurate with all the test hives used.

```
There are 2 USB device(s) associated with this system (M57-TERRY)
```

```
Serial num/iid      : 00D04B881007C255&0
Long name           : Disk&Ven_LaCie&Prod_Rugged_FW/USB&Rev_
Friendly name       : LaCie Rugged FW/USB USB
Last in dc 307      : 2009-12-11 16:58:35 (UTC)
Last in dc 30d      :
Last in enumusb     : 2009-12-11 19:16:55 (UTC) Time may not be device specific
Last in dc a5       : 2009-12-11 16:58:35 (UTC)
Last test time      :
Vendor ID           : 059F (LaCie, Ltd)
Product ID          : 100C
Drive\Volume        :
Volume GUID         :
Volume s/n          :
First install       : 2009/11/19 16:02:31.261 (Local Time)
Username            :

Serial num/iid      : 51491E64&0
Long name           : Disk&Ven_USB_2.0&Prod_Flash_Disk&Rev_8.00
Friendly name       : USB 2.0 Flash Disk USB
Last in dc 307      : 2009-12-07 16:46:33 (UTC)
Last in dc 30d      : 2009-12-07 16:46:33 (UTC)
Last in enumusb     : 2009-12-11 19:16:55 (UTC) Time may not be device specific
Last in dc a5       : 2009-12-07 16:46:33 (UTC)
Last test time      : 2009-11-20 18:31:06 (UTC)
Vendor ID           : 058F (Alcor Micro Corp.)
Product ID          : 6387
Drive\Volume        : TERRYS WORK (E:) 2009-11-20 18:31:22 (UTC)
                    : TERRYS WORK 2009-12-07 16:46:35 (UTC)
Volume GUID         : {cdf3e920-d5f0-11de-937c-0008743801b4}
Volume s/n          : 1249120776 (0x4A741208)
First install       : 2009/11/20 10:31:00.891 (Local Time)
Username            : terry 2009-12-07 16:46:35 (UTC)
```

Fig 32 Sample of Enum event warning

5.2.5.4 Use of external database to associate vendor name

As seen in Fig 32, the four digit Vendor ID recorded in the registry is reported along with a name. This name is derived from looking up the database usb.ids (28) which is maintained as part of the Linux USB project. One could question the validity of looking up an external database maintained by an individual. However, the lookup process is probably no different than the manual lookup process an examiner would go through to retrieve this information. The main difference is that in this case the interpretation is automated.

5.2.5.5 Use of script to associate .lnk files with USB devices

In Windows Vista and 7 Usbdevices.sh will attempt to associate .lnk files in the default directory (See Fig 16) with recorded USB devices. This is done by searching the link file for the USB volume serial number that is recorded by the EMDMgt key. The discovery of a USB volume serial number in a .lnk file proves that that file is associated with a device (29). Absence of a serial number does not prove that a file was never associated with a device. If an association is found further inspection of the .lnk file would be required to establish if the USB device was the birth volume. This could help establish if a file was copied to or from a device. There several link file parsers available both open and closed source such as linkalyser from Sanderson Forensics.

5.3 Userinfo

5.3.1 Goals

- To identify, correlate and report *commonly used* registry artefacts relating to user specific actions on a system.
- To make this process Windows platform independent. No intervention should be required to report artefacts such as open/save MRUs which are stored in different formats for XP/Windows 7.
- To run one script that will report on all users and available user hives.
- To document and log the process of data interpretation and manipulation in order to provide full transparency. (Full details for this script are provided in Appendix B)

5.3.2 Artefact identification

Two main sources were used as starting points to identify commonly used user related artefacts, Harlan Carveys Windows Forensic Analysis book (3) and “Beginning to see the light” (30). These areas of the registry are well traversed so nothing new came to light despite an exhaustive trawl searching for usernames and SIDs in various formats. The focus here was to report the artefacts from multiple hives in a concise, comprehensive manner by running a single script.

One of the most challenging aspects of this script was deciding which registry keys to include. Microsoft and other application providers generate new registry entries or key layouts all the time. A list of commonly used registry entries could never be complete. This explains why products such as Regripper use a modular approach. However, this project was about attempting to use automation and correlation to remove the requirement for user intervention so specific keys had to be selected. Disk base digital forensic investigations try to report evidential artefacts that may support or refute a specific user activity; questions like whether a user ran a particular application, accessed or transferred a file, often need to be addressed. So it was decided to concentrate on keys that may record this type of action. Data from the applications reviewed in section 2.3 and other books/articles relating to the registry (see references generally) were analysed to discover the most commonly reported keys. Ten of the eighteen test hive sets contained a “scenario” on them. When the userinfo.sh script reporting the selected keys was run against these scenarios, it reported values that would highlight the system for further investigation, in each of the ten cases.

5.3.3 Sample Output

(note for brevity some data has been removed)

```
There are 3 users associated with this system (M57-TERRY)

Username : Administrator
  RID                     : 500 (000001F4)
  Last logon              : Mon Jan 21 02:48:23 GMT 2008
  Last password change    : Mon Jan 21 02:57:51 GMT 2008
  Account expires         : No expiry set
  Account enabled         : No
  Last failed logon       : Never
  Number of logons        : 8
  Member of group(s)      : Administrators
```

```
Username : terry
   RID                          : 1000 (000003E8)
   Last logon                   : Fri Dec 11 19:21:18 GMT 2009
   Users Profile Folder         : C:\Users\terry
   Users NTUSER.DAT             : NTUSER.DAT.terry
   Last password change         : Thu Nov 19 00:57:29 GMT 2009
   Account expires              : Does not expire
   Account enabled              : Yes (Password required)
   Last failed logon            : Mon Dec 7 16:06:20 GMT 2009
   Number of logons             : 24
   Member of group(s)           : Administrators

There are 14 groups associated with this system (M57-TERRY)

Groupname : Administrators
   RID                          : 00000220
   Number of members            : 2
   Group members                : Administrator (000001F4)
                                  terry (000003E8)

RECENTDOCS: 2009-12-11 19:22:56 (UTC)
MEDIAPLAYER MRU - There are no entries associated with this user
TYPED URLs - There are no entries associated with this user
RUN MRU LIST - There are no entries associated with this user

USERASSIST EXPLORER -2009-12-11 16:55:49 (UTC)
 UEME_CTLSESSION (14)

USERASSIST DESKTOP - 2009-12-11 19:51:25 (UTC)
 UEME_CTLSESSION (14)
UEME_RUNPATH:C:\Program Files\Internet Explorer\iexplore.exe (1) Thu Nov 19 01:28:26 GMT 2009
 UEME_RUNPIDL:%csidl2%\Windows Mail.lnk (1) Thu Nov 19 19:39:58 GMT 2009
 UEME_RUNPIDL:%csidl23%\Accessories\Paint.lnk (1) Thu Nov 19 20:04:46 GMT 2009
 UEME_RUNPATH:C:\Windows\System32\cmd.exe (29) Fri Dec 11 19:51:25 GMT 2009
 UEME_RUNPIDL:%csidl2%\Accessories\Command Prompt.lnk (6) Fri Dec 11 19:50:23 GMT 2009
 UEME_RUNPATH:C:\Windows\system32\NOTEPAD.EXE (4) Fri Dec  4 20:31:31 GMT 2009
UEME_RUNPATH:D:\start.exe (1) Mon Nov 30 17:18:39 GMT 2009
 UEME_RUNPATH:C:\Program Files\OpenOffice.org 3\program\\swriter.exe (2) Thu Dec  3 17:28:32 GMT 2009
 UEME_RUNPATH:C:\Users\terry\Documents\Downloads\xpadvancedkeylogger.exe (1) Thu Dec  3 17:30:56
GMT 2009
 UEME_RUNPIDL:%csidl23%\XP Advanced Keylogger\Uninstall XP Advanced Keylogger.lnk (1) Thu Dec  3
17:31:36 GMT 2009
 UEME_RUNPATH:C:\Users\terry\Documents\Downloads\keylogger\FamilyKeyLogger-setup.exe (1) Thu Dec  3
17:33:30 GMT 2009
 UEME_RUNPIDL:%csidl23%\Family Keylogger\Family Keylogger.lnk (9) Thu Dec  3 17:34:47 GMT 2009
 UEME_RUNPIDL:%csidl23%\Family Keylogger\Uninstall.lnk (2) Thu Dec  3 17:55:07 GMT 2009
 UEME_RUNPATH:C:\Users\terry\AppData\Local\Google\Chrome\Application\chrome.exe (5) Wed Dec  9
16:53:09 GMT 2009
```

```
 UEME_RUNPATH:C:\Users\terry\Downloads\vnc-4_1_3-x86_win32\vnc-4_1_3-x86_win32.exe (1) Mon Dec  7
18:17:35 GMT 2009
 UEME_RUNPATH:C:\Program Files\RealVNC\VNC4\vncviewer.exe (10) Thu Dec 10 22:17:51 GMT 2009
 UEME_RUNPIDL:%csidl23%\RealVNC\VNC Viewer 4\Run VNC Viewer.lnk (4) Thu Dec 10 22:17:51 GMT 2009
 UEME_RUNPATH:%csidl0%"cmd.exe"  (6) Thu Dec 10 17:55:50 GMT 2009
 UEME_RUNPATH:C:\Windows\system32\FirewallSettings.exe (1) Mon Dec  7 18:23:31 GMT 2009
 UEME_RUNPATH:C:\Users\terry\Documents\Downloads\ccsetup226.exe (1) Tue Dec  8 21:09:15 GMT 2009
 UEME_RUNPATH:C:\Users\terry\Documents\Downloads\Eraser-5.8.7_setup.exe (1) Thu Dec 10 16:28:03 GMT
2009
 UEME_RUNPIDL:%csidl23%\Eraser\Eraser.lnk (9) Thu Dec 10 18:49:34 GMT 2009
 UEME_RUNPATH:C:\Program Files\Eraser\Eraser.exe (1) Thu Dec 10 18:49:34 GMT 2009
UEME_RUNPIDL:%csidl2%\CCleaner\CCleaner.lnk (9) Fri Dec 11 19:22:16 GMT 2009

USER SPECIFIC AUTORUNS -
 C:\Program Files\Windows Sidebar\sidebar.exe /autoRun
 "C:\Users\terry\AppData\Local\Google\Update\GoogleUpdate.exe" /c
 C:\Program Files\Eraser\Eraser.exe -hide

REMOTE DESKTOP TERMINAL SERVERS
  There are no Terminal Server entries associated with this user

SYSTEMS SEEN BY NETWORK BROWSER - There are no entries associated with this user
RECENTLY MAPPED NETWORK DRIVES - There are no entries associated with this user

RECONNECT AT LOGIN NETWORK DRIVES
  There are no entries associated with this user

PRINTERS
  There are no printers listed in HKCU\Printers\Connections
  There are no printers listed in HKCU\Printers\DevModes2

OPEN/SAVE MRUs
```

Fig 33 Sample userinfo.sh output from m57 scenario

5.3.4 Experiments and observations

5.3.4.1 Userassist

In the process of testing the Windows 7 userassist data interpretation, it was noticed that the usage counter did not behave as expected. Some further digging and the reading of Didier Stevens excellent article (31), led to the understanding that a second counter, a focus counter existed in Windows 7. In prior versions of Windows the usage counter starts at 5, with lower numbers indicating focus such as a “mouseover” rather than a run (32). Stevens states that the counter now starts at 1 and was functionally the same other than this. Findings here indicate this not to be the case. Steven’s research was done over a couple of days within the same month. My research was somewhat slower and so I noted that the counter was often reset to zero. Further research indicates that this appears to happen by calendar month and that the counts are a record of the number of times, an application was run in a particular month. To test this, four hives from different months belonging to same system were parsed using the userinfo.sh script. Note the March hive while dated early April, was extracted from the system before any April usage.

2011-06-29 18:27	NTUSER.DAT.jun (2011)
2012-02-14 13:22	NTUSER.DAT.feb
2012-04-02 19:11	NTUSER.DAT.mar
2012-04-15 21:22	NTUSER.DAT.apr

Fig 34 hives from the same system over four chronological months

Usage Count	Focus count?	Last used	Hive month
(1)	(3)	Sun Feb 12 20:07:09 GMT 2012	February
(0)	(0)	Thu Feb 16 13:13:14 GMT 2012	March

Fig 35 \AccessData\AccessDataForensicToolKit1.80.0\Program\ftk.exe

Usage Count	Focus count?	Last used	Hive month
(6)	(9)	Fri Jun 24 10:56:12 IST 2011	June
(8)	(13)	Tue Feb 14 20:19:12 GMT 2012	February
(1)	(2)	Thu Mar 15 09:34:39 GMT 2012	March
(0)	(0)	Thu Mar 15 09:34:39 GMT 2012	April

Fig 36 \notepad.exe

Usage Count	Focus count?	Last used	Hive month
-	-	-	June hive
(1)	(6)	Sun Feb 12 16:51:32 GMT 2012	February hive
(5)	(29)	Mon Mar 19 22:24:49 GMT 2012	March hive
(0)	(0)	Mon Mar 19 22:24:49 GMT 2012	April hive

Fig 37 \Microsoft Office\Office12\VISIO.EXE

This would be very useful if applied over volume shadows as a report showing a month by month picture of application usage could be generated.

It would also be interesting to spend further time researching what constitutes as focus for the "focus count". The hives above are from my own system and my usage pattern is to hibernate the system and leave applications open. Hence applications are not launched or moused-over often. My recent memory recalls using Visio most days in early March however it was only launched five times. In this case the focus may be a post hibernate launch. More research also needs to be completed to ascertain exactly when the counters are set back to zero. Is it perhaps the first system usage in a new month? Note the last used date is retained even though the usage counter is zero in Fig 37.

5.3.4.2 Number of SIDs in users group

The windows registry stores information about each group in the 'C' value underneath the key HKLM\SAM\Domains\Builtin\Aliases\xxxxxxxx, where xxxxxxxx is the group RID. Fig 38-Fig 41 show examples of this value, where the first highlighted area shows the number of members in a group (in hexadecimal) and the next highlighted area shows the SIDs of the group members(if present). If the member count is set to three you would expect to find three user SIDs listed below. However for the built in group users, the member count is always set to two greater than the number of listed SIDs. This is the case even when no SID is listed; the member count is set to 2. All other groups behave as expected where the number of SIDs tally with the group user count, see Fig 41.

```
CMI-CreateHive{87E016C8-C811-4B12-9C3A-
CDA552F3458D}\SAM\Domains\Builtin\Aliases\00000221 [2010-01-
24T09:57:42Z]
C (REG_BINARY) = 21 02 00 00 00 00 00 00 c8 00 00 00 02 00 01 00 c8 00
00 00 0a 00 00 00 00 00 00 00 d4 00 00 00 d6 00 00 00 00 00 00 00 ac 01
00 00 6c 00 00 00 05 00 00 00 01 00 ........ 01 01 00 00 00 00 00 05 04
```

```
00 00 00 01 01 00 00 00 00 00 05 0b 00 00 00 01 05 00 00 00 00 00 05 15
00 00 00 5e 0a 71 b6 5e 9d 78 47 be b0 42 62 e9 03 00 00 01 05 00 00 00
00 00 05 15 00 00 00 5e 0a 71 b6 5e 9d 78 47 be b0 42 62 eb 03 00 00 01
05 00 00 00 00 00 05 15 00 00 00 5e 0a 71 b6 5e 9d 78 47 be b0 42 62 ec
03 00 00
```

Fig 38 Vista Registry C value

```
CMI-CreateHive{899121E8-11D8-44B6-ACEB-
301713D5ED8C}\SAM\Domains\Builtin\Aliases\00000221 [2009-11-
12T06:55:01Z]
C (REG_BINARY) = 21 02 00 00 00 00 00 00 c8 00 00 00 02 00 01 00 c8 00
00 00 0a 00 00 00 00 00 00 00 d4 00 00 00 d6 00 00 00 00 00 00 00 ac 01
00 00 18 00 00 00 02 00 00 00 01 00 ....... no sids listed
```

Fig 39 Windows 7 Registry C Value

```
SAM\SAM\Domains\Builtin\Aliases\00000221 [2004-08-19T17:01:55Z]
C (REG_BINARY) = 21 02 00 00 00 00 00 00 c8 00 00 00 02 00 01 00 c8 00
00 00 0a 00 00 00 00 00 00 00 d4 00 00 00 38 01 00 00 00 00 00 00 0c 02
00 00 18 00 00 00 02 00 00 00 01 00 ....... no sids listed
```

Fig 40 XP Registry C value

```
CMI-CreateHive{87E016C8-C811-4B12-9C3A-
CDA552F3458D}\SAM\Domains\Builtin\Aliases\00000222 [2007-12-
03T17:58:35Z]
C (REG_BINARY) = 22 02 00 00 00 00 00 00 c8 00 00 00 02 00 01 00 c8 00
00 00 0c 00 00 00 00 00 00 00 d4 00 00 00 fc 00 00 00 00 00 00 00 d0 01
00 00 1c 00 00 00 01 00 00 00 01 00 ...........00 65 00 64 00 01 05 00
00 00 00 00 05 15 00 00 00 5e 0a 71 b6 5e 9d 78 47 be b0 42 62 f5 01 00
00
```

Fig 41 Windows 7 non-users group (Guests)

5.3.4.3 Simultaneous run against multiple restore points from a single system

One of the factors enabling multiple user hives to be reported simultaneously is that the script will find any file in the default directory starting with ntuser.dat or NTUSER.DAT. If the extractreg.sh and getraw.sh are used to extract the required files from an image, all the user hives will be automatically postfixed with the name of their source directory. This is often the username e.g. NTUSER.DAT.John. If userhive files from another source are used they will be recognized as long as the filename starts with NTUSER.DAT or ntuser.dat. This means that in one iteration of the script, all the users on a single system or a complete set of restore points can be analysed. For example if twenty versions of one user hive from system restore points, are extracted and placed in a single directory named ntuser.dat.rp1, ntuser.dat.rp2…… The userinfo.sh script will automatically run against each hive producing a report that details user activity over time. This script would help with applying a comparative method for reconstructing digital events using restore points (33).

5.3.4.4 Open/Save MRUs

The Windows 7/Vista OpenSavePidlMru was particularly challenging to interpret. Some of the values store a very large amount of binary data. Being unable to find much verifiable information about the layout of these values, another approach was needed. When parsing the values manually Unicode filenames stand out so a data mask was developed that when applied produced the same results. This mask was successful when run against the three Vista and seven Windows 7 hives, parsing more than five hundred values correctly and zero incorrectly. For further explanation see section 29 in *Appendix B – Details of data manipulations performed by userinfo.sh*.

5.4 Networkinfo

5.4.1 Goals

- To identify, correlate and report registry artefacts relating to network activity on a system. User specific network actions are reported in userinfo.sh.
- The process should be Windows platform independent.
- To document and log the process of data interpretation and manipulation to provide transparency. (Full details for this script are provided in Appendix C)

5.4.2 Sample Output

```
NETWORK PROFILES RECORDED ON THIS SYSTEM

Wayport_Access 3
   guid                : {65D9480C-D8FB-495F-95E3-7149A800E281}
   Date created        : Wednesday 5 August 2009 22:58:49:63
   Last connected      : Friday 7 August 2009 21:15:54:18
   Default Gateway MAC : 00 90 fb 11 cf 96 ( PORTWELL, INC.)
   DNS suffix          : ncv.lax.wayport.net

avoca
   guid                : {6EE1C79E-6128-492A-9068-D41FFAD021A7}
   Date created        : Saturday 30 August 2008 16:24:19:411
   Last connected      : Sunday 25 October 2009 17:18:30:971
   Default Gateway MAC : 00 00 c5 e9 20 98 ( FARALLON COMPUTING/NETOPIA)
   DNS suffix          : <none>

gods
   guid                : {9CFB0795-3F47-4AE8-B69D-9B7F0F6A5680}
   Date created        : Wednesday 7 July 2010 13:45:21:97
   Last connected      : Friday 9 July 2010 02:59:56:55
   Default Gateway MAC : 00 0c e5 73 67 55 ( Motorola Mobility, Inc.)
   DNS suffix          : hsd1.ma.comcast.net.

ACTIVE NETWORK INSTANCES RECORDED ON THIS SYSTEM

Network instance guid {CF7A4542-9C5E-490A-A459-426893C508F2}

  Hardware            : Dell Wireless 1390 WLAN Mini-Card
  Domain name         : (no data)
  Dhcp IP Address     : 192.168.1.14 (255.255.255.0)
  Dhcp Server         : 192.168.1.254
  Dhcp Enabled        : yes
  Dhcp gateway MAC    : Not recorded
  Lease period (secs): 0x00000e10 (3600)
  Lease obtained      : Thu Jan 19 16:09:11 GMT 2012
  Lease Terminates    : Thu Jan 19 17:09:11 GMT 2012
  Static IP Address   : Not recorded
  Network Connection : Wireless Network Connection
  Media Subtype       : Not Available
  Pnp Inst ID :
PCI\VEN_14E4&DEV_4311&SUBSYS_00071028&REV_01\4&13F7E9EA&0&00E1

OUTGOING SHARES :

  name : print$

  path : C:\Windows\system32\spool\drivers
  name : Dell V505

  path : Dell V505,LocalsplOnly



  name : Broderbund PDF Creator

  path : Broderbund PDF Creator,LocalsplOnly
```

Fig 42 Sample Vista output from networkinfo.sh

```
Network instance guid {86FC0C96-3FF2-4D59-9ABA-C602F213B5D2}

  Hardware            : Compaq WL110 Wireless LAN PC Card
  Domain name         : (no data)
  Dhcp IP Address     : Not recorded ()
  Dhcp Server         : 255.255.255.255
  Dhcp Enabled        : yes
  Dhcp gateway MAC    : Not recorded
  Lease period (secs): 0x00000e10 (3600)
  Lease obtained      : Fri Aug 27 16:46:18 IST 2004
  Lease Terminates    : Fri Aug 27 17:46:18 IST 2004
  Static IP Address  : Not recorded
  Network Connection : Wireless Network Connection
  Media Subtype       : Not Available
  Pnp Inst ID : PCMCIA\COMPAQ-COMPAQ_WL110_PC_CARD-E648\1
  Wireless access points accessed by this NIC :


     SpeedStream

       WAP MAC        - 00 c0 02 b9 00 78 (SERCOMM CORPORATION)

       Encryption     - WEP

       Authentication - Open
       Last access    -

OUTGOING SHARES :

  name : Temp

  path : C:\Temp
```

Fig 43 Sample XP output from networkinfo.sh

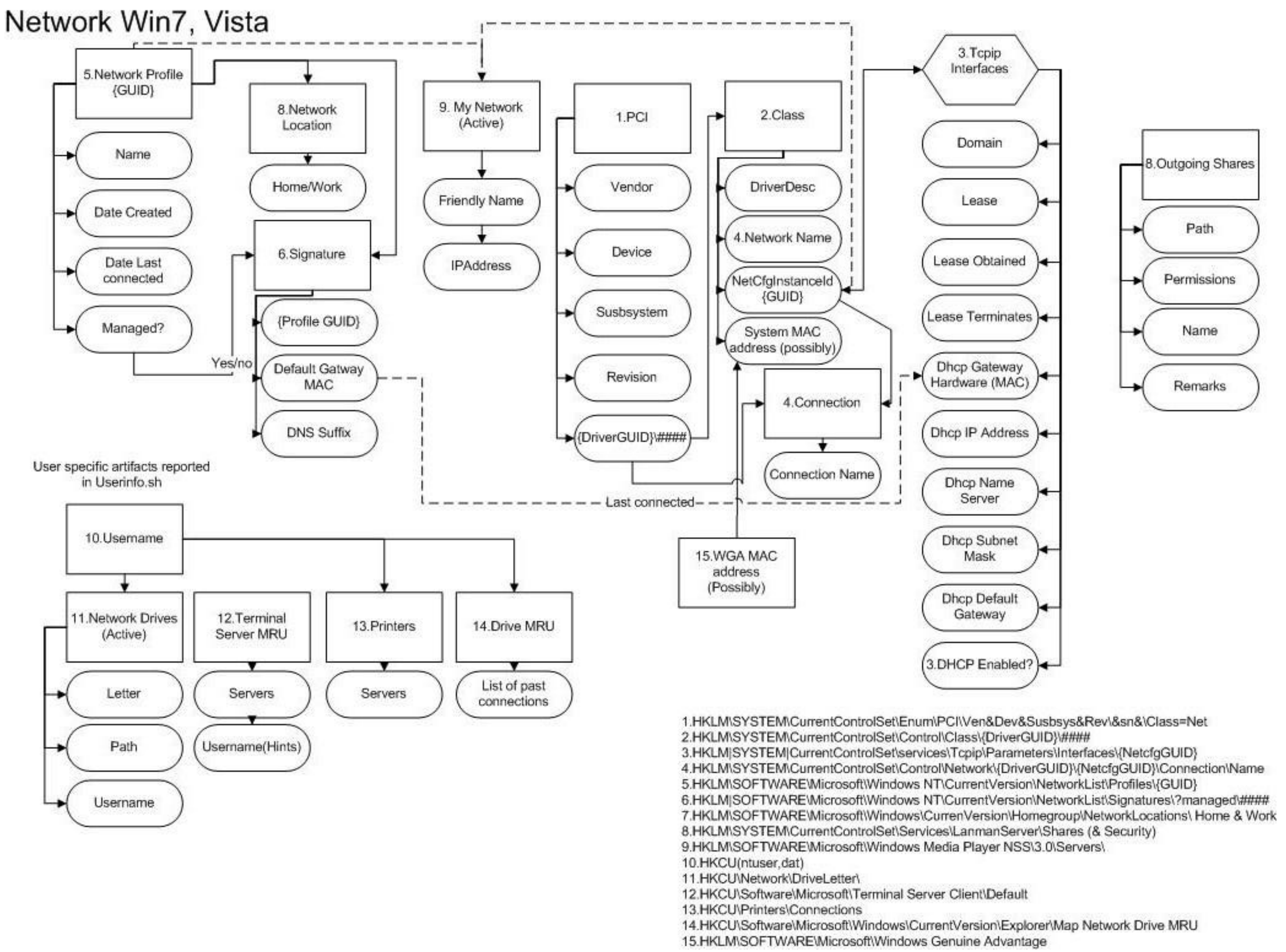
```
NETWORK PROFILES RECORDED ON THIS SYSTEM

eircom5376 0322
   guid               : {106AEA67-3D31-464B-B6B4-2ABC4DF1DAD2}
   Date created       : Saturday 14 August 2010 10:46:55:603
   Last connected     : Wednesday 29 June 2011 18:23:53:865
   Default Gateway MAC : 00 24 92 af c4 40 ( Motorola, Broadband Solutions
Group)
   DNS suffix         : <none>
   Profile location   : Home

JACKY-LAPTOP-83090
   guid               : {7A4CA7EB-AA56-4BB3-9B45-6D10B72A8A7E}
   Date created       : Wednesday 18 August 2010 20:15:04:869
   Last connected     : Wednesday 18 August 2010 20:32:44:184
   Default Gateway MAC : (no data)
   DNS suffix         : <none>
   Profile location   : Work
```

Fig 44 Sample Windows 7 output showing profile location data

5.4.3 Artefact identification

Windows Forensic Analysis by Harlan Carvey was used as a starting point to identify network related artefacts. Across each operating system supported, one network device was identified and then registry searches were performed using the identifying features of that device. This did not uncover any additional useful information but it did assist in generating a flowchart of the interconnections between the artefacts (see 5.4.4). Another interesting resource was a presentation by Eric Rowe (34) detailing where to find information such as encryption and authentication settings for a wireless access connection in Windows XP.

5.4.4 Connections Diagram



1.HKLM\SYSTEM\CurrentControlSet\Enum\PCI\Ven&Dev&Susbsys&Rev\&sn&\Class=Net
2.HKLM\SYSTEM\CurrentControlSet\Control\Class\{DriverGUID}\####
3.HKLM|SYSTEM|CurrentControlSet\services\Tcpip\Parameters\Interfaces\{NetcfgGUID}
4.HKLM\SYSTEM\CurrentControlSet\Control\Network\{DriverGUID}\{NetcfgGUID}\Connection\Name
5.HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Profiles\{GUID}
6.HKLM|SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Signatures\?managed\####
7.HKLM\SOFTWARE\Microsoft\Windows\CurrenVersion\Homegroup\NetworkLocations\ Home & Work
8.HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Shares (& Security)
9.HKLM\SOFTWARE\Microsoft\Windows Media Player NSS\3.0\Servers\
10.HKCU(ntuser,dat)
11.HKCU\Network\DriveLetter\
12.HKCU\Software\Microsoft\Terminal Server Client\Default
13.HKCU\Printers\Connections
14.HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Map Network Drive MRU
15.HKLM\SOFTWARE\Microsoft\Windows Genuine Advantage

5.4.5 Experiments and observations

5.4.5.1 Interpretation of MAC addresses

If a MAC address is recorded for a Wireless Access Point or Network Card, this can reveal more information than just the address. The IEEE maintains a database of the organisationally unique identifiers (oui) of vendors which relate to the first six characters of a MAC address (35). For example the value DhcpGatewayHardware is stored under the key HKLM\SYSTEM\CurrentControlSet\Services \Tcpip\Parameters\Adapters. If this value is set to c0 a8 01 fe 06 00 00 00 00 24 92 af c4 40. The highlighted characters are the MAC address with 00 24 92 being the *oui*. Networkinfo.sh looks up the IEEE database and reports the vendor name, which in this case is Motorola Broadband solutions. This data could be useful for highlighting the existence and identification of additional network adapters or WAPs.

The MAC address could be further interpreted with WAP MAC address geolocation services from sources such as Google, Skyhookwireless.com or wigle.net. It was decided not to attempt to automate this at present as there is a level of uncertainty about the validity of this information. Unlike the IEEE oui database which is gathered from manufacturers, this geolocation data is often pooled from multiple sources including unverified wardriving (36) dumps.

5.4.5.2 Differences in WAP associations for XP and Vista/7

In Windows XP/Vista/7 each network card or controller has a network instance id in the form of a guid. Connections to wireless access point are also recorded in the registry and reported by the script. In Windows XP WAPs can be associated with a particular network controller and so are reported beneath the controller. This association is not as definite in Windows Vista/7 so the WAPs are reported separately. Windows Vista/7 registries were searched by network instance guid, hardware, pnp instance id, WAP MAC and gateway MAC to find any artefacts that may have made associations here. The only consistent association that could be made was the connection between the last connected WAP *default MAC gateway* and the *dhcp gateway MAC.*

5.4.5.3 Regripper comparative run

Networkinfo.sh was selected to do a comparative run against Regripper (37) because the output generated from the sample hives, is smaller than from the other scripts. The output is still a few pages long so it has been included as Appendix E – Output from networkinfo.sh and regripper.

- To generate the output from regripper eight plugins/commands had to be run versus one for networkinfo.sh.
- With Regripper the examiner must know for each plugin, which hive the relevent keys are stored in. Sometimes the original logic used for deciding which hive data should be recorded in is unclear. It can be like deciding into which category to place a multifaceted book in a library. Also regripper is case sensitive so even if you do get the hive right you may have to try system as opposed to SYSTEM. Networkinfo.sh does not require the user to know which hive the keys reside in or the case of the hive name, to run the tool.
- The output generated from regripper is considerably larger and requires manual correlation to connect the entries. For example the output from "network" plugin could be correlated with "nic" plugin by the network instance guid.
- Further automated interpretation could be performed by regripper. For example EnableDHCP is reported as "1", this is reported as "yes" by neworkinfo.sh.

- The output from ssid.pl could be reported by network instance rather than as a list.
- The selection of an appropriate script can be operating system dependant with regripper. For example to report WAPs you should run ssid.pl for XP and for Vista & 7 you should run vista_wireless.pl. If key/data formats differ across operating systems typically Vista & 7 are grouped together but this is not always the case. Windows XP & Vista have similar values for userassist but for Windows 7 the layout is different. Networkinfo will select the appropriate part of the script to run for you depending on the operating system.
- In Windows XP networkinfo.sh reports additional data than ssid.pl. Both scripts report the WAP name, MAC and last accessed time. Networkinfo also reports the encryption and authentication types.
- The output from vista_wireless.pl omits three network profiles that are reported by networkinfo.sh. See Appendix E, the profiles Network, Network 2 and Network 3 are not reported. Regripper and networkinfo.sh report the profile name and last connected time. In addition Networkinfo.sh reports the profile guid, date created, DNS suffix, default gateway MAC address and WAP manufacturer. For Windows 7 networkinfo also reports the profile location e.g. home, work etc.
- Regripper nic2.pl reports some additional information to networkinfo.sh. It reports the uninitiated network instances and the values from HKLM\SYSTEM\CurrentControlSet\Services \Tcpip\Parameters\{network instance guid}. These are included in the expert logs generated by networkinfo.

5.5 Systeminfo

5.5.1 Goals

- To identify, correlate and report commonly used artefacts relating to the system that are stored in the registry.
- This process should be Windows platform independent. No intervention should be required to report artefacts such as Timezone name which differ from XP to Windows Vista/7.
- There is no great scope for correlation in this area but this script is viewed as being necessary to provide a more complete and usable set by its inclusion.
- To document and log the process of data interpretation and manipulation to provide transparency. (Full details for this script are provided in Appendix D)

5.5.2 Sample Output

```
Current Control Set            : 001
Registered Organisation        : Acme
Registered Owner               : Acme
Computer Name                  : ACME-N6A1H8ZLJ1
Product serial number          : 55277-005-0859956-21148
Product Name                   : Microsoft Windows XP Service Pack 1
Current Version                : 5.1
System Root                    : C:\WINDOWS
Installation Date              : Wed Jan 30 13:41:00 GMT 2008
Last logged Shutdown Time      : Wed Jan 30 14:51:23 GMT 2008
Last user logged in            : Caster Troy
System Directory               : %SystemRoot%\system32
Drive letters                  : A: D: C: E:
Daylight savings Timezone      : Pacific Standard Time (Bias -60 Minutes)
Standard Timezone              : Pacific Standard Time (Bias +0 Minutes)
Timezone bias                  : UTC +480 Minutes
Current time bias              : UTC +480 Minutes
Network time protocol is       : synchronised
Timezone last updated          : 2008-01-31 04:32:17 (UTC)
Daylight saving starts on Sunday in the 1st week of April at 02:00:00:00
Standard time starts on Sunday in the last week of October at 02:00:00:00

SYSTEM WIDE AUTORUNS
 C:\Program Files\VMware\VMware Tools\VMwareTray.exe
 C:\Program Files\VMware\VMware Tools\VMwareUser.exe

INSTALLED APPLICATIONS (Uninstall)
 AddressBook --- 2008-01-30 13:37:43 (UTC)
 Branding --- 2008-01-30 13:39:22 (UTC)
 Connection Manager --- 2008-01-30 13:33:59 (UTC)
 DirectAnimation --- 2008-01-30 13:37:43 (UTC)
```

```
DirectDrawEx --- 2008-01-30 13:37:36 (UTC)
Fontcore --- 2008-01-30 13:37:36 (UTC)
ICW --- 2008-01-30 13:37:43 (UTC)
IE40 --- 2008-01-30 13:37:36 (UTC)
IE4Data --- 2008-01-30 13:37:36 (UTC)
IE5BAKEX --- 2008-01-30 13:37:36 (UTC)
IEData --- 2008-01-30 13:37:36 (UTC)
Microsoft NetShow Player 2.0 --- 2008-01-31 04:32:54 (UTC)
MobileOptionPack --- 2008-01-30 13:37:36 (UTC)
MPlayer2 --- 2008-01-31 04:32:54 (UTC)
NetMeeting --- 2008-01-30 13:37:43 (UTC)
OutlookExpress --- 2008-01-30 13:37:43 (UTC)
PCHealth --- 2008-01-30 13:37:46 (UTC)
SchedulingAgent --- 2008-01-30 13:37:36 (UTC)
Google Toolbar for Internet Explorer --- 2008-01-30 14:09:23 (UTC)
VMware Tools --- 2008-01-30 13:58:23 (UTC)
WinZip 11.1 --- 2008-01-30 14:10:08 (UTC)
Google Toolbar for Internet Explorer --- 2008-01-30 14:09:23 (UTC)
```

Fig 45 Sample output from systeminfo.sh

```
Current Control Set            : 001
Registered Organisation        : (no data)
Registered Owner               : user
Computer Name                  : 514-06
Product serial number          : 76478-OEM-0017337-69599
Product Name                   : Microsoft Windows XP Service Pack 3
Current Version                : 5.1
System Root                    : C:\WINDOWS
Installation Date              : Fri Nov 5 01:18:18 GMT 2010
Last logged Shutdown Time      : Fri Mar 9 04:21:28 GMT 2012
Last user logged in            : user
System Directory               : C:\WINDOWS\system32
Drive letters                  : C: D: E: F: G: H: I: J: K: L: M: N: O: P:
Daylight savings Timezone      : 대한민국 표준시 (Bias +0 Minutes)
Standard Timezone              : 대한민국 표준시 (Bias +0 Minutes)
Timezone bias                  : UTC -540 Minutes
Current time bias              : UTC -540 Minutes
Network time protocol is       : synchronised
Timezone last updated          : 2010-11-05 03:23:37 (UTC)
Daylight saving starts on Sunday in the not week of at 00:00:00:00
Standard time starts on Sunday in the not week of at 00:00:00:00
```

Fig 46 Sample systeminfo.sh output from a Korean language hive

5.5.3 Artefact identification

The selection of artefacts for inclusion was again difficult and somewhat subjective. For this script, It was decided to only report artefacts that are system wide, as opposed to any user specific aspects. The evaluation of other tools (see section 2.3) illustrated the artefacts that are commonly reported. A registry trawl was also performed with regedt32.exe to find any other keys that may be of interest to this section. Some values are included that are not commonly reported such as last user logged in and a list of all the drive letters assigned.

5.5.4 Experiments and observations

5.5.4.1 Timezone

Particular attention was given to reporting the timezone and bias information correctly, see comparision with regripper output. Fig 47 shows that the *twos compliment* calculations are not done to report a UTC negative time, such as -540 correctly. Fig 48 shows that the bias for daylight saving and standard timezones is not reported.

```
 ControlSet001\Control\TimeZoneInformation
LastWrite Time Fri Nov 5 03:23:37 2010 (UTC)
  DaylightName          -> 대한민국 표준시
  StandardName          -> 대한민국 표준시
  Bias                   -> 4294966756 (71582779.2666667 hours)
  ActiveTimeBias         -> 4294966756 (71582779.2666667 hours)
  TimeZoneKeyName        -> N/A
```

Fig 47 Regripper timezone output for Korean hive in Fig 46

```
ControlSet001\Control\TimeZoneInformation
LastWrite Time Thu Jan 31 04:32:17 2008 (UTC)
  DaylightName           -> Pacific Standard Time
  StandardName           -> Pacific Standard Time
  Bias                  -> 480 (8 hours)
  ActiveTimeBias         -> 480 (8 hours)
  TimeZoneKeyName        -> N/A
```

Fig 48 Regripper timezone output for hive in Fig 45

5.5.4.2 Number of drive letters

Across the test hives the number of drive letters recorded appeared to relate to the technical competency of the system user(s). The hive sample of nineteen was too small to be definitive but there was a pattern that indicated, on a standalone system; more drive letters are used by technically competent users. More research would be needed to prove or disprove this theory but it was decided to report the drive letter count anyway as it may prove useful.

6 Evaluation and Discussion of Results

Part of the outcome from this project was the production of a suite of open source Linux tools for registry correlation and interpretation of commonly reported registry artefacts. The tools facilitated some interesting experiments and research. They produced readable, repeatable and verifiable interpretations of data stored in the registry with minimal user input required. They will run on hives produced from 92% of desktop PCs so the potential for usage is high. The typical run times vary by tool from under one minute to the maximum tested fifteen minutes, which was on a heavily used system with multiple user hives.

When compared with Regripper the main difference is an approach of modularity versus integration. Regripper is more comprehensive due to its modular design. There are definite advantages to be had from taking a modular approach to programming, particularly in an environment that changes as frequently as the Windows Registry. However, there are also advantages to using an integrated approach, namely the ability to provide greater correlation and automated interpretation. The examiner has less to do to get the tools running, less data to review and less manual interpretation to perform. The output from the tools produced campares well with Regripper for accuracy and forensic soundness.

Every effort has been made to make sure the tools produce output that would be acceptable in court. Design and testing can only be as good as the test data and processes used. Test hives were retrieved from nineteen different sources but this is a small sample. It is planned to distribute these tools to the forensic community so it is hoped that feedback will further enhance the suite. The tools are open source, have the facility to provide a digital chain of custody and are fully documented. This should ensure that the reports produced can be presented and explained in court.

The selected scripting language BASH, did not cause any problems or limitations. However if the project was to start again more consideration would be given to Perl as it appears to be a more widely used language for forensic scripting.

There are no regrets about picking this topic. A deeper understanding was gained of the chosen topic and also other peripheral areas such as admissibility and data interpretation. Some interesting findings were made, such as that a usb device registers to all logged in users and that userassist reports monthly usage as opposed to total usage in Windows 7. These findings were easier to spot as a direct result of using automated cross hive registry correlation.

Some areas that if time had permitted have the scope for further development would be to investigate; producing the output in DFXML, improving the user interface, incorporating Windows 8 specific artefacts, testing the tools with Cygwin and VMs and developing a tool to produce a timelined report of Windows 7 Userassist values from volume shadows.

It is firmly intended to publish these tools so any feedback positive or negative that could improve them would be greatly appreciated to jackyfox@eircom.net.

7 References

1. **Farmer, Derrick J.** A Windows Registry Quick Reference:. *www.forensicfocus.com.* [Online] [Cited: 2012 04 20.] http://www.forensicfocus.com/downloads/windows-registry-quick-reference.pdf.

2. **Russinovich, Mark.** Insidethe Registry. [Online] [Cited: 2012 04 20.] http://technet.microsoft.com/en-us/library/cc750583.aspx.

3. **Carvey, Harlan.** *Windows Forensic Analysis.* 2009. ISBN 978-1-59749-422-9.

4. **Microsoft.** Microsoft knowledgebase article 256986. [Online] [Cited: 13 01 2012.] http://support.microsoft.com/kb/256986.

5. **Locard, Edmund.** Exchange principal. [Online] [Cited: 24 04 2012.] http://en.wikipedia.org/wiki/Locard%27s_exchange_principle.

6. **Beecher, Ed.** Automation and the Trade Unions. [Online] [Cited: 04 04 2012.] http://www.marxists.org/history/etol/newspape/isr/vol23/no03/beecher.html.

7. **al, Wytze P. Oosterhuis et.** Evaluation of LabRespond, a New Automated Validation System for Clinical Laboratory Test Results. [Online] http://www.clinchem.org/content/46/11/1811.full.

8. **Mueller, Lance.** Forensickb.com. [Online] 2012 02 29. http://www.forensickb.com/2009/10/enscript-to-obtain-connected-usb.html.

9. **UK, Association of Chief Police Officers.** ACPO Guidelines. [Online] [Cited: 2012 01 10.] http://www.7safe.com/electronic_evidence/ACPO_guidelines_computer_evidence.pdf.

10. **Carrier, Brian.** Open Source Digital Forensics Tools The legal Argument. *digital-evidence.org.* [Online] [Cited: 2012 01 10.] http://www.digital-evidence.org/papers/opensrc_legal.pdf.

11. *Bringing Science to Digitial Forensics with standardized forensic Corpora.* **Garfinkel, Simson.** Supplement 2-11, s.l. : Digital Investigation, Vol. 6.

12. *Risk Sensitive Digital Evidence Collection.* **Brown, Erin E. Keneally Christopher L. T.** 2, s.l. : Digital Investigation, June 2005, Vol. 2. Pages 101-119.

13. *What does "foresnsically sound" really mean?* **Casey, Eoghan.** 2, June 2007, Vol. 4. Pages 49-50.

14. *Unification of digital evidence from disparate sources.* **Turner, Philip.** s.l. : Digital Forensic Workshop, 2005. http://www.dfrws.org/2005/proceedings/turner_evidencebags.pdf.

15. http://www.netmarketshare.com. *Netmarketshare.* [Online] 08 01 2012. [Cited: 08 01 2012.] http://www.netmarketshare.com/operating-system-market-share.aspx?qprid=10&qpcustomd=0.

16. **Raphael Vasquez et al.** PC Installed Base worldwide 2006-2015. [Online] [Cited: 08 01 2012.] http://www.gartner.com/id=1602818.

17. *Digital Forensics with open source tools (Chap 2 page 11).* **Altheide, Cory & Carvey, Harlan.** s.l. : Syngress, 2011. ISBN 978-1-59749-586-8.

18. **Balenthin, Willi.** [Online] [Cited: 2012 02 07.] http://www.williballenthin.com/registry/#introduction.

19. **Macfarlane, James.** [Online] [Cited: 2012 02 07.] http://search.cpan.org/~jmacfarla/Parse-Win32Registry-0.60/lib/Parse/Win32Registry.pm.

20. *Extending the advanced forensic format to accommodate multiple data sources, logical evidence, arbitrary information and forensic workflow.* **Garfinkel, Simson.** S57-68, s.l. : Digital Investigation, 2009, Vol. 6 .

21. **Carrier, Brian.** http://www.sleuthkit.org/. [Online] [Cited: 02 02 2012.] http://www.sleuthkit.org/.

22. **Michael Cohen*, Bradley Schatz.** Hash based disk imaging using AFF4. [Online] [Cited: 12 02 2012.] http://www.dfrws.org/2010/proceedings/2010-314.pdf.

23. **Carvey, Harlan.** *Windows Registry Forensics.* 2011. ISBN 1597495808.

24. **Lee, Rob.** USBKEY-Guide. *blogs.sans.org.* [Online] [Cited: 13 01 2012.] http://blogs.sans.org/computer-forensics/files/2009/09/USBKEY-Guide.pdf.

25. **Reninger, Brad.** EMDMgmt Registry Key. [Online] [Cited: 21 02 2012.] http://tech.groups.yahoo.com/group/win4n6/message/3294.

26. **Microsoft.** http://technet.microsoft.com/en-us/magazine/ff356869.aspx. [Online] [Cited: 21 02 2012.]

27. Device Interface Classes for storage devices. *Microsoft Technet.* [Online] [Cited: 21 02 2012.] http://msdn.microsoft.com/en-us/library/windows/hardware/ff541389(v=vs.85).aspx.

28. **Gowdy, Stephen J.** List of USB ids. [Online] [Cited: 20 01 2012.] http://www.linux-usb.org/usb.ids.

29. **Parsonage, Harry.** TheMeaningofLIFE.pdf. [Online] [Cited: 12 03 2012.] http://computerforensics.parsonage.co.uk/downloads/TheMeaningofLIFE.pdf.

30. **clark@hushmail.com.** Security Account Manager. *www.beginningtoseethelight.org.* [Online] [Cited: 02 02 2012.] http://www.beginningtoseethelight.org/ntsecurity/index.php.

31. **Stevens, Didier.** Windows 7 Userassist registry keys. [Online] [Cited: 20 03 2012.] http://intotheboxes.files.wordpress.com/2010/04/intotheboxes_2010_q1.pdf.

32. UserAssist Registry Key 09-8-08.pdf. *www.accessdata.com.* [Online] [Cited: 01 04 2012.] http://accessdata.com/downloads/media/UserAssist%20Registry%20Key%209-8-08.pdf.

33. *A comparative methodology for the reconstruction of digital events using windows restorepoints.* **Yuandong Zhu, Joshua James, Pavel Gladyshev.** 1-6, s.l. : Elsevier, 2009, Vol. 6.

34. **Rowe, Eric.** WiFi related registry keys. [Online] [Cited: 12 02 2012.] http://www.iccyber.org/2009/uploads/trabalhos/20090925/RCMP_Eric_Rowe.pdf.

35. **ieee.** oui.txt. [Online] [Cited: 24 01 2012.] http://standards.ieee.org/develop/regauth/oui/oui.txt.

36. http://en.wikipedia.org/wiki/Wardriving. [Online] [Cited: 29 04 2012.]

37. **Carvey, Harlan.** http://code.google.com/p/regripperplugins/. [Online] [Cited: 24 04 2012.] http://code.google.com/p/regripperplugins/downloads/detail?name=regripperplugins_20120224.zip.

38. Pc Tools Registry Guide. [Online] [Cited: 13 01 2012.] http://www.pctools.com/guides/registry/.

39. *The Windows Registry as a Forensic Resource.* **Carvey, Harlan.** 3, 2005, Vol. 2. Pages 201-5.

40. *Tracking USB storage: Analysis of windows artifacts generated by USB storage devices.* **Cory Altheide, Harlan Carvey.** 2 (Pages 94-100), s.l. : Digital Investigation, 2005, Vol. 2.

Appendix A – Details of data manipulations performed by usbdevices.sh

Sample output

```
1 Serial num/iid  : 200602668009F6B085B3&0
2 Long name       : Disk&Ven_SanDisk&Prod_Cruzer_Edge&Rev_1.10
3 Friendly name   : SanDisk Cruzer Edge USB Device
4 Last in dc 307  : 2012-02-24 09:24:31 (UTC)
5 Last in dc 30d  : 2012-02-24 09:24:32 (UTC)
6 Last in enumusb : 2012-02-29 19:22:13 (UTC) Time may not be device specific
7 Last in dc a5   : 2012-02-24 09:24:31 (UTC)
8 Last test time  : 2011-09-19 13:17:45 (UTC) Mon Sep 19 14:17:45 IST 2011
9 Vendor ID       : 0781 (SanDisk Corp.)
10 Product ID     : 556B
11 Drive\Volume   : E:\ 2011-09-19 13:15:02 (UTC)
12 Volume GUID    : {55782b83-ca94-11e0-b929-0026b9f52bfa}
13 Volume s/n     : 3368049607 (0xC8C05BC7)
.lnk files        :
14 First install  : 2011/09/19 14:14:52.609 (Local Time)
15 Username       : Jenny 2012-02-24 09:24:36 (UTC)

                  : Jacky 2012-02-24 09:24:36 (UTC)
                  : Stephanie 2012-02-24 09:24:36 (UTC)

Serial num/iid    : 1223&0
Long name         : Disk&Ven_USB2.0&Prod_Flash_Disk&Rev_1.00
Friendly name     : USB2.0 Flash Disk USB Device
Last in dc 307    : 2012-02-21 12:27:56 (UTC)
Last in dc 30d    : 2012-02-21 12:27:58 (UTC)
Last in enumusb   : 2012-02-29 19:22:13 (UTC) Time may not be device specific
Last in dc a5     : 2012-02-21 12:27:55 (UTC)
Last test time    : 2012-02-21 12:27:59 (UTC)
Vendor ID         : 1516 (CompUSA)
Product ID        : 8628
Drive\Volume      : E:\ 2012-02-21 12:28:02 (UTC)
Volume GUID       : {bfb12de0-5b3a-11e1-b180-0026b9f52bfa}
Volume s/n        : 3224752956 (0xC035D33C)
16 .lnk files     : 012.lnk
                  : IMG_2614.lnk
                  : markschool.lnk
                  : Removable Disk (E).lnk
First install     : 2012/02/21 12:27:55.700 (Local Time)
Username          : Jacky 2012-02-21 12:52:47 (UTC)

17 ParentID Prefix      : 7&166789fc&0
```

1 Serial num/iid	
Key	HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR (for each device)
Original Data (sampled)	..\Disk&Ven_SanDisk&Prod_Cruzer_Edge&Rev_1.10 ..\Disk&Ven_USB2.0&Prod_Flash_Disk&Rev_1.00 [2012-02-29T19:22:13Z] ..\200602668009F6B085B3&0
Output	200602668009F6B085B3&0

2 Long name	
Key	HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR (for each device)
Original Data	..\Disk&Ven_SanDisk&Prod_Cruzer_Edge&Rev_1.10
Output	Disk&Ven_SanDisk&Prod_Cruzer_Edge&Rev_1.10

3 Friendly name	
Key	HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR (for each device)
Original Data	FriendlyName (REG_SZ) = SanDisk Cruzer Edge USB Device
Output	SanDisk Cruzer Edge USB Device

4 Last in dc 307	
Key	HKLM\SYSTEM\CurrentControlSet\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}(scroll through entries)
Original Data (Sampled)	..\##?#USBSTOR#Disk&Ven_SanDisk&Prod_Cruzer_Edge&Rev_1.10#200602668009F6B085B3&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b} [2012-02-24T09:24:31Z] ..\# DeviceInstance (REG_SZ) = USBSTOR\Disk&Ven_SanDisk&Prod_Cruzer_Edge&Rev_1.10\200602668009F6B085B3&0
Manipulation	Search for serial number/iid match and assign timestamp
Output	: 2012-02-24 09:24:31 (UTC)

5 Last in dc 30d	
Key	HKLM\SYSTEM\CurrentControlSet\Control\DeviceClasses\{53f5630d-b6bf-11d0-94f2-00a0c91efb8b }(scroll through entries)
Original Data (sampled)	..\##?#STORAGE#VOLUME#_??_USBSTOR#DISK&VEN_SANDISK&PROD_CRUZER_EDGE&REV_1.10#200602668009F6B085B3&0#{53F56307-B6BF-11D0-94F2-00A0C91EFB8B}#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b} CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB51D4737F5}\ControlSet001\Control\DeviceClasses\{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}\##?#STORAGE#VOLUME#_??_USBSTOR#DISK&VEN_SANDISK&PROD_CRUZER_EDGE&REV_1.10#200602668009F6B085B3&0#{53F56307-B6BF-11D0-94F2-00A0C91EFB8B}#{53f5630d-b6bf-11d0-94f2-00a0c91efb8b} [2012-02-24T09:24:32Z]
Manipulation	Search for serial number/iid match and assign timestamp
Output	2012-02-24 09:24:32 (UTC)

6 Last in enumusb	
Key	HKLM\SYSTEM\CurrentControlSet\Enum\USB
Original Data (sampled)	..\VID_0781&PID_556B \ControlSet001\Enum\USB\VID_0781&PID_556B\200602668009F6B085B3 [2012-02-29T19:22:13Z]
Manipulation	Search for matching vendor & product id => match serial number/iid key and assign timestamp If several enum keys have similar time stamps warn that it may be global enum event rather than device specific timestamp
Output	2012-02-29 19:22:13 (UTC) Time may not be device specific

7 Last in dc a5	
Key	HKLM\SYSTEM\CurrentControlSet\Control\DeviceClasses\{ a5dcbf10-6530-11d2-901f-00c04fb951ed}(scroll through entries)
Original Data	[2012-02-24T09:24:31Z] ..\# DeviceInstance (REG_SZ) = USB\VID_0781&PID_556B\200602668009F6B085B3
Manipulation	Search for matching vendor & product id => match serial

	number/iid key and assign timestamp
Output	2012-02-24 09:24:31 (UTC)

8 Last test time	
Key	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\EMDMgmt
Original Data	.._??_USBSTOR#Disk&Ven_SanDisk&Prod_Cruzer_Edge&Rev_1.10# 20060266800 9F6B085B3&0#{53f56307-b6bf-11d0-94f2- 00a0c91efb8b}_3368049607 [2011-09-19T13:17:45Z] CacheSizeInMB (REG_DWORD) = 0x00000000 (0) Attributes (REG_BINARY) = 03 00 00 00 2c d1 bd ad 0e b6 8f a1 df 0d 00 00 7f 0a 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 DeviceStatus (REG_DWORD) = 0x00000007 (7) LastTestedTime (REG_QWORD) = 19 35 f3 84 ce 76 cc 01
Manipulation	Reverse endian 01cc76ce84f33519 Convert to Unixtime /10000000 - 11644473600 Pass the unix timestamp through the date command The first time is the timestamp for the attempted test The second time is only reported if the test completes
Output	2011-09-19 13:17:45 (UTC) Mon Sep 19 14:17:45 IST 2011

9 Vendor ID	
Key	HKLM\SYSTEM\CurrentControlSet\Enum\USB
Original Data	..\VID_0781&PID_556B \ControlSet001\Enum\USB\VID_0781&PID_556B\200602668009F6B0 85B3 [2012-02-29T19:22:13Z]
Manipulation	Search for serial number report assigned VID Lookup vendor id in usb.ids usb database maintained at http://www.linux-usb.org/usb.ids to assign vendor name
Output	0781 (SanDisk Corp.)

10 Product ID	
Key	HKLM\SYSTEM\CurrentControlSet\Enum\USB
Original Data	..\VID_0781&PID_556B \ControlSet001\Enum\USB\VID_0781&PID_556B\200602668009F6B0 85B3 [2012-02-29T19:22:13Z]
Output	556B

11 Drive\Volume	
Key	HKLM\SYSTEM\MountedDevices HKLM\SOFTWARE\Microsoft\Windows Portable Devices\Devices (not XP)- scroll through entries
Original Data (sampled)	\DosDevices\E: (REG_BINARY) = 5f 00 3f 00 3f 00 5f 00 55 00 53 00 42 00 53 00 54 00 4f 00 52 00 23 00 44 00 69 00 73 00 6b 00 26 00 56 00 65 00 6e 00 5f 00 4b 00 69 00 6e 00 67 00 73 00 74 00 6f 00 6e 00 26 00 50 00 72 00 6f 00 64 00 5f 00 44 00 61 00 74 00 61 00 54 00 72 00 61 00 76 00 65 00 6c 00 65 00 72 00 5f 00 32 00 2e 00 30 00 26 00 52 00 65 00 76 00 5f 00 23 00 34 00 42 00 34 00 39 00 34 00 45 00 34 00 37 00 35 00 33 00 35 00 34 00 34 00 46 00 34 00 45 00 34 00 38 00 45 00 44 00 35 00 36 00 44 00 45 00 34 00 44 00 26 00 30 00 23 00 7b 00 35 00 33 00 66 00 35 00 36 00 33 00 30 00 37 00 2d 00 62 00 36 00 62 00 66 00 2d 00 31 00 31 00 64 00 30 00 2d 00 39 00 34 00 66 00 32 00 2d 00 30 00 30 00 61 00 30 00 63 00 39 00 31 00 65 00 66 00 62 00 38 00 62 00 7d 00 Devices\Devices\WPDBUSENUMROOT#UMB#2&37C186B&1&STORAGE#VOL

	UME#_??_USBSTOR#DISK&VEN_SANDISK&PROD_CRUZER_EDGE&REV_1.10 #200602668009F6B085B3&0# [2011-09-19T13:15:02Z] FriendlyName (REG_SZ) = E:\
Manipulation	If XP check dosdevice entries for Unicode parentid If vista/7 check dosdevice entries for Unicode serial number Search windows portable devices for serial numbers report friendly name which may be drive letter or volume name + timestamp
Output	E:\ 2011-09-19 13:15:02 (UTC)

12 Volume GUID	
Key	HKLM\SYSTEM\MountedDevices
Original Data (sampled)	\??\Volume{55782b83-ca94-11e0-b929-0026b9f52bfa} (REG_BINARY) = 5f 00 3f 00 3f 00 5f 00 55 00 53 00 42 00 53 00 54 00 4f 00 52 00 23 00 44 00 69 00 73 00 6b 00 26 00 56 00 65 00 6e 00 5f 00 53 00 61 00 6e 00 44 00 69 00 73 00 6b 00 26 00 50 00 72 00 6f 00 64 00 5f 00 43 00 72 00 75 00 7a 00 65 00 72 00 5f 00 45 00 64 00 67 00 65 00 26 00 52 00 65 00 76 00 5f 00 31 00 2e 00 31 00 30 00 23 00 32 00 30 00 30 00 36 00 30 00 32 00 36 00 36 00 38 00 30 00 30 00 39 00 46 00 36 00 42 00 30 00 38 00 35 00 42 00 33 00 26 00 30 00 23 00 7b 00 35 00 33 00 66 00 35 00 36 00 33 00 30 00 37 00 2d 00 62 00 36 00 62 00 66 00 2d 00 31 00 31 00 64 00 30 00 2d 00 39 00 34 00 66 00 32 00 2d 00 30 00 30 00 61 00 30 00 63 00 39 00 31 00 65 00 66 00 62 00 38 00 62 00 7d 00
Manipulation	match unicode serial number in \??\volume entries and assign volume guid
Output	{55782b83-ca94-11e0-b929-0026b9f52bfa}

13 Volume s/n	
Key	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\EMDMgmt
Original Data (sampled)	.._??_USBSTOR#Disk&Ven_SanDisk&Prod_Cruzer_Edge&Rev_1.10# 200602668009F6B085B3&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}_3368049607
Manipulation	Volume s/n is converted to hex to facilitate lnk file matching
Output	: 3368049607 (0xC8C05BC7)

14 First install	
File	setupapi.log (xp) or setupapi.dev.log (vista & win7)

Original Data (sampled)	>>> [Device Install (Hardware initiated) - USBSTOR\Disk&Ven_SanDisk&Prod_Cruzer_Edge&Rev_1.10\200602668009F6B085B3&0]
	>>> Section start 2011/09/19 14:14:52.609
	ump: Creating Install Process: DrvInst.exe 14:14:52.609

Manipulation	Append (local time) for clarity
Output	2011/09/19 14:14:52.609 (Local Time)

15 Username	
Key	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer \MountPoints2 (for each ntuser.dat present)
Original Data (sampled)	..\{55782b83-ca94-11e0-b929-0026b9f52bfa} [2012-02-24T09:24:36Z]

	..\shell
Manipulation	Searches available ntuser.dats. Reports timestamp from all hives with an entry
Output	: Jenny 2012-02-24 09:24:36 (UTC) : Jacky 2012-02-24 09:24:36 (UTC) : Stephanie 2012-02-24 09:24:36 (UTC)

16 .lnk files	
Files	Scroll through all available .lnk files
Original Data (sampled)	Reverse endian hex volume serial number(13 above) => convert .lnk files to hex with xxd Search for serial number report filename for hits
Output	: 012.lnk : IMG_2614.lnk : markschool.lnk : Removable Disk (E).lnk

17 ParentID Prefix (XP only)	
Key	HKLM\SYSTEM\CurrentControlSet\Enum\USBSTOR (for each device)
Original Data	ParentIdPrefix (REG_SZ) = 7&166789fc&0
Output	7&166789fc&0

Appendix B – Details of data manipulations performed by userinfo.sh

Sample userinfo.sh output

```
1 There are 6 users associated with this system (LAURA-PC)

2 Username : Jacky
3  RID                  : 1001 (000003E9)
4  Last logon           : Fri Mar 11 07:40:28 GMT 2011
5  Users Profile Folder : C:\Users\Jacky
6  Users NTUSER.DAT     : NTUSER.DAT.Jacky
7  Last password change : Sat Aug 30 20:29:54 IST 2008
8  Account expires      : Does not expire
9  Account enabled      : Yes
10 Last failed logon    : Wed Jan 4 12:55:33 GMT 2012
11 Number of logons     : 12
12 Member of group(s)   : Administrators
                        : Users


13 There are 8 groups associated with this system (LAURA-PC)

14 Groupname : Users
    RID                  : 00000221
    Number of members    : 5
    Group members        : Jacky (000003E9)
                           ASPNET (000003EB)
                           Family and Friends (000003EC)

15 Username = Unable to associate with a username
   Hive = ../hives/vista1/ntuser.dat.Laura

16 User Profile subfolders:
   %USERPROFILE% = Unrecorded
   AppData = %USERPROFILE%\AppData\Roaming
   Cache = %USERPROFILE%\AppData\Local\Microsoft\Windows\Temporary Internet
   Files
   Cookies = %USERPROFILE%\AppData\Roaming\Microsoft\Windows\Cookies

17 RECENTDOCS: 2011-09-26 10:35:43 (UTC)
.BMP MRU List - 2009-06-13 13:25:47 (UTC)
 20080317-185344.BMP

.contact MRU List - 2008-09-30 16:57:29 (UTC)
 Laura.contact

.doc MRU List - 2009-11-17 12:11:30 (UTC)
 alphfictest.doc
 Alphabetised Fiction.doc

.zip MRU List - 2009-07-26 14:38:10 (UTC)
 T-115980-Inkheart (2008).zip
 T-78310-inkdeath.zip

Folder MRU List - 2011-09-26 10:35:43 (UTC)
 iTunes
 Web Products
 Previous iTunes Libraries

18 MEDIAPLAYER MRU - 2011-09-26 10:20:11 (UTC)
File0 C:\Users\Laura\Music\iTunes\iTunes Music\Podcasts\ABBA - Mama Mia.mp3
```

```
File1 C:\Users\Laura\Documents\ADELE - SOMEONE LIKE YOU w_ OFFICIAL
LYRICS.flv
File2 C:\Users\Laura\Documents\Jessie J - Price Tag ft. B.o.B..flv

19 TYPED URLs - 2011-09-28 18:24:32 (UTC)
 1
C:\Users\Jackyle.com/search?q=everything+left+handed&rls=com.microsoft:en-
ie:IE-Address&ie=UTF-8&oe=UTF-8&sourceid=ie7&rlz=1I7GGLL_en-GB
 2 F:\laura\itunes
 3 C:\Users\Laura\Documents\LimeWire\Saved
 4 C:\Windows\System32
 5 http://www.cnbc.com/


20 RUN MRU LIST - 2011-07-14 08:11:43 (UTC)
 cmd
 hdwwiz


21 USERASSIST EXPLORER -2011-09-28 18:24:30 (UTC)

 UEME_CTLSESSION (204)

22 USERASSIST DESKTOP - 2011-10-17 16:07:16 (UTC)

 UEME_CTLSESSION (206)
 UEME_RUNPIDL:%csidl23%\Windows Live.lnk (18) Sat Aug 30 16:26:24 IST 2008
 UEME_CTLCUACount:ctor (2)
 UEME_RUNPATH (1906) Mon Oct 17 17:01:30 IST 2011
  (254) Mon Oct 17 17:01:30 IST 2011
 UEME_RUNPIDL (313) Wed Sep 28 19:24:39 IST 2011
  (77) Tue Sep 27 19:49:33 IST 2011
 UEME_RUNPATH:iTunes.lnk (64) Fri May 20 20:45:36 IST 2011
 UEME_RUNPATH:C:\Program Files\iTunes\iTunes.exe (161) Tue Sep 27 19:18:17
IST 2011
  (19) Sun Sep 14 10:03:36 IST 2008
 UEME_RUNPATH:U.B. Funkeys.lnk (5) Tue Sep  2 18:27:52 IST 2008
 UEME_RUNPATH:LimeWire 4.18.6.lnk (2) Sun Oct  4 15:48:13 IST 2009
  (21) Mon Sep 29 18:32:56 IST 2008
 UEME_RUNPATH:Windows Live.lnk (1) Sun Jul 10 10:08:47 IST 2011
  (6) Thu Nov 26 20:51:28 GMT 2009
 UEME_RUNPIDL:%csidl23% (2) Sun Aug 23 12:02:16 IST 2009
 UEME_RUNPATH:C:\Program Files\Internet Explorer\iexplore.exe (219) Tue Sep
27 19:49:33 IST 2011
  (10) Sat Nov 22 18:06:41 GMT 2008
 UEME_RUNPATH:C:\Windows\system32\rundll32.exe (1) Sun Sep 25 19:36:16 IST
2011
  (99) Sun May 24 11:02:07 IST 2009
 UEME_RUNPATH:Internet Explorer.lnk (60) Tue Aug 18 17:38:04 IST 2009
  (6) Mon Apr  4 21:15:10 IST 2011
 UEME_RUNPATH:E-mail - Shortcut.lnk (2) Sun Jun 28 19:55:05 IST 2009


23 USER SPECIFIC AUTORUNS -
 "C:\Program Files\Dell Support Center\bin\sprtcmd.exe" /P
DellSupportCenter
 C:\Windows\ehome\ehTray.exe
 rundll32.exe "C:\Users\Laura\AppData\Roaming\xjoyifid.dll",autorun
 "C:\Program Files\Windows Live\Messenger\msnmsgr.exe" /background

24 REMOTE DESKTOP TERMINAL SERVERS
```

```
  CCICONNECT.UCD.IE = CCI\jafox (2011-11-18 08:22:57 (UTC))
  ucdcci02.ucd.ie = CCI\fjacky (2012-01-08 20:15:25 (UTC))
   (Default since 2012-03-30 12:55:37 (UTC))
  ucdcci11.ucd.ie = CCI\fjacky (2011-01-09 21:46:20 (UTC))


25 SYSTEMS SEEN BY NETWORK BROWSER - 2004-08-26 15:07:12 (UTC)
  4.12.220.254  (m1200)
  TOWER  (Tower)
  TOWER2
  ECSAP_LAB_SRVR
  ANDREWS-1

26 RECENTLY MAPPED NETWORK DRIVES - 2012-04-02 17:08:26 (UTC)

  \\JACKY-PC\Users\jacky\Documents
  \\JACKY-PC\Users\Public
  \\JACKY-PC\Users\jacky\Documents\california laura

27 RECONNECT AT LOGIN NETWORK DRIVES
  Z: = \\JACKY-PC\Users\jacky\Documents - Username = (no data)

28 PRINTERS
  ,,JACKY-PC,HP LaserJet 1200 Series PCL 5
  Auto HP LaserJet 2100 PCL6 on ANDREWS-1

29 OPEN/SAVE MRUs


* Most Recently Used List - 2011-09-27 11:39:53 (UTC)
 itunes
 iTunes Library.itl
 ABBA - Mama Mia.mp3

 iTunes Library 2008-04-14.itl
 Saved
 Beyonce - If I Were A Boy.mp3
 dldwutil.dll

dll Most Recently Used List - 2011-08-11 18:12:35 (UTC)
 dldwutil.dll

doc Most Recently Used List - 2009-11-17 12:00:06 (UTC)
 alphfictest.doc
 alphfictest.doc
 Alphabetised Fiction.doc
```

1 There are ? users associated with this system (?)	
Key	HKLM\SAM\Domains\Account\Users\Names HKLM\SYSTEM\CurrentControlSet\Control\ComputerName\ComputerName
Original Data (sampled)	..\Administrator ..\Jacky ..\Laura ComputerName (REG_SZ) = LAURA-PC
Manipulation	Count the number of subkeys
Output	6 users (LAURA-PC)

2 Username	
Key	HKLM\SAM\Domains\Account\Users\Names
Original Data	..\Jacky
Output	Jacky

3 RID (relative identifier)	
Key	HKLM\SAM\Domains\Account\Users\Names\Jacky (for each name)
Original Data	(Default) (REG_1001) = (no data)
Manipulation	Convert Rid => hex => mask preceding 0 to make length 8
Output	1001 (000003E9)

4 Last logon	
Key	HKLM\SAM\Domains\Account\Users\Names\000003E9 (for each rid)
Original Data	F (REG_BINARY) = 02 00 01 00 00 00 00 00 8a 8a 8e 97 bf df cb 01 00 00 00 00 00 00 00 00 71 d2 be c7 d6 0a c9 01 ff ff ff ff ff ff ff 7f d0 85 1a 25 e0 ca cc 01 e9 03 00 00 01 02 00 00 10 02 00 00 00 00 00 00 06 00 0c 00 01 00 00 00 00 00 f6 76 08 f5 f6 01
Manipulation	Reverse endian 01c90ad6c7bed271 Convert to Unixtime /10000000 - 11644473600 Pass the unix timestamp through the date command If value = 0 then report “No logon recorded”
Output	Fri Mar 11 07:40:28 GMT 2011

5 Users Profile Folder	
Key	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Profilelist (for each sid)
Original Data (sampled)	..\S-1-5-21-3060861534-1199086942-1648537790-1000 ..\S-1-5-21-3060861534-1199086942-1648537790-1001 ProfileImagePath (REG_EXPAND_SZ) = C:\Users\Jacky
Manipulation	Retrieve all sids Match rid of sid with SAM stored rid
Output	C:\Users\Jacky

6 Users NTUSER.DAT	
Key	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Profilelist (for each sid)
Original Data (sampled)	..\S-1-5-21-3060861534-1199086942-1648537790-1001 ProfileImagePath (REG_EXPAND_SZ) = C:\Users\Jacky
Manipulation	If getreg.sh has been used to acquire the registry files the user hive for this user will be....
Output	NTUSER.DAT.Jacky

7 Last password change	
Key	HKLM\SAM\Domains\Account\Users\Names\000003E9 (for each rid)
Original Data	F (REG_BINARY) = 02 00 01 00 00 00 00 00 8a 8a 8e 97 bf df cb 01 00 00 00 00 00 00 00 00 71 d2 be c7 d6 0a c9 01 ff ff ff ff ff ff ff 7f d0 85 1a 25 e0 ca cc 01 e9 03 00 00 01 02 00 00 10 02 00 00 00 00 00 00 06 00 0c 00 01 00 00 00 00 00 f6 76 08 f5 f6 01
Manipulation	Reverse endian 01c90ad6c7bed271 Convert to Unixtime /10000000 - 11644473600 Pass the unix timestamp through the date command 0000000000000000 => report “Never”
Output	Sat Aug 30 20:29:54 IST 2008

8 Account expires	
Key	HKLM\SAM\Domains\Account\Users\Names\000003E9 (for each rid)
Original Data	F (REG_BINARY) = 02 00 01 00 00 00 00 00 8a 8a 8e 97 bf df cb 01 00 00 00 00 00 00 00 00 71 d2 be c7 d6 0a c9 01 ff ff ff ff ff ff ff 7f d0 85 1a 25 e0 ca cc 01 e9 03 00 00 01 02 00 00 10 02 00 00 00 00 00 00 06 00 0c 00 01 00 00 00 00 00 f6 76 08 f5 f6 01
Manipulation	Reverse endian 7fffffffffffffff Convert to Unixtime /10000000 - 11644473600 Pass the unix timestamp through the date command 7fffffffffffffff => report "Does not expire" 0000000000000000 => report "No expiry set"
Output	Does not expire

9 Account Enabled	
Key	HKLM\SAM\Domains\Account\Users\Names\000003E9 (for each rid)
Original Data	F (REG_BINARY) = 02 00 01 00 00 00 00 00 8a 8a 8e 97 bf df cb 01 00 00 00 00 00 00 00 00 71 d2 be c7 d6 0a c9 01 ff ff ff ff ff ff ff 7f d0 85 1a 25 e0 ca cc 01 e9 03 00 00 01 02 00 00 10 02 00 00 00 00 00 00 06 00 0c 00 01 00 00 00 00 00 f6 76 08 f5 f6 01
Manipulation	0,2,6,8,a,c,e,A,C,E => Yes 1,3,5,7,b,d,f,B,D,F => No 4 => Unknown
Output	Yes

10 Failed logon	
Key	HKLM\SAM\Domains\Account\Users\Names\000003E9 (for each rid)
Original Data	F (REG_BINARY) = 02 00 01 00 00 00 00 00 8a 8a 8e 97 bf df cb 01 00 00 00 00 00 00 00 00 71 d2 be c7 d6 0a c9 01 ff ff ff ff ff ff ff 7f d0 85 1a 25 e0 ca cc 01 e9 03 00 00 01 02 00 00 10 02 00 00 00 00 00 00 06 00 0c 00 01 00 00 00 00 00 f6 76 08 f5 f6 01
Manipulation	Reverse endian 01cccae0251a85d0 Convert to Unixtime /10000000 - 11644473600 Pass the unix timestamp through the date command 0000000000000000 => report "Never"
Output	Wed Jan 4 12:55:33 GMT 2012

11 Number of Logons	
Key	HKLM\SAM\Domains\Account\Users\Names\000003E9 (for each rid)
Original Data	F (REG_BINARY) = 02 00 01 00 00 00 00 00 8a 8a 8e 97 bf df cb 01 00 00 00 00 00 00 00 00 71 d2 be c7 d6 0a c9 01 ff ff ff ff ff ff ff 7f d0 85 1a 25 e0 ca cc 01 e9 03 00 00 01 02 00 00 10 02 00 00 00 00 00 00 06 00 0c 00 01 00 00 00 00 00 f6 76 08 f5 f6 01
Manipulation	Convert to decimal
Output	12

12 Member of group(s)	
Key	HKLM\SAM\SAM\Domains\Builtin\Aliases\Names HKLM\SAM\SAM\Domains\Builtin\Aliases (for each group rid)
Original Data (sampled)	..\Administrators (Default) (REG_544) = (no data) C (REG_BINARY) = 20 02 00 00 00 00 00 00 98 00 00 00 02 00 01 00 98 00 00 00 1c 00 00 00 00 00 00 00 b4 00 00 00 .... be b0 42 62 f4 01 00 00 01 05 00 00 00 00 00 05 15 00 00 00 5e 0a 71 b6 5e 9d 78 47 be b0 42 62 e8 03 00 00 01 05 00 00 00 00 00 05 15 00 00 00 5e 0a 71 b6 5e 9d 78 47 be b0 42 62 e9 03 00 00
Manipulation	Scroll through group names => identify decimal rid => hex => Retrieve "C" value for rid => identify user sids => report any group memberships
Output	Administrators Users

13 There are ? groups associated with this system (?)	
Key	HKLM\SAM\SAM\Domains\Builtin\Aliases\Names HKLM\SYSTEM\CurrentControlSet\Control\ComputerName\ComputerName
Original Data (sampled)	..\Administrators ..\Users ComputerName (REG_SZ) = LAURA-PC
Manipulation	Count the number of subkeys
Output	8 groups (LAURA-PC)

14 Groupname RID Number of members Group members	
Key	HKLM\SAM\SAM\Domains\Builtin\Aliases\Names HKLM\SAM\SAM\Domains\Builtin\Aliases\00000221 (for each group rid)
Original Data (sampled)	..\Users (Default) (REG_545) = (no data) C (REG_BINARY) = 21 02 00 00 00 00 00 00 c8 00 00 00 02 00 01 00 c8 00 00 00 0a 00 00 00 00 00 00 00 d4 00 00 00 d6 00 00 00 00 00 00 00 ac 01 00 00 6c 00 00 00 05 00 00 00 01 00 14 80 a8 00 00 00 b8 00 00 00 14 00 00 00 44 00 00 00 02 00 30 00 02 00 00 00 02 c0 14 00 13 00 05 01 01 ....05 04 00 00 00 01 01 00 00 00 00 00 05 0b 00 00 00 01 05 00 00 00 00 00 05 15 00 00 00 5e 0a 71 b6 5e 9d 78 47 be b0 42 62 e9 03 00 00 01 05 00 00 00 00 00 05 15 00 00 00 5e 0a 71 b6 5e 9d 78 47 be b0 42 62 eb 03 00 00 01 05 00 00 00 00 00 05 15 00 00 00 5e 0a 71 b6 5e 9d 78 47 be b0 42 62 ec 03 00 00
Manipulation	Scroll through group names => identify decimal rid => hex => Retrieve "C" value for rid Identify number of members => reverse endian => decimal Identify user sids => lookup & report usernames
Output	Users, 00000221, 5, Jacky (000003E9), ASPNET (000003EB), Family and Friends (000003EC)

15 Username Hive	
Key	HKCU\Software\Microsoft\Windows NT\CurrentVersion\Profilelist (for each sid)
Original Data (sampled)	..\S-1-5-21-3060861534-1199086942-1648537790-1001 ProfileImagePath (REG_EXPAND_SZ) = C:\Users\Jacky
Manipulation	Generates a list of ntuser.dats in the present working directory. If getreg.sh has been used to acquire the registry files the user hive for each can usually be matched with a username
Output	Unable to associate with a username ../hives/vistal/ntuser.dat.Laura

16 User Profile subfolders	
Key	HKCU\Software\Microsoft\Windows NT\CurrentVersion\Explorer\User Shell Folders
Original Data (sampled)	AppData (REG_EXPAND_SZ) = %USERPROFILE%\AppData\Roaming
Output	AppData = %USERPROFILE%\AppData\Roaming

17 RECENTDOCS	
Key	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs (scroll through each extension)
Original Data (sampled)	[2009-11-17T12:11:30Z] MRUListEx (REG_BINARY) = 01 00 00 00 00 00 00 00 ff ff ff ff 0 (REG_BINARY) = 41 00 6c 00 70 00 68 00 61 00 62 00 65 00 74 00 69 00 73 00 65 00 64 00 20 00 46 00 69 00 63 00 74 00 69 00 6f 00 6e 00 2e 00 64 00 6f 00 63 00 00 00 82 00 32 00 00 00 00 00 00 00 00 00 00 00 41 6c 70 68 61 62 65 74 69 73 65 64 20 46 69 63 74 69 6f 6e 2e 6c 6e 6b 00 00 5a 00 07 00 04 00 ef be 00 00 00 00 00 00 00 00 26 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 41 00 6c 00 70 00 68 00 61 00 62 00 65 00 74 00 69 00 73 00 65 00 64 00 20 00 46 00 69 00 63 00 74 00 69 00 6f 00 6e 00 2e 00 6c 00 6e 00 6b 00 00 00 28 00 00 00
Manipulation	Reverse endian each 32 bit number in MRUlist => 00000001, 00000000, ffffffff => 1,0,end Retrieve entries in order and translate zero terminated Unicode name 4100 6c00 7000 6800 => 0041 006c 0070 0068 => Alph...etc
Output	.doc MRU List - 2009-11-17 12:11:30 (UTC) alphfictest.doc Alphabetised Fiction.doc

18 MEDIAPLAYER MRU	
Key	HKCU\Software\Microsoft\MediaPlayer\Player\RecentFileList
Original Data	[2011-09-26T10:20:11Z] File0 (REG_SZ) = C:\Users\Laura\Music\iTunes\iTunes Music\Podcasts\ABBA - Mama Mia.mp3 File1 (REG_SZ) = C:\Users\Laura\Documents\ADELE - SOMEONE LIKE YOU w_ OFFICIAL LYRICS.flv
Output	2011-09-26 10:20:11 (UTC) File0 C:\Users\Laura\Music\iTunes\iTunes Music\Podcasts\ABBA - Mama Mia.mp3 File1 C:\Users\Laura\Documents\ADELE - SOMEONE LIKE YOU w_ OFFICIAL LYRICS.flv

19 TYPED URLs	
Key	HKCU\Software\Microsoft\Internet Explorer\TypedURLs
Original Data	[2011-09-28T18:24:32Z] url1 (REG_SZ) = C:\Users\Jackyle.com/search?q=everything+left+handed&rls=com.microsoft:en-ie:IE-Address&ie=UTF-8&oe=UTF-8&sourceid=ie7&rlz=1I7GGLL_en-GB url2 (REG_SZ) = F:\laura\itunes
Output	2011-09-28 18:24:32 (UTC) 1 C:\Users\Jackyle.com/search?q=everything+left +handed&rls=com.microsoft:en-ie:IE-Address&ie=UTF-8&oe=UTF-8&sourceid=ie7&rlz=1I7GGLL_en-GB 2 F:\laura\itunes

20 RUN MRU LIST	
Key	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU
Original Data	[2011-07-14T08:11:43Z] a (REG_SZ) = hdwwiz\1 MRUList (REG_SZ) = ba b (REG_SZ) = cmd\1
Manipulation	Read the MRUlist in order and report the associated commands
Output	2011-07-14 08:11:43 (UTC) cmd hdwwiz

21 USERASSIST explorer (XP & Vista)	
Key	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{5E6AB780-7743-11CF-A12B-00AA004AE837}\Count
Original Data	[2011-09-28T18:24:30Z] HRZR_PGYFRFFVBA (REG_BINARY) = 5e f0 63 0e cc 00 00 00
Manipulation	Do ROT13 decryption HRZR_PGYFRFFVBA => UEME_CTLSESSION Determine time(if recorded)reverse endian 01c90ab4c33d5f20 Convert to Unixtime /10000000 - 11644473600 Pass the unix timestamp through the date command Determine usage count cc 00 00 00 => 000000cc => 204 Subtract 5 from count if entry = UEME_RUNPIDL, UEME_RUNPATH or UEME_RUNCPL
Output	USERASSIST EXPLORER -2011-09-28 18:24:30 (UTC) □ UEME_CTLSESSION (204)

21a USERASSIST explorer (Windows 7)	
Key	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}\Count
Original Data (sampled)	[2012-01-30T13:11:12Z] HRZR_PGYFRFFVBA (REG_BINARY) = 1d 00 00 00 00 00 00 00 2d 00 00 00 b0 46 82 05 00 00 00 00 03 00 00 00 97 70 43 05 7b 00 37 00 43 00 35 00 41 00 34 00 30 00 45 00 46 00 2d 00 41 00 30 00 46 00 42 00 2d 00 34 00 42 00..... Zvpebfbsg.Jvaqbjf.FgvpxlAbgrf (REG_BINARY) = 1d 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 08 00 00 00 5e 47 09 4d 65 63 ca 01 00 00 00 00
Manipulation	Do ROT13 decryption HRZR_PGYFRFFVBA => UEME_CTLSESSION Determine time(if recorded)reverse endian 01ca63654d09475e Convert to Unixtime /10000000 - 11644473600 Pass the unix timestamp through the date command Determine usage count 00 00 00 00 => 00000000 => 0 Determine focus count 2d 00 00 00 => 0000002d => 45 Usage count appears to be for current month only
Output	USERASSIST EXPLORER -2012-01-30 13:11:12 (UTC) UEME_CTLSESSION (0)(45) □Microsoft.Windows.StickyNotes (0)(0) Thu Nov 12 06:56:46 GMT 2009

22 USERASSIST DESKTOP (XP & Vista)	
Key	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75048700-EF1F-11D0-9888-006097DEACF9}\Count
Original Data	[2011-10-17T16:07:16Z] HRZR_PGYFRFFVBA (REG_BINARY) = 36 67 64 0e ce 00 00 00 HRZR_EHACVQY:%pfvqy23%\Jvaqbjf Yvir.yax (REG_BINARY) = 00 00 00 00 17 00 00 00 20 5f 3d c3 b4 0a c9 01 HRZR_PGYPHNPbhag:pgbe (REG_BINARY) = 00 00 00 00 02 00 00 00 00 00 00 00 00 00 00 00 HRZR_EHACNGU (REG_BINARY) = cd 00 00 00 77 07 00 00 e0 5e 8b 08 e6 8c cc 01 HRZR_EHACNGU:{7S0P4457-8R64-491O-8Q7O-991504365Q1R} (REG_BINARY) = cd 00 00 00 fe 00 00 00 e0 5e 8b 08 e6 8c cc 01
Manipulation	Do ROT13 decryption HRZR_EHACVQY... => UEME_RUNPIDL... Determine time(if recorded)reverse endian 01c90ab4c33d5f20 Convert to Unixtime /10000000 - 11644473600 Pass the unix timestamp through the date command Determine usage count 02 00 00 00 => 00000002 => 2 Subtract 5 from count unless entry = UEME_RUNPIDL & count < 6 (no actual usage eg. Mouseover =2) note no time associated if <6
Output	2011-10-17 16:07:16 (UTC) UEME_CTLSESSION (206) UEME_RUNPIDL:%csidl23%\Windows Live.lnk (18) Sat Aug 30 16:26:24 IST 2008 □UEME_CTLCUACount:ctor (2)

22a USERASSIST DESKTOP (Windows 7)	
Key	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{F4E57C4B-2036-45F0-A9AB-443BCFE33D9F}\Count
Original Data	HRZR_PGYFRFFVBA (REG_BINARY) = 1d 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 {N77S5Q77-2R2O-44P3-N6N2-NON601054N51}\Npprffbevrf\Npprffvovyvgl\Zntavsl.yax (REG_BINARY) = 1d 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 08 00 00 00 5e 47 09 4d 65 63 ca 01 00 00 00 00
Manipulation	Do ROT13 decryption HRZR_PGYFRFFVBA => UEME_CTLSESSION Determine time(if recorded)reverse endian 01ca63654d09475e Convert to Unixtime /10000000 - 11644473600 Pass the unix timestamp through the date command Determine usage count 00 00 00 00 => 00000000 => 0 Determine focus count 00 00 00 00 => 00000000 => 0 Usage count is for current month
Output	USERASSIST DESKTOP - 2012-01-16 10:04:37 (UTC) UEME_CTLSESSION (0)(0) □\Accessories\Accessibility\Magnify.lnk (0)(0) Thu Nov 12 06:56:46 GMT 2009

23 USER SPECIFIC AUTORUNS	
Key	HKCU\Software\Microsoft\Windows\CurrentVersion\Run
Original Data (sampled)	DellSupportCenter (REG_SZ) = "C:\Program Files\Dell Support Center\bin\sprtcmd.exe" /P DellSupportCenter ehTray.exe (REG_SZ) = C:\Windows\ehome\ehTray.exe xjoyifid (REG_SZ) = rundll32.exe
Output	"C:\Program Files\Dell Support Center\bin\sprtcmd.exe" /P DellSupportCenter C:\Windows\ehome\ehTray.exe rundll32.exe Files\Windows Live\Messenger\msnmsgr.exe" /background

24 REMOTE DESKTOP TERMINAL SERVERS	
Key	HKCU\Software\Microsoft\Terminal Server Client\Servers (scroll through each server) HKCU\Software\Microsoft\Terminal Server Client\Default
Original Data samples	[2011-11-18T08:22:48Z] ..\CCICONNECT.UCD.IE ..\ucdcci02.ucd.ie ..\ucdcci11.ucd.ie [2011-11-18T08:22:57Z] CertHash (REG_BINARY) = 6e 34 f1 b1 33 d7 fa d9 49 a0 bd 54 05 dd 9b 42 11 fc b0 d9 UsernameHint (REG_SZ) = CCI\jafox [2012-03-30T12:55:37Z] MRU0 (REG_SZ) = ucdcci02.ucd.ie MRU1 (REG_SZ) = cciconnect.ucd.ie MRU2 (REG_SZ) = ucdcci11.ucd.ie

Output	CCICONNECT.UCD.IE = CCI\jafox (2011-11-18 08:22:57 (UTC))
	ucdcci02.ucd.ie = CCI\fjacky (2012-01-08 20:15:25 (UTC))
	(Default since 2012-03-30 12:55:37 (UTC))
	ucdcci11.ucd.ie = CCI\fjacky (2011-01-09 21:46:20 (UTC))

25 SYSTEMS SEEN BY NETWORK BROWSER	
Key	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComputerDescriptions
Original Data (sampled)	[2004-08-26T15:07:12Z] 4.12.220.254 (REG_SZ) = m1200 TOWER (REG_SZ) = Tower TOWER2 (REG_SZ) = (no data) ANDREWS-1 (REG_SZ) = (no data)

Output	2004-08-26 15:07:12 (UTC)
	  4.12.220.254 (m1200)
	  TOWER (Tower)
	  TOWER2
	□ ANDREWS-1

26 RECENTLY MAPPED NETWORK DRIVES	
Key	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Map Network Drive MRU
Original Data	[2012-04-02T17:08:26Z] a (REG_SZ) = \\JACKY-PC\Users\jacky\Documents\california laura MRUList (REG_SZ) = bca b (REG_SZ) = \\JACKY-PC\Users\jacky\Documents c (REG_SZ) = \\JACKY-PC\Users\Public
Manipulation	Read the MRUlist in order and report the associated mappings
Output	2012-04-02 17:08:26 (UTC) \\JACKY-PC\Users\jacky\Documents \\JACKY-PC\Users\Public □\\JACKY-PC\Users\jacky\Documents\california laura

27 RECONNECT AT LOGIN NETWORK DRIVES	
Key	HKCU\Network (scroll through any letters)
Original Data (sampled)	..\Z RemotePath (REG_SZ) = \\JACKY-PC\Users\jacky\Documents UserName (REG_SZ) = (no data)
Output	Z: = \\JACKY-PC\Users\jacky\Documents - Username = (no data)

28 PRINTERS	
Key	HKCU\Printers\Connections HKCU\Printers\DevModes2
Original Data (sampled)	..\,,JACKY-PC,HP LaserJet 1200 Series PCL 5 Auto HP LaserJet 2100 PCL6 on ANDREWS-1 (REG_BINARY) = 5c 00 5c 00 41 00 4e 00 44 00 52 00 45 00 57 00 53 00 2d 00 31 00 5c 00 48 00 50 00 20 00...etc
Output	,,JACKY-PC,HP LaserJet 1200 Series PCL 5 □ Auto HP LaserJet 2100 PCL6 on ANDREWS-1

29 OPEN/SAVE MRUs (Vista, Windows 7)	
Key	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePidlMRU (scroll through each extension)
Original Data (sampled)	[2011-08-11T18:12:35Z] ..* ..\dll ..\doc [2011-09-27T11:39:53Z] MRUListEx (REG_BINARY) = 01 00 00 00 13 00 00 00 00 00 00 00 12 00 00 00 11 00 00 00 10 00 00 00 0e 00 00 00 0d 00 00 00 0c 00 00 00 0b 00 00 00 04 00 00 00 0f 00 00 00 0a 00 00 00 09 00 00 00 08 00 00 00 07 00 00 00 06 00 00 00 05 00 00 00 03 00 00 00 02 00 00 00 ff ff ff ff 1 (REG_BINARY) = 14 00 1f 50 e0 4f d0 20 ea 3a 69 10 a2 d8 08 00 2b 30 30 9d 19 00 2f 46 3a 5c 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 48 00 31 00 00 00 00 00 3b 3f 05 5c 10 00 6c 61 75 72 61 00 34 00 07 00 04 00 ef be 3b 3f 28 5c 3a 3f 00 b8 26 00 00 00 c0 ca 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 6c 00 61 00 75 00 72 00 61 00 00 00 14 00 4c 00 31 00 00 00 00 00 3b 3f 11 5c 10 00 69 74 75 6e 65 73 00 00 36 00 07 00 04 00 ef be 3b 3f 28 5c 3a 3f 00 b8 26 00 00 00 40 00 95 07 00 00 00 00 00 00 00 00 00 00 00 00 00 00 69 00 74 00 75 00 6e 00 65 00 73 00 00 00 16 00 00 00
Manipulation	Read the MRUlist in order Report the last unicode string that’s ending can be masked with 00 ?? 00 ?? 00 00 00 ?? - where ?? match [1-9a-fA-F][0-9a-fA-F] Translate strings with xxd –r -p
Output	* Most Recently Used List - 2011-09-27 11:39:53 (UTC) itunes iTunes Library.itl ABBA - Mama Mia.mp3 iTunes Library 2008-04-14.itl □Saved

29a OPEN/SAVE MRUs (XP)	
Key	HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU (scroll through each extension)
Original Data (sampled)	[2004-08-25T15:50:26Z] ..* ..\exe $$$PROTO.HIV\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU\* [2004-08-27T15:18:05Z] a (REG_SZ) = C:\Documents and Settings\Mr. Evil\Desktop\lalsetup250.exe MRUList (REG_SZ) = cdba b (REG_SZ) = C:\Documents and Settings\Mr. Evil\Desktop\netstumblerinstaller_0_4_0.exe c (REG_SZ) = C:\Documents and Settings\Mr. Evil\Desktop\ethereal-setup-0.10.6.exe d (REG_SZ) = C:\Documents and Settings\Mr. Evil\Desktop\WinPcap_3_01_a.exe
Manipulation	Read the MRUlist in order and report the associated file

Output	* Most Recently Used List - 2004-08-27 15:18:05 (UTC)
	  C:\Documents and Settings\Mr. Evil\Desktop\ethereal-setup-0.10.6.exe
	   C:\Documents and Settings\Mr. Evil\Desktop\WinPcap_3_01_a.exe

Appendix C - Details of data manipulations performed by systeminfo.sh

Sample systeminfo.sh output

```
1  Current Control Set       : 001
2  Registered Organisation   : (no data)
3  Registered Owner          : UCD
4  Computer Name             : UCD-083B1381901
5  Product serial number     : 76487-018-3066024-22297
6  Product Name              : Microsoft Windows XP Service Pack 3
7  Current Version           : 5.1
8  System Root               : C:\WINDOWS
9  Installation Date         : Fri Feb 19 20:26:20 GMT 2010
10 Last logged Shutdown Time : Fri Apr 1 05:27:19 IST 2011
11 Last user logged in       : student
12 System Directory          : %SystemRoot%\system32
13 Drive letters             : C: A: D: E: G:
14 Daylight savings Timezone : Pacific Daylight Time (Bias -60 Minutes)
15 Standard Timezone         : Pacific Standard Time (Bias +0 Minutes)
16 Timezone bias             : UTC +480 Minutes
17 Current time bias         : UTC +420 Minutes
18 Network time protocol is  : synchronised
19 Timezone last  updated    : 2011-04-01 04:20:15 (UTC)
20 Daylight saving starts on Sunday in the 2nd week of March at 02:00:00:00
21 Standard time starts on Sunday in the 1st week of November at 02:00:00:00

22 SYSTEM WIDE AUTORUNS
 "C:\Program Files\QuickTime\QTTask.exe" -atboottime
 "C:\Program Files\iTunes\iTunesHelper.exe"
 C:\Windows\system32\igfxtray.exe
 C:\Windows\system32\hkcmd.exe
 C:\Windows\system32\igfxpers.exe

23 INSTALLED APPLICATIONS (Uninstall)
 Adobe Flash Player 10 ActiveX --- 2011-04-01 04:39:27 (UTC)
 Connection Manager --- 2010-02-19 20:09:15 (UTC)
 DirectAnimation --- 2010-02-19 20:17:49 (UTC)
 DirectDrawEx --- 2010-02-19 20:17:41 (UTC)
 DXM_Runtime --- 2010-02-19 20:20:35 (UTC)
 Fontcore --- 2010-02-19 20:17:41 (UTC)
 ICW --- 2010-02-19 20:17:49 (UTC)
 IDNMitigationAPIs --- 2011-04-01 04:06:18 (UTC)
 IE40 --- 2010-02-19 20:17:41 (UTC)
 IE4Data --- 2010-02-19 20:17:41 (UTC)
 IE5BAKEX --- 2010-02-19 20:17:41 (UTC)
 ie7 --- 2011-04-01 04:06:18 (UTC)
 Windows Internet Explorer 8 --- 2011-04-01 04:06:18 (UTC)
 IEData --- 2010-02-19 20:17:41 (UTC)
 MobileOptionPack --- 2010-02-19 20:17:41 (UTC)
 Mozilla Firefox 4.0 (x86 en-US) --- 2011-04-01 01:10:44 (UTC)
 MPlayer2 --- 2011-04-01 04:19:15 (UTC)
 NetMeeting --- 2010-02-19 20:17:49 (UTC)
 NLSDownlevelMapping --- 2011-04-01 04:06:18 (UTC)
 Opera 11.01 --- 2011-04-01 01:15:27 (UTC)
 OutlookExpress --- 2010-02-19 20:17:49 (UTC)
 PCHealth --- 2010-02-19 20:17:58 (UTC)
 SchedulingAgent --- 2010-02-19 20:17:41 (UTC)
 TrueCrypt --- 2011-04-01 01:36:31 (UTC)
 Windows XP - Software Updates --- 2011-04-01 04:32:31 (UTC)
```

```
GIMP 2.6.11 --- 2011-04-01 01:12:36 (UTC)
WPT Poker --- 2011-04-01 04:43:02 (UTC)
WebFldrs XP --- 2010-02-19 20:30:19 (UTC)
Data Stash --- 2011-04-01 02:43:02 (UTC)
```

1 Current Control Set	
Key	HKLM\SYSTEM\Select
Original Data	Current (REG_DWORD) = 0x00000001 (1)
Output	001

2 Registered Organisation	
Key	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion
Original Data	RegisteredOrganisation (REG_SZ) = (no data)
Output	(no data)

3 Registered Owner	
Key	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion
Original Data	RegisteredOwner (REG_SZ) = UCD
Output	UCD

4 Computer Name	
Key	HKLM\SYSTEM\CurrentControlSet\Control\ComputerName\ComputerName
Original Data	ComputerName (REG_SZ) = UCD-083B1381901
Output	UCD-083B1381901

5 Product serial number	
Key	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion
Original Data	ProductId (REG_SZ) = 76487-018-3066024-22297
Output	76487-018-3066024-22297

6 Product Name	
Key	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion
Original Data	ProductName (REG_SZ) = Microsoft Windows XP CSDVersion (REG_SZ) = Service Pack 3
Output	Microsoft Windows XP Service Pack 3

7 Current Version	
Key	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion
Original Data	CurrentVersion (REG_SZ) = 5.1
Output	5.1

8 System Root	
Key	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion
Original Data	SystemRoot (REG_SZ) = C:\WINDOWS
Output	C:\WINDOWS

9 Installation Date	
Key	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion
Original Data	InstallDate (REG_DWORD) = 0x4b7ef3ec (1266611180)
Manipulation	Pass the unix timestamp through the date command
Output	Fri Feb 19 20:26:20 GMT 2010

10 Last logged Shutdown Time	
Key	HKLM\SYSTEM\CurrentControlSet\Control\Windows
Original Data	ShutdownTime (REG_BINARY) = 44 4d 91 16 25 f0 cb 01
Manipulation	Reverse endian 01cbf02516914d44 Convert to Unixtime /10000000 - 11644473600 Pass the unix timestamp through the date command
Output	Fri Feb 19 20:26:20 GMT 2010

11 Last user logged in	
Key	HKLM\SYSTEM\Microsoft\Windows NT\CurrentVersion\Winlogon
Original Data	DefaultUserName (REG_SZ) = student
Output	student

12 System Directory	
Key	HKLM\SYSTEM\CurrentControlSet\Control\Windows
Original Data	SystemDirectory (REG_EXPAND_SZ) = %SystemRoot%\system32
Output	%SystemRoot%\system32

13 Drive Letters	
Key	HKLM\SYSTEM\MountedDevices
Original Data (sampled)	\??\Volume{06169e44-1d4b-11df-9508-806d6172696f} (REG_BINARY) = 83 8b 83 8b 00 7e 00 00 00 00 00 00 \DosDevices\C: (REG_BINARY) = 83 8b 83 8b 00 7e 00 00 00 00 00 00 \??\Volume{a6f278c2-1d4d-11df-a398-806d6172696f} (REG_BINARY) = 5c 00 3f 00 3f 00 5c 00 46 00 44 00 43 00 23 00 47 00 45 00 4e 00 45 00 52 00 49 00 43 00 5f 00 46 00 4c 00 4f 00 50 00 50 00 59 00 5f 00 44 00 52 00 49 00 56 00 45 00 23 00 35 00 26 00 31 00 64 00 65 00 31 00 36 00 36 00 38 00 39 00 26 00 30 00 26 00 30 00 23 00 7b 00 35 00 33 00 66 00 35 00 36 00 33 00 30 00 64 00 2d 00 62 00 36 00 62 00 66 00 2d 00 31 00 31 00 64 00 30 00 2d 00 39 00 34 00 66 00 32 00 2d 00 30 00 30 00 61 00 30 00 63 00 39 00 31 00 65 00 66 00 62 00 38 00 62 00 7d 00 #{aa894f6d-5bfc-11e0-a3a5-00b0d0e9d26f} (REG_BINARY) = 54 72 75 65 43 72 79 70 74 56 6f 6c 75 6d 65 46
Manipulation	Search for any value containing DosDevices Report characters 12 and 13
Output	C: A: D: E: G:

14 Daylight savings Timezone	
Key	HKLM\SYSTEM\CurrentControlSet\Control\TimeZoneInformation
Original Data	DaylightName (REG_SZ) = Pacific Daylight Time DaylightBias (REG_DWORD) = 0xffffffc4 (4294967236)
Manipulation	Convert Bias => uppercase => binary => decimal n.b. Twos compliment is performed if binary number is 32 bits long i.e. leading bit is 1(negative) Make readable -60 => (Bias -60 Minutes)
Output	Pacific Daylight Time (Bias -60 Minutes)

15 Standard Timezone	
Key	HKLM\SYSTEM\CurrentControlSet\Control\TimeZoneInformation
Original Data	StandardName (REG_SZ) = Pacific Standard Time StandardBias (REG_DWORD) = 0x00000000 (0)
Manipulation	Convert Bias => uppercase => binary => decimal n.b. Twos compliment is performed if binary number is 32 bits long i.e. leading bit is 1(negative) Make readable 0 => (Bias +0 Minutes)
Output	Pacific Daylight Time (Bias +0 Minutes)

15a System Directory (Windows 7 only)	
Key	HKLM\SYSTEM\CurrentControlSet\Control\TimeZoneInformation
Original Data	TimeZoneKeyName (REG_SZ) = GMT Standard TimeTimeimeime^A^A<U+E074>©耀<U+E1D4>©拣瘕膷Ʊ��袍痍肝痍玑?겓琨 Ê^Aŋ<U+E1BC>©<U+E120>©胧痍겓琨 Ê^Aŋ<U+E1BC>©ŋ<U+E140>©‿玳겓琨 Ê^Aŋŋ<U+E16C>©蛯痍 Ê^Aŋŋ<U+ABCD>�<U+E1BC>©
Manipulation	Windows 7 stores .dll filenames for timezone names, this extra key is present, data is truncated before reporting
Output	GMT Standard Time

16 Timezone bias	
Key	HKLM\SYSTEM\CurrentControlSet\Control\TimeZoneInformation
Original Data	Bias (REG_DWORD) = 0x000001e0 (480)
Manipulation	Convert Bias => uppercase => binary => decimal n.b. Twos compliment is performed if binary number is 32 bits long i.e. leading bit is 1(negative) Make readable 480 => UTC +480 Minutes
Output	UTC +480 Minutes

17 Current time bias	
Key	HKLM\SYSTEM\CurrentControlSet\Control\TimeZoneInformation
Original Data	ActiveTimeBias (REG_DWORD) = 0x000001a4 (420)
Manipulation	Convert Bias => uppercase => binary => decimal n.b. Twos compliment is performed if binary number is 32 bits long i.e. leading bit is 1(negative) Make readable 420 => UTC +420 Minutes
Output	UTC +420 Minutes

18 Network time protocol is	
Key	HKLM\SYSTEM\CurrentControlSet\Services\W32Time\Parameters
Original Data	Type (REG_SZ) = NTP
Manipulation	If type is set to NTP the script reports synchronised otherwise it reports not synchronised
Output	synchronised

19 Timezone last updated	
Key	HKLM\SYSTEM\CurrentControlSet\Control\TimeZoneInformation
Original Data	$$$PROTO.HIV\ControlSet001\Control\TimeZoneInformation [2011-04-01T04:20:15Z]
Manipulation	Remove [] convert Z => (UTC) convert T => " "
Output	2011-04-01 04:20:15 (UTC)

20 Daylight saving start	
Key	HKLM\SYSTEM\CurrentControlSet\Control\TimeZoneInformation
Original Data	DaylightStart (REG_BINARY) = 00 00 03 00 02 00 02 00 00 00 00 00 00 00 00 00
Manipulation	For each 16 bit pair (e.g. 0a 00 => 000a => 000A => 10) Reverse endian convert hex => uppercase => decimal 1st - Unused 2nd - Month => translate month number to name 3rd - Week of the month 1st-4th or last 4th - Hour 5th - Minute 6th - Second 7th - Fraction of a second 8th - Day of week => translate number to day name
Output	Daylight saving starts on Sunday in the 2nd week of March at 02:00:00:00

21 Standard Time starts	
Key	HKLM\SYSTEM\CurrentControlSet\Control\TimeZoneInformation
Original Data	StandardStart (REG_BINARY) = 00 00 0b 00 01 00 02 00 00 00 00 00 00 00 00 00
Manipulation	As per “Daylight Saving start”
Output	Standard time starts on Sunday in the 1st week of November at 02:00:00:00

22 SYSTEM WIDE AUTORUNS	
Key	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
Original Data (sampled)	CMI-CreateHive{3D971F19-49AB-4000-8D39-A6D9C673D809}\Microsoft\Windows\CurrentVersion\Run [2010-08-16T17:20:56Z] ..\OptionalComponents QuickTime Task (REG_SZ) = "C:\Program Files\QuickTime\QTTask.exe" -atboottime iTunesHelper (REG_SZ) = "C:\Program Files\iTunes\iTunesHelper.exe" IgfxTray (REG_SZ) = C:\Windows\system32\igfxtray.exe HotKeysCmds (REG_SZ) = C:\Windows\system32\hkcmd.exe Persistence (REG_SZ) = C:\Windows\system32\igfxpers.exe
Manipulation	Scroll through the values and report the commands run
Output	"C:\Program Files\QuickTime\QTTask.exe" -atboottime "C:\Program Files\iTunes\iTunesHelper.exe" C:\Windows\system32\igfxtray.exe C:\Windows\system32\hkcmd.exe C:\Windows\system32\igfxpers.exe

23 INSTALLED APPLICATIONS (Uninstall)	
Key	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall
Original Data (sampled)	$$$PROTO.HIV\Microsoft\Windows\CurrentVersion\Uninstall [2011-04-01T04:43:02Z] ..\Adobe Flash Player ActiveX ..\Connection Manager ..\DirectAnimation ..\DirectDrawEx ..\DXM_Runtime ..\Fontcore $$$PROTO.HIV\Microsoft\Windows\CurrentVersion\Uninstall\Adobe Flash Player ActiveX [2011-04-01T04:39:27Z] DisplayName (REG_SZ) = Adobe Flash Player 10 ActiveX $$$PROTO.HIV\Microsoft\Windows\CurrentVersion\Uninstall\Connection Manager [2010-02-19T20:09:15Z] SystemComponent (REG_DWORD) = 0x00000001 (1) $$$PROTO.HIV\Microsoft\Windows\CurrentVersion\Uninstall\DirectAnimation [2010-02-19T20:17:49Z] (Default) (REG_SZ) = (no data)
Manipulation	Scroll through each subkey of Uninstall If the DisplayName value exists use that to describe the software otherwise use the key name To make timestamps readable remove [] convert Z => (UTC) convert T => “ “
Output	Adobe Flash Player 10 ActiveX --- 2011-04-01 04:39:27(UTC) Connection Manager --- 2010-02-19 20:09:15 (UTC) DirectAnimation --- 2010-02-19 20:17:49 (UTC) DirectDrawEx --- 2010-02-19 20:17:41 (UTC) DXM_Runtime --- 2010-02-19 20:20:35 (UTC)

Appendix D - Details of data manipulations performed by networkinfo.sh

Windows 7/Vista Sample networkinfo.sh output

```
1 eircom5376 0322
2    guid               : {106AEA67-3D31-464B-B6B4-2ABC4DF1DAD2}
3    Date created       : Saturday 14 August 2010 10:46:55:603
4    Last connected     : Wednesday 29 June 2011 18:23:53:865
5   Default Gateway MAC : 00 24 92 af c4 40 ( Motorola, Broadband Solutions Group)
6   DNS suffix          : <none>
7   Profile location    : Home

WaveLAN Network 2
   guid                : {5E2A1835-7549-4762-95A2-5A7E6B904FCA}
   Date created        : Thursday 30 September 2010 09:30:35:491
   Last connected      : Monday 11 October 2010 09:23:55:841
   Default Gateway MAC : 00 00 0c 07 ac 07 ( CISCO SYSTEMS, INC.)
   DNS suffix          : ucd.ie
   Profile location    :

8 Network instance guid {52EF53BF-2F0E-4B01-9511-F39EA0858A9B}

9   Hardware            : Intel(R) WiFi Link 5100 AGN
10  Domain name         : (no data)
11  Dhcp IP Address     : 192.168.1.26 (255.255.255.0)
12  Dhcp Server         : 192.168.1.254
13  Dhcp Enabled        : yes
14  Dhcp gateway MAC    : 00 24 92 af c4 40 ( Motorola, Broadband Solutions Group )
15  Lease period (secs): 0x00000e10 (3600)
16  Lease obtained      : Wed Jun 29 18:23:51 IST 2011
17  Lease Terminates    : Wed Jun 29 19:23:51 IST 2011
18  Static IP Address   : Not recorded
19  Network Connection : Wireless Network Connection
20  Media Subtype       : 0x00000002 (2)
21  Pnp Inst ID : PCI\VEN_8086&DEV_4232&SUBSYS_13218086&REV_00\4&1CFC9AC8&0&00E1

22 OUTGOING SHARES :

  name : print$
  path : C:\Windows\system32\spool\drivers

  name : Users
  path : C:\Users

  name : Computerforensics
  path : C:\Users\Jacky\Documents\Computerforensics
```

XP Sample networkinfo.sh output

```
Network instance guid {8191D8CC-2A02-4A0F-86FF-5C19D1021670}

  Hardware            : Intel(R) PRO/Wireless 3945ABG Network Connection
  Domain name         : (no data)
  Dhcp IP Address     : 192.168.1.164 (255.255.255.0)
  Dhcp Server         : 192.168.1.1
  Dhcp Enabled        : yes
  Dhcp gateway MAC    : Not recorded
  Lease period (secs): 0x0000a8c0 (43200)
  Lease obtained      : Tue Aug 2 13:26:09 IST 2011
  Lease Terminates    : Wed Aug 3 01:26:09 IST 2011
  Static IP Address   : Not recorded
  Network Connection : Wireless Network Connection
  Media Subtype      : 0x00000002 (2)
  Pnp Inst ID : PCI\VEN_8086&DEV_4222&SUBSYS_135C103C&REV_02\4&29E2C51B&0&00E1
  Wireless access points accessed by this NIC :


23   PavelHotspot
24     WAP MAC        - 7c 61 93 e9 05 fa (HTC Corporation)
25     Encryption     - WEP
26     Authentication - Open
27     Last access    - Tue Aug  2 13:26:09 IST 2011

     HTC Portable Hotspot
       WAP MAC        - 7c 61 93 e9 05 fa (HTC Corporation)
       Encryption     - WEP
       Authentication - Open
       Last access    - Tue Aug  2 13:12:33 IST 2011

     WaveLAN Network
       WAP MAC        - 00 22 0d e1 dd e1 (Cisco Systems)
       Encryption     - Disabled
       Authentication - Open
       Last access    - Tue Aug  2 07:59:57 IST 2011
```

1 network name	
Key	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion \NetworkList\Profiles (scroll through the list)
Original Data	..\{106AEA67-3D31-464B-B6B4-2ABC4DF1DAD2} [2011-06-29T17:23:53Z] ProfileName (REG_SZ) = eircom5376 0322
Output	eircom5376 0322

2 guid	
Key	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion \NetworkList\Profiles (scroll through the list)

Original Data	..\{106AEA67-3D31-464B-B6B4-2ABC4DF1DAD2}
	..\{5E2A1835-7549-4762-95A2-5A7E6B904FCA}
	..\{7A4CA7EB-AA56-4BB3-9B45-6D10B72A8A7E}
	..\{A2F9FED5-FAD8-40F1-88CA-DC20E066B561}

Output	{106AEA67-3D31-464B-B6B4-2ABC4DF1DAD2}

3 Date created	
Key	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion \NetworkList\Profiles (scroll through the list)
Original Data	..\{106AEA67-3D31-464B-B6B4-2ABC4DF1DAD2} [2011-06-29T17:23:53Z] ProfileName (REG_SZ) = eircom5376 0322 Description (REG_SZ) = eircom5376 0322 Managed (REG_DWORD) = 0x00000000 (0) Category (REG_DWORD) = 0x00000001 (1) DateCreated (REG_BINARY) = da 07 08 00 06 00 0e 00 0a 00 2e 00 37 00 5b 02
Manipulation	Reverse endian => decimal => translate if useful da 07 => 07da => 2010 08 00 => 0008 => 8 => August 06 00 => 0006 => 6 => Saturday 0e 00 => 000e => 14 0a 00 => 000a => 10 2e 00 => 002e => 46 37 00 => 0037 => 55 5b 02 => 025b => 603
Output	Saturday 14 August 2010 10:46:55:603

4 Last connected	
Key	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion \NetworkList\Profiles (scroll through the list)
Original Data	..\{106AEA67-3D31-464B-B6B4-2ABC4DF1DAD2} [2011-06-29T17:23:53Z] ProfileName (REG_SZ) = eircom5376 0322 Description (REG_SZ) = eircom5376 0322 Managed (REG_DWORD) = 0x00000000 (0) Category (REG_DWORD) = 0x00000001 (1) DateCreated (REG_BINARY) = da 07 08 00 06 00 0e 00 0a 00 2e 00 37 00 5b 02 NameType (REG_DWORD) = 0x00000047 (71) DateLastConnected (REG_BINARY) = db 07 06 00 03 00 1d 00 12 00 17 00 35 00 61 03
Manipulation	Reverse endian => decimal => translate if useful db 07 => 07db => 2011 06 00 => 0006 => 8 => June 03 00 => 0003 => 3 => Wednesday 1d 00 => 001d => 29 12 00 => 0012 => 18 17 00 => 0017 => 23 35 00 => 0035 => 53 61 03 => 0361 => 865
Output	Wednesday 29 June 2011 18:23:53:865

5 Default Gateway MAC	
Key	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Signatures\Unmanaged
Original Data	CMI-CreateHive{199DAFC2-6F16-4946-BF90-5A3FC3A60902}\Microsoft\Windows NT\CurrentVersion\NetworkList\Signatures\Unmanaged\010103000F0000F0080000000F0000F07CB5E1B29932DD4A16BC48E3874CCC1EDC8C75AF028547E086ADAF76700AA598 [2010-08-14T09:46:55Z] ProfileGuid (REG_SZ) = {106AEA67-3D31-464B-B6B4-2ABC4DF1DAD2} Description (REG_SZ) = eircom5376 0322 Source (REG_DWORD) = 0x00000008 (8) DnsSuffix (REG_SZ) = <none> FirstNetwork (REG_SZ) = eircom5376 0322 DefaultGatewayMac (REG_BINARY) = 00 24 92 af c4 40
Manipulation	Scroll through subkeys and look for matching guids Report MAC Compare first 6 characters of MAC address to ieee database to get WAP manufacturer and report
Output	00 24 92 af c4 40 (Motorola, Broadband Solutions Group)

6 Profile location (Windows 7 only)	
Key	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Homegroup\NetworkLocations (scroll through values)
Original Data (sampled)	..\Home ..\Work \Home [2010-08-14T09:47:15Z] {106AEA67-3D31-464B-B6B4-2ABC4DF1DAD2} (REG_SZ) = eircom5376 0322
Manipulation	Check inside each location Report if specified guid value is present
Output	Home

7 DNS suffix	
Key	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Signatures\Unmanaged
Original Data	CMI-CreateHive{199DAFC2-6F16-4946-BF90-5A3FC3A60902}\Microsoft\Windows NT\CurrentVersion\NetworkList\Signatures\Unmanaged\010103000F0000F0080000000F0000F020EBA907701B5DC91B7FD5517EA7C48D5E0063A5D221EA19F9A76C9F6EA4DF6B [2010-08-16T08:08:41Z] ProfileGuid (REG_SZ) = {A2F9FED5-FAD8-40F1-88CA-DC20E066B561} Description (REG_SZ) = WaveLAN Network Source (REG_DWORD) = 0x00000008 (8) DnsSuffix (REG_SZ) = ucd.ie FirstNetwork (REG_SZ) = WaveLAN Network DefaultGatewayMac (REG_BINARY) = 00 00 0c 07 ac 09
Manipulation	Scroll through subkeys and look for matching guids And report DNS suffix
Output	ucd.ie

8 Network instance guid	
Key	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters (scroll through instances)
Original Data	CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB51D4737F5}\ControlSet001\services\Tcpip\Parameters\Interfaces\{52EF53BF-2F0E-4B01-9511-F39EA0858A9B}
Output	{52EF53BF-2F0E-4B01-9511-F39EA0858A9B}

9 Hardware	
Key	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkCards
Original Data	CMI-CreateHive{199DAFC2-6F16-4946-BF90-5A3FC3A60902}\Microsoft\Windows NT\CurrentVersion\NetworkCards [2009-12-16T18:31:39Z] ..\10 ..\8 CMI-CreateHive{199DAFC2-6F16-4946-BF90-5A3FC3A60902}\Microsoft\Windows NT\CurrentVersion\NetworkCards\10 [2009-12-16T18:57:49Z] ServiceName (REG_SZ) = {651D7F7A-39B3-415C-A4B8-D510F3A3D693} Description (REG_SZ) = Realtek PCIe GBE Family Controller CMI-CreateHive{199DAFC2-6F16-4946-BF90-5A3FC3A60902}\Microsoft\Windows NT\CurrentVersion\NetworkCards\8 [2009-12-16T18:31:17Z] ServiceName (REG_SZ) = {52EF53BF-2F0E-4B01-9511-F39EA0858A9B} Description (REG_SZ) = Intel(R) WiFi Link 5100 AGN
Manipulation	Match servicename with network instance guid and report description
Output	Intel(R) WiFi Link 5100 AGN

10 Domain name	
Key	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters (scroll through instances)
Original Data	CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB51D4737F5}\ControlSet001\services\Tcpip\Parameters\Interfaces\{52EF53BF-2F0E-4B01-9511-F39EA0858A9B} [2011-06-29T17:27:22Z] UseZeroBroadcast (REG_DWORD) = 0x00000000 (0) EnableDeadGWDetect (REG_DWORD) = 0x00000001 (1) EnableDHCP (REG_DWORD) = 0x00000001 (1) NameServer (REG_SZ) = (no data) Domain (REG_SZ) = (no data)
Output	(no data)

11 Dhcp IP Address	
Key	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters (scroll through instances)
Original Data	DhcpIPAddress (REG_SZ) = 192.168.1.26 DhcpSubnetMask (REG_SZ) = 255.255.255.0
Output	192.168.1.26 (255.255.255.0)

12 Dhcp Server	
Key	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters (scroll through instances)
Original Data	DhcpServer (REG_SZ) = 192.168.1.254
Output	192.168.1.254

13 Dhcp Enabled	
Key	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters (scroll through instances)
Original Data	EnableDHCP (REG_DWORD) = 0x00000001 (1)
Manipulation	1 => yes All other values => no
Output	yes

14 Dhcp gateway MAC	
Key	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters (scroll through instances)
Original Data	DhcpGatewayHardware (REG_BINARY) = c0 a8 01 fe 06 00 00 00 00 24 92 af c4 40
Manipulation	Compare first 6 characters of MAC address to ieee database to get WAP manufacturer and report
Output	00 24 92 af c4 40 (Motorola, Broadband Solutions Group)

15 Lease period (secs)	
Key	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters (scroll through instances)
Original Data	Lease (REG_DWORD) = 0x00000e10 (3600)
Output	0x00000e10 (3600)

16 Lease obtained	
Files	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters (scroll through instances)
Original Data	LeaseObtainedTime (REG_DWORD) = 0x4e0b5fa7 (1309368231)
Manipulation	Unixtime => date command
Output	Wed Jun 29 18:23:51 IST 2011

17 Lease Terminates	
Key	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters (scroll through instances)
Original Data	LeaseTerminatesTime (REG_DWORD) = 0x4e0b6db7 (1309371831)
Manipulation	Unixtime => date command
Output	Wed Jun 29 19:23:51 IST 2011

18 Static IP Address	
Key	HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Adapters (scroll through instances)
Original Data	IPAddress (REG_SZ) = 0.0.0.0
Manipulation	Only exists if static IP addressing used If the value doesn’t exist reports “Not recorded”
Output	Not recorded

19 Network Connection	
Key	HKLM\SYSTEM\CurrentControlSet\Control\Network\
Original Data	CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB51D4737F5}\ControlSet001\Control\Network\{4D36E972-E325-11CE-BFC1-08002BE10318}\{52EF53BF-2F0E-4B01-9511-F39EA0858A9B}\Connection [2009-12-16T18:34:49Z] DefaultNameResourceId (REG_DWORD) = 0x0000070e (1806) DefaultNameIndex (REG_DWORD) = 0x00000000 (0) Name (REG_SZ) = Wireless Network Connection PnpInstanceID (REG_SZ) = PCI\VEN_8086&DEV_4232&SUBSYS_13218086&REV_00\4&1CFC9AC8&0&00E1 MediaSubType (REG_DWORD) = 0x00000002 (2)
Output	Wireless Network Connection

20 Media Subtype	
Key	HKLM\SYSTEM\CurrentControlSet\Control\Network\
Original Data	CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB51D4737F5}\ControlSet001\Control\Network\{4D36E972-E325-11CE-BFC1-08002BE10318}\{52EF53BF-2F0E-4B01-9511-F39EA0858A9B}\Connection [2009-12-16T18:34:49Z] DefaultNameResourceId (REG_DWORD) = 0x0000070e (1806) DefaultNameIndex (REG_DWORD) = 0x00000000 (0) Name (REG_SZ) = Wireless Network Connection PnpInstanceID (REG_SZ) = PCI\VEN_8086&DEV_4232&SUBSYS_13218086&REV_00\4&1CFC9AC8&0&00E1 MediaSubType (REG_DWORD) = 0x00000002 (2)
Output	0x00000002 (2)

21 Pnp Inst ID	
Key	HKLM\SYSTEM\CurrentControlSet\Control\Network\
Original Data	CMI-CreateHive{2A7FB991-7BBE-4F9D-B91E-7CB51D4737F5}\ControlSet001\Control\Network\{4D36E972-E325-11CE-BFC1-08002BE10318}\{52EF53BF-2F0E-4B01-9511-F39EA0858A9B}\Connection [2009-12-16T18:34:49Z] DefaultNameResourceId (REG_DWORD) = 0x0000070e (1806) DefaultNameIndex (REG_DWORD) = 0x00000000 (0) Name (REG_SZ) = Wireless Network Connection PnpInstanceID (REG_SZ) = PCI\VEN_8086&DEV_4232&SUBSYS_13218086&REV_00\4&1CFC9AC8&0&00E1 MediaSubType (REG_DWORD) = 0x00000002 (2)
Output	PCI\VEN_8086&DEV_4232&SUBSYS_13218086&REV_00\4&1CFC9AC8&0&00E1

22 OUTGOING SHARES	
Key	HKLM\SYSTEM\CurrentControlSet\Services\LanmanServer\Shares
Original Data (sampled)	print$ (REG_MULTI_SZ) = [0] CSCFlags=768 [1] MaxUses=4294967295 [2] Path=C:\Windows\system32\spool\drivers [3] Permissions=0 [4] Remark=Printer Drivers [5] ShareName=print$ [6] Type=0
Output	name : print$ path : C:\Windows\system32\spool\drivers

23 Wireless Access Points	
Key	HKLM\SOFTWARE\Microsoft\WZCSVC\Parameters\Interfaces\ (scroll through guids)
Original Data (sampled)	$$$PROTO.HIV\Microsoft\WZCSVC\Parameters\Interfaces [2011-08-02T12:33:17Z] ..\{8191D8CC-2A02-4A0F-86FF-5C19D1021670} Static#0000 (REG_BINARY) = c8 02 00 00 01 00 00 00 7c 61 93 e9 05 fa 00 00 0c 00 00 00 50 61 76 65 6c 48 6f 74 73 70 6f 74 00 00
Manipulation	0c => 12 = length of name 50 61 76 65 6c 48 6f 74 73 70 6f 74 => PavelHotspot
Output	PavelHotspot

24 WAP MAC	
Key	HKLM\SOFTWARE\Microsoft\WZCSVC\Parameters\Interfaces\ (scroll through guids)
Original Data (sampled)	$$$PROTO.HIV\Microsoft\WZCSVC\Parameters\Interfaces [2011-08-02T12:33:17Z] ..\{8191D8CC-2A02-4A0F-86FF-5C19D1021670} Static#0000 (REG_BINARY) = c8 02 00 00 01 00 00 00 7c 61 93 e9 05 fa 00 ....
Manipulation	Compare first 6 characters of MAC address to ieee database to get WAP manufacturer and report
Output	7c 61 93 e9 05 fa (HTC Corporation)

25 Encryption	
Key	HKLM\SOFTWARE\Microsoft\WZCSVC\Parameters\Interfaces\ (scroll through guids)
Original Data (sampled)	$$$PROTO.HIV\Microsoft\WZCSVC\Parameters\Interfaces [2011-08-02T12:33:17Z] ..\{8191D8CC-2A02-4A0F-86FF-5C19D1021670} Static#0000 (REG_BINARY) = c8 02 00 00 01 00 00 00 7c 61 93 e9 05 fa 00 00 0c 00 00 00 50 61 76 65 6c 48 6f 74 73 70 6f 74 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Manipulation	00 => WEP 01 => Disabled 04 => TKIP 06 => AES
Output	WEP

26 Authentication	
Key	HKLM\SOFTWARE\Microsoft\WZCSVC\Parameters\Interfaces\ (scroll through guids)
Original Data (sampled)	$$$PROTO.HIV\Microsoft\WZCSVC\Parameters\Interfaces [2011-08-02T12:33:17Z] ..\{8191D8CC-2A02-4A0F-86FF-5C19D1021670} Static#0000 (REG_BINARY) = c8 02 00 00 01 00 00 00 7c 61 93 e9 05 fa 00 00 0c 00 00 00 50 61 76 65 6c 48 6f 74 73 70 6f 74 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 e2 ff ff ff 03 00 00 00 20 00 00 00 64 00 00 00 00 00 00 00 e0 cd 24 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 82 84 8b 96 24 30 48 6c 00 00 00 00 0d 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Manipulation	00 => Open 01 => Shared 04 => WPA 06 => WPA-PSK
Output	Open

27 Last access	
Key	HKLM\SOFTWARE\Microsoft\WZCSVC\Parameters\Interfaces\ (scroll through guids)
Original Data (sampled)	$$$PROTO.HIV\Microsoft\WZCSVC\Parameters\Interfaces [2011-08-02T12:33:17Z] ..\{8191D8CC-2A02-4A0F-86FF-5C19D1021670} Static#0000 (REG_BINARY) = c8 02 00 00 01 00 00 00 7c 61 93 e9 05 fa 00 00 0c 00 00 00 50 61 76 65 6c 48 6f 74 73 70 6f 74 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 e2 ff ff ff 03 00 00 00 20 00 00 00 64 00 00 00 00 00 00 00 e0 cd 24 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 82 84 8b 96 24 30 48 6c 00 00 00 00 0d 00 00 .... 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 6c 27 ba 5b 0f 51 cc 01 01
Manipulation	Last access time does not always exist Reverse endian 01cc510f5bba276c Convert to Unixtime /10000000 - 11644473600 Pass the unix timestamp through the date command
Output	Tue Aug 2 13:26:09 IST 2011

Appendix E – Output from networkinfo.sh and regripper

```
ACTIVE NETWORK INSTANCES RECORDED ON THIS SYSTEM

Network instance guid {36F1A0E2-A988-40EF-A18C-D76322FFB663}

  Hardware              : Broadcom NetLink Gigabit Ethernet
  Domain name            : (no data)
  Dhcp IP Address       : 137.43.155.85 (255.255.254.0)
  Dhcp Server           : 137.43.116.17
  Dhcp Enabled          : yes
  Dhcp gateway MAC      : Not recorded
  Lease period (secs   ): 0x0000a8c0 (43200)
  Lease obtained       : Tue Aug 2 07:59:44 IST 2011
  Lease Terminates      : Tue Aug 2 19:59:44 IST 2011
  Static IP Address     : Not recorded
  Network Connection   : Local Area Connection
  Media Subtype        : 0x00000001 (1)
  Pnp Inst ID : PCI\VEN_14E4&DEV_1693&SUBSYS_30C0103C&REV_02\4&227633DA&0&00E2

Network instance guid {8191D8CC-2A02-4A0F-86FF-5C19D1021670}

  Hardware             : Intel(R) PRO/Wireless 3945ABG Network Connection
  Domain name          : (no data)
  Dhcp IP Address      : 192.168.1.164 (255.255.255.0)
  Dhcp Server          : 192.168.1.1
  Dhcp Enabled         : yes
  Dhcp gateway MAC     : Not recorded
  Lease period (secs)  : 0x0000a8c0 (43200)
  Lease obtained       : Tue Aug 2 13:26:09 IST 2011
  Lease Terminates     : Wed Aug 3 01:26:09 IST 2011
  Static IP Address    : Not recorded
  Network Connection   : Wireless Network Connection
  Media Subtype        : 0x00000002 (2)
  Pnp Inst ID : PCI\VEN_8086&DEV_4222&SUBSYS_135C103C&REV_02\4&29E2C51B&0&00E1
  Wireless access points accessed by this NIC :

      PavelHotspot
        WAP MAC            - 7c 61 93 e9 05 fa (HTC Corporation)
        Encryption         - WEP
        Authentication     - Open
        Last access        - Tue Aug  2 13:26:09 IST 2011

      HTC Portable Hotspot
        WAP MAC            - 7c 61 93 e9 05 fa (HTC Corporation)
        Encryption         - WEP
```

```
    Authentication       - Open
    Last access          - Tue Aug  2 13:12:33 IST 2011

   WaveLAN Network
    WAP MAC              - 00 22 0d e1 dd e1 (Cisco Systems)
    Encryption           - Disabled
    Authentication       - Open
    Last access          - Tue Aug  2 07:59:57 IST 2011
```

Fig 49 Windows XP networkinfo output

```
NETWORK PROFILES RECORDED ON THIS SYSTEM

Network 3
  guid                   : {02C1B784-578D-45AB-870E-5EEC7FD803A9}
  Date created           : Friday 12 February 2010 20:15:40:912
  Last connected         : Monday 15 February 2010 08:58:25:240
  Default Gateway MAC : 00 50 e8 00 6e 0a ( NOMADIX INC.)
  DNS suffix             : nomadix.com

Radisson
  guid                   : {03383345-A507-4C0D-9B78-CF0F184EE23F}
  Date created           : Friday 25 September 2009 10:02:25:89
  Last connected         : Friday 25 September 2009 10:03:22:195
  Default Gateway MAC : 00 00 c5 e9 20 98 ( FARALLON COMPUTING/NETOPIA)
  DNS suffix             : <none>

Wayport_Access 3
  guid                   : {65D9480C-D8FB-495F-95E3-7149A800E281}
  Date created           : Wednesday 5 August 2009 22:58:49:63
  Last connected         : Friday 7 August 2009 21:15:54:18
  Default Gateway MAC : 00 90 fb 11 cf 96 ( PORTWELL, INC.)
  DNS suffix             : ncv.lax.wayport.net

avoca
  guid                   : {6EE1C79E-6128-492A-9068-D41FFAD021A7}
  Date created           : Saturday 30 August 2008 16:24:19:411
  Last connected         : Sunday 25 October 2009 17:18:30:971
  Default Gateway MAC : 00 00 c5 e9 20 98 ( FARALLON COMPUTING/NETOPIA)
  DNS suffix             : <none>

gods
  guid                   : {9CFB0795-3F47-4AE8-B69D-9B7F0F6A5680}
  Date created           : Wednesday 7 July 2010 13:45:21:97
  Last connected         : Friday 9 July 2010 02:59:56:55
  Default Gateway MAC : 00 0c e5 73 67 55 ( Motorola Mobility, Inc.)
```

```
   DNS suffix                 : hsd1.ma.comcast.net.

eircom5376 0322
   guid                       : {B036A9A5-1E5D-467F-A387-70A0FEF6CCCC}
   Date created               : Friday 30 October 2009 15:26:51:255
   Last connected             : Thursday 19 January 2012 10:54:29:265
   Default Gateway MAC : 00 24 92 af c4 40 ( Motorola, Broadband Solutions Group)
   DNS suffix                 : <none>

Network 2
   guid                       : {B67A024A-FF31-43E7-A645-1A485DF18809}
   Date created               : Saturday 8 August 2009 21:49:36:462
   Last connected             : Sunday 9 August 2009 21:53:31:219
   Default Gateway MAC : 00 90 fb 21 01 ec ( PORTWELL, INC.)
   DNS suffix                 : grand.las.wayport.net

Network
   guid                       : {B99AB9C0-3BA7-49D9-96C8-F849C3D8EAA3}
   Date created               : Monday 15 September 2008 18:49:03:767
   Last connected             : Monday 31 August 2009 12:36:11:657
   Default Gateway MAC : 00 90 4b 34 c7 5c ( GemTek Technology Co., Ltd.)
   DNS suffix                 : <none>

Wayport_Access
   guid                       : {C3ED7326-0B86-4A33-99C0-EA92B360F021}
   Date created               : Sunday 8 March 2009 10:43:47:44
   Last connected             : Wednesday 18 March 2009 14:04:42:538
   Default Gateway MAC : 00 08 54 24 bf 7e ( Netronix, Inc.)
   DNS suffix                 : grav.orl.wayport.net

Wayport_Access 2
   guid                       : {C5736C6F-D5E5-4DB1-ACD6-3D22A5299F75}
   Date created               : Saturday 25 July 2009 22:29:07:738
   Last connected              : Monday 27 July 2009 07:54:50:482
   Default Gateway MAC : 00 90 fb 0d 19 dc ( PORTWELL, INC.)
   DNS suffix                 : tim.sjc.wayport.net

hhonors
   guid                       : {EB855896-5876-4D6B-A33C-E9D2D25C9BE1}
   Date created               : Tuesday 21 July 2009 02:23:07:77
   Last connected             : Friday 24 July 2009 08:11:02:957
   Default Gateway MAC : 00 50 e8 01 a1 f9 ( NOMADIX INC.)
   DNS suffix                 : Hilton.com
```

Fig 50 Windows Vista networkinfo.sh WAP output

```
>>>>perl rip.pl -r ../hives/vistal/software -p vista_wireless

Microsoft\Windows NT\CurrentVersion\NetworkList\Profiles

LastWrite = Fri Sep 25 09:03:48 2009 Z
  Radisson [Radisson]
LastWrite = Sat Aug  8 04:15:54 2009 Z
  Wayport_Access  3 [Wayport_Access]
LastWrite = Sun Oct 25 17:18:30 2009 Z
  avoca [avoca]
LastWrite = Fri Jul  9 01:59:56 2010 Z
  gods [gods]
LastWrite = Thu Jan 19 10:54:29 2012 Z
  eircom5376 0322 [eircom5376 0322]
LastWrite = Wed Mar 18 14:04:42 2009 Z
  Wayport_Access [Wayport_Access]
LastWrite = Mon Jul 27 14:54:50 2009 Z
  Wayport_Access  2 [Wayport_Access]
LastWrite = Fri Jul 24 15:11:02 2009 Z
  hhonors [hhonors]
```

Fig 51 Regripper Windows Vista WAP report

```
>>>>perl rip.pl -r ../hives/xpknap2a/software -p networkcards

NetworkCards
Microsoft\Windows NT\CurrentVersion\NetworkCards

{36F1A0E2-A988-40EF-A18C-D76322FFB663} Broadcom NetLink Gigabit Ethernet [Thu Jul 28
21:25:39 2011]
{EE213748-59FA-4ED6-B31B-2153AD5E5783} 1394 Net Adapter [Wed Jul 20 12:20:32 2011]
{8191D8CC-2A02-4A0F-86FF-5C19D1021670} Intel(R) PRO/Wireless 3945ABG Network Connection
[Thu Jul 28 21:30:05 2011]

>>>>perl rip.pl -r ../hives/xpknap2a/system -p network



Network key
ControlSet001\Control\Network\{4D36E972-E325-11CE-BFC1-08002BE10318}
Interface {36F1A0E2-A988-40EF-A18C-D76322FFB663}
LastWrite time Thu Jul 28 21:25:41 2011 (UTC)
   Name            = Local Area Connection
   PnpInstanceID    =
PCI\VEN_14E4&DEV_1693&SUBSYS_30C0103C&REV_02\4&227633DA&0&00E2
   MediaSubType     = 1
```

```
Interface {8191D8CC-2A02-4A0F-86FF-5C19D1021670}
LastWrite time Thu Jul 28 21:30:05 2011 (UTC)
   Name           = Wireless Network Connection
   PnpInstanceID    =
PCI\VEN_8086&DEV_4222&SUBSYS_135C103C&REV_02\4&29E2C51B&0&00E1
   MediaSubType     = 2

Interface {EE213748-59FA-4ED6-B31B-2153AD5E5783}
LastWrite time Wed Jul 20 19:44:06 2011 (UTC)
   Name           = 1394 Connection
   PnpInstanceID    = V1394\NIC1394\29310B1023F99
   MediaSubType     = 5

>>>>perl rip.pl -r ../hives/xpknap2a/software -p networkuid



Microsoft\Windows NT\CurrentVersion\Network
LastWrite time = Wed Jul 20 19:46:58 2011

UID value not found.

>>>>perl rip.pl -r ../hives/xpknap2a/system -p nic

Adapter: {36F1A0E2-A988-40EF-A18C-D76322FFB663}
LastWrite Time: Thu Jul 28 21:25:38 2011 Z
  EnableDHCP         1
  IPAddress          0.0.0.0
  SubnetMask         0.0.0.0
  DefaultGateway
  DhcpIPAddress      137.43.155.85
  DhcpSubnetMask     255.255.254.0
  DhcpServer         137.43.116.17
  Lease              43200
  LeaseObtainedTime    Tue Aug  2 06:59:44 2011 Z
  T1                 Tue Aug  2 12:59:44 2011 Z
  T2                 Tue Aug  2 17:29:44 2011 Z
  LeaseTerminatesTime  Tue Aug  2 18:59:44 2011 Z

Adapter: {8191D8CC-2A02-4A0F-86FF-5C19D1021670}
LastWrite Time: Thu Jul 28 21:30:05 2011 Z
  EnableDHCP         1
  IPAddress          0.0.0.0
  SubnetMask         0.0.0.0
  DefaultGateway
  DhcpIPAddress      192.168.1.164
  DhcpSubnetMask      255.255.255.0
  DhcpServer         192.168.1.1
  Lease              43200
  LeaseObtainedTime    Tue Aug  2 12:26:09 2011 Z
  T1                 Tue Aug  2 18:26:09 2011 Z
  T2                 Tue Aug  2 22:56:09 2011 Z
  LeaseTerminatesTime  Wed Aug  3 00:26:09 2011 Z
  DhcpDefaultGateway   192.168.1.1
  DhcpSubnetMaskOpt    255.255.255.0

Adapter: {EE213748-59FA-4ED6-B31B-2153AD5E5783}
LastWrite Time: Wed Jul 20 19:44:02 2011 Z
  EnableDHCP         1
  IPAddress          0.0.0.0
  SubnetMask         0.0.0.0
  DefaultGateway

>>>>perl rip.pl -r ../hives/xpknap2a/system -p nic2

Adapter: {2D4F1EEC-4A21-4289-BA3B-933F2DBA7DF6}
LastWrite Time: Wed Jul 20 19:44:05 2011 Z
  UseZeroBroadcast           0
  EnableDHCP                 0
  IPAddress                0.0.0.0
  SubnetMask                 0.0.0.0
  DefaultGateway
  EnableDeadGWDetect          1
  DontAddDefaultGateway       0

Adapter: {36F1A0E2-A988-40EF-A18C-D76322FFB663}
LastWrite Time: Tue Aug  2 07:03:31 2011 Z
  UseZeroBroadcast           0
  EnableDeadGWDetect          1
  EnableDHCP                 1
  IPAddress                0.0.0.0
  SubnetMask                 0.0.0.0
  DefaultGateway
  DefaultGatewayMetric
  NameServer
  Domain
  RegistrationEnabled         1
  RegisterAdapterName         0
  TCPAllowedPorts            0
  UDPAllowedPorts            0
  RawIPAllowedProtocols       0
  NTEContextList            0x00000002
  DhcpClassIdBin
  DhcpServer               137.43.116.17
  Lease                    43200
  LeaseObtainedTime          Tue Aug  2 06:59:44 2011 Z
  T1                       Tue Aug  2 12:59:44 2011 Z
  T2                       Tue Aug  2 17:29:44 2011 Z
  LeaseTerminatesTime        Tue Aug  2 18:59:44 2011 Z
  IPAutoconfigurationAddress   0.0.0.0
  IPAutoconfigurationMask     255.255.0.0
  IPAutoconfigurationSeed     0
  AddressType              0
  IsServerNapAware          0
  DhcpIPAddress             137.43.155.85
  DhcpSubnetMask            255.255.254.0

Adapter: {5C1789BD-3DC8-44DF-B92B-B9238C458616}
LastWrite Time: Wed Jul 20 19:44:05 2011 Z
  UseZeroBroadcast          0
  EnableDHCP               0
  IPAddress               0.0.0.0
  SubnetMask               0.0.0.0
  DefaultGateway
  EnableDeadGWDetect         1
  DontAddDefaultGateway       0

Adapter: {8191D8CC-2A02-4A0F-86FF-5C19D1021670}
LastWrite Time: Tue Aug  2 12:33:17 2011 Z
  UseZeroBroadcast          0
  EnableDeadGWDetect         1
  EnableDHCP               1
  IPAddress               0.0.0.0
  SubnetMask               0.0.0.0
  DefaultGateway
  DefaultGatewayMetric
  NameServer
  Domain
  RegistrationEnabled        1
  RegisterAdapterName        0
  TCPAllowedPorts           0
  UDPAllowedPorts           0
  RawIPAllowedProtocols       0
  NTEContextList            0x00000002
  DhcpClassIdBin
  DhcpServer               192.168.1.1
  Lease                    43200
  LeaseObtainedTime          Tue Aug  2 12:26:09 2011 Z
  T1                       Tue Aug  2 18:26:09 2011 Z
  T2                       Tue Aug  2 22:56:09 2011 Z
  LeaseTerminatesTime        Wed Aug  3 00:26:09 2011 Z
  IPAutoconfigurationAddress   0.0.0.0
  IPAutoconfigurationMask     255.255.0.0
  IPAutoconfigurationSeed     2734101704
  AddressType              0
  IsServerNapAware          0
  DhcpIPAddress            192.168.1.164
  DhcpSubnetMask            255.255.255.0
  DhcpNameServer            192.168.1.1
  DhcpDefaultGateway         192.168.1.1
  DhcpSubnetMaskOpt          255.255.255.0

Adapter: {EE213748-59FA-4ED6-B31B-2153AD5E5783}
LastWrite Time: Wed Jul 20 19:44:02 2011 Z
  UseZeroBroadcast          0
  EnableDHCP               1
  IPAddress               0.0.0.0
  SubnetMask               0.0.0.0
  DefaultGateway
  DefaultGatewayMetric
  NameServer
  Domain
  RegistrationEnabled        1
  RegisterAdapterName        0
  TCPAllowedPorts           0
  UDPAllowedPorts           0
  RawIPAllowedProtocols       0


>>>>perl rip.pl -r ../hives/xpknap2a/system -p nic_mst2

Network key
ControlSet001\Control\Network\{4D36E972-E325-11CE-BFC1-08002BE10318}

ControlSet001\Services\Tcpip\Parameters\Interfaces
LastWrite time Thu Jul 28 21:30:05 2011 (UTC)

Interface {36F1A0E2-A988-40EF-A18C-D76322FFB663}
Name: Local Area Connection
Control\Network key LastWrite time Thu Jul 28 21:25:41 2011 (UTC)
Services\Tcpip key LastWrite time Tue Aug  2 07:03:31 2011 (UTC)
```

```
   DhcpDomain     =
   DhcpIPAddress  = 137.43.155.85
   DhcpSubnetMask = 255.255.254.0
   DhcpNameServer =
   DhcpServer     = 137.43.116.17

Interface {EE213748-59FA-4ED6-B31B-2153AD5E5783}
Name: 1394 Connection
Control\Network key LastWrite time Wed Jul 20 19:44:06 2011 (UTC)
Services\Tcpip key LastWrite time Wed Jul 20 19:44:02 2011 (UTC)
   DhcpDomain     =
   DhcpIPAddress  =
   DhcpSubnetMask =
   DhcpNameServer =
   DhcpServer     =

Interface {8191D8CC-2A02-4A0F-86FF-5C19D1021670}
Name: Wireless Network Connection
Control\Network key LastWrite time Thu Jul 28 21:30:05 2011 (UTC)
Services\Tcpip key LastWrite time Tue Aug  2 12:33:17 2011 (UTC)
   DhcpDomain     =
   DhcpIPAddress  = 192.168.1.164
   DhcpSubnetMask = 255.255.255.0
   DhcpNameServer = 192.168.1.1
   DhcpServer     = 192.168.1.1

>>>>perl rip.pl -r ../hives/xpknap2a/software -p ssid
SSID
Microsoft\WZCSVC\Parameters\Interfaces

NIC: Intel(R) PRO/Wireless 3945ABG Network Connection
Key LastWrite: Tue Aug  2 12:33:17 2011 UTC

Tue Aug  2 12:26:09 2011 MAC: 7C-61-93-E9-05-FA  PavelHotspot
Tue Aug  2 12:12:33 2011 MAC: 7C-61-93-E9-05-FA  HTC Portable Hotspot
Tue Aug  2 06:59:57 2011 MAC: 00-22-0D-E1-DD-E1  WaveLAN Network

Microsoft\EAPOL\Parameters\Interfaces

NIC: Intel(R) PRO/Wireless 3945ABG Network Connection
LastWrite time: Tue Aug  2 07:15:40 2011 UTC
1    WaveLAN Network
2    HTC Portable Hotspot
3    PavelHotspot
```

Fig 52 Regripper XP output from various network plugins