

**WINDOWS REGISTRY: A GENERALIZED
METHODOLOGY BASED ON CROSS-VALIDATED
FORENSIC ANALYSIS**



Author

Amir Amin

273794-MS(IS)-11-2018F

Supervisor

Dr. Omar Arif

A thesis submitted in partial fulfilment of the requirements for the degree of
Masters in Information Security (MS IS)

Department of Computing (DoC)
School of Electrical Engineering and Computer Science (SEECS)
National University of Science and Technology (NUST)
Islamabad, Pakistan
(January 2020)

CERTIFICATE OF ORIGINALITY

I hereby declare that this submission titled “Windows Registry: A Generalized Methodology based on Cross-Validated Forensic Analysis” is my own work and to the best of my knowledge. It contains material not published previously or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgment, is made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project’s design and conception or in style, presentation and linguistic is acknowledged. I also verified the originality of contents through plagiarism software.

Amir Amin

273794-MS(IS)-11-2018F

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS thesis written by Mr. Amir Amin,
(Registration No 273794), of SEECs (School/College/Institute) has
been vetted by undersigned, found complete in all respects as per NUST Statutes/Regulations,
is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS
degree. It is further certified that necessary amendments as pointed out by GEC members of
the scholar have also been incorporated in the said thesis.

Signature: _____

Name of Supervisor: Dr. Omar Arif

Date: _____

Signature (HOD): _____

Date: _____

Signature (Dean/Principal): _____

Date: _____

APPROVAL

It is certified that the contents and form of the thesis titled “**Windows Registry: A Generalized Methodology based on Cross-Validated Forensic Analysis**” submitted by **Amir Amin** have been found satisfactory for the requirement of the degree.

Advisor: **Dr. Omar Arif**

Signature: _____

Date: _____

Committee Member 1: **Dr. Shahzad Saleem**

Signature: _____

Date: _____

Committee Member 2: **Dr. Sharifullah Khan**

Signature: _____

Date: _____

Committee Member 3: **Dr. Mehdi Hussain**

Signature: _____

Date: _____

ACKNOWLEDGEMENTS

I am thankful to Allah Subhana wa Taalah for being with me and nothing is possible without the will of Allah. I could not have achieved anything without Allah's ultimate support and blessings. I would like to thank my father and mother for much needed prayers and guidance throughout my life. They are such an inspiration, may Allah SWT bless them ever, Aameen. Thank you for being my parents. You have not only taught me academic subjects but the true meaning of being a Momin and a good human being. I also want to thank my wife and children for their immense support and patience, to my brothers and sisters and all family members for remembering me in prayers.

Special thanks to my Supervisor Dr. Omar Arif and Ex Supervisor and Mentor Dr. Shahzad Saleem who were always there to help, support and guide me during my thesis phase. I have learned a great deal from them. I feel so lucky while I count you people as my supervisors. Thank you

I would like to thank my committee members Dr. Sharifullah Khan; always big source of guidance and Dr. Mehdi Hussain; the one who simplified things for me during research activities.

I would like to express my gratitude to Pakistan Air Force (PAF) for giving me this opportunity while seeing confidence in me and permitting me to complete this honourable qualification. At the end I would like to give special thanks to my friends and fellows for those worth remembering moments we have enjoyed during our tenure in the campus.

Amir Amin

Dedication

This dissertation is dedicated to

My beloved parents, wife, children,
brothers, sisters, friends and specially
Pakistan Air Force.

Abstract

Enhancement in technology is rapidly increasing the usage of computer devices. With the increase in usage and due to the popularity of Microsoft Windows, more than 80% of computer users work on windows operating system that brings into play Windows Registry as a repository which keeps configuration of almost all applications. Following a Windows based digital crime; Data stored in Windows Registry is important for collecting evidence in most of the digital forensic investigations. Registry evidence helps in solving the puzzle of whom, what, when and how in forensics analysis. Collection of relevant artifacts from Windows Registry corpus is a cumbersome task which requires a lot of time and effort. In this research, a generalized methodology is introduced in the field of Windows Registry Forensics to collect forensic artifacts produced as a result of an examination performed on an application or activity with minimum contamination. The proposed methodology will define a simple way to perform Windows Registry forensics and will be helpful for researchers and forensic investigators working on Registry Forensics. Resulted methodology is produced after execution and comparison of different types of forensic tools.

Proposed methodology will be a mixture for multiple forensic tools which can be used in a way to efficiently extract and analyze the artifacts. Filtration and validation process is part of the methodology and will help in collection of most relevant and purified Windows registry artifacts. Digital forensic researchers can use such methodology to efficiently perform research in the field of Windows registry forensics to filter out most worthy registry values which will be revealing traces about the users' activities performed in a Windows based environment. It will simplify digital investigations related to Windows Operating System.

Keywords: *Digital Forensics, Digital Investigations, Windows Registry, Computer Forensics, Registry Forensics, Forensic Tools, Forensic Methodology.*

TABLE OF CONTENTS

Chapter 1

INTRODUCTION	1
1.1 Background.....	1
1.2 Significance of Digital Evidence.....	1
1.3 Computers and Digital Crimes	1
1.4 Microsoft Windows: The Most Popular Operating System	2
1.5 Microsoft Windows Registry.....	3
1.5.1 Forensic Importance of Registry	3
1.6 Registry Evolution.....	4
1.7 Registry Complications for Forensic Investigator	4
1.8 Problem Statement.....	5
1.9 Motivation	5
1.10 Solution Description.....	5
1.11 Organization of Thesis	5

Chapter 2

WINDOWS REGISTRY ARCHITECTURE.....	7
2.1 Introduction	7
2.2 Registry Hives.....	7
2.3 Registry Root Keys.....	8
2.4 Registry Hierarchical Structure	9
2.5 Registry Values.....	10

Chapter 3

RELATED WORK.....	12
3.1 Introduction	12
3.2 Digital Forensic: A Separate Field.....	12
3.3 Digital Forensic as a Methodology.....	12
3.4 Windows Registry Forensics	13
3.5 Latest Studies in Registry Forensic	14

3.6	Overview of Related Work	16
-----	--------------------------------	----

Chapter 4

REGISTRY FORENSIC TOOLS.....	17	
4.1	Introduction	17
4.2	Types of Registry Forensic Tools.....	17
4.2.1	Live Monitoring tool.....	17
4.2.2	Registry Snapshot tool.....	17
4.3	Characteristics of Registry Forensic Tools	18
4.3.1	Launch Process from Tool	18
4.3.2	Time Stamps	18
4.3.3	Relevant Data Filtering	18
4.3.4	Previous Values	18
4.3.5	Result Export Formats	18
4.4	Tools Comparison	19

Chapter 5

PROPOSED METHODOLOGY	21	
5.1	Introduction	21
5.2	Proposed Methodology	21
5.2.1	Before Activity Operations.....	21
5.2.2	After Activity Operations.....	22
5.2.3	Comparison	23
5.2.4	Filtration	23
5.2.5	Validation.....	24

Chapter 6

CASE STUDY: MICROSOFT OFFICE RESULTS	27	
6.1	Introduction	27
6.2	Results	28
6.2.1	Microsoft Office Suite Installation	29
6.2.2	Opening Microsoft Word, Excel and Power Point	31
6.2.3	Closing Microsoft Word, Excel and Power Point	33

6.2.4	Creating File	34
6.2.5	Accessing File	36
6.2.6	Modifying File	37
6.2.7	Closing File	38
6.2.8	Microsoft Office Suite Un-installation	39
6.3	How Proposed Methodology is Beneficial?	40

Chapter 7

LIMITATIONS, CONCLUSION AND FUTURE REQUIREMENTS	41
7.1 Limitations and Future Work	41
7.2 Conclusion.....	41
REFERENCES.....	42

LIST OF FIGURES

FIGURE 1. DEVICE DIVERSITY	2
FIGURE 2. OPERATING SYSTEMS MARKET SHARE	2
FIGURE 3. OS VERSIONS MARKET SHARE	3
FIGURE 4. WINDOWS REGISTRY EDITOR	8
FIGURE 5. WINDOWS REGISTRY STRUCTURE	10
FIGURE 6. VIRTUAL MACHINE FORENSICS METHODOLOGY	16
FIGURE 7. FLOW CHART FOR COLLECTING REGISTRY CHANGES OF A SINGLE ACTIVITY	22
FIGURE 8. COLUMN WISE SEPARATE WORKING OF EACH TOOL	24
FIGURE 9. VALIDATION PROCESS	25
FIGURE 10. MS OFFICE – ARTIFACTS COLLECTION PROCESS	27
FIGURE 11. REGFROMAPP RESULTS	28
FIGURE 12. MS OFFICE INSTALLATION - REGISTRY CHANGES	29
FIGURE 13. INSTALLER TIMESTAMP IN REGISTRY	30
FIGURE 14. SETUP DURATION TIME (SEC)	31
FIGURE 15. FIRST RUN REGISTRY CHANGES	32
FIGURE 16. CLOSING REGISTRY CHANGES	33
FIGURE 17. CREATING FILE REGISTRY CHANGES	34
FIGURE 18. FILE MRU DESCRIPTION	35
FIGURE 19. DOCUMENT RECOVERY KEY	36
FIGURE 20. READING LOCATION CURSOR	36
FIGURE 21. ACCESSING FILE REGISTRY CHANGES	37
FIGURE 22. MODIFYING FILE REGISTRY CHANGES	38
FIGURE 23. CLOSING FILE REGISTRY CHANGES	38
FIGURE 24. OFFICE UN-INSTALLATION REGISTRY CHANGES	39

LIST OF TABLES

TABLE 1.	LONGITUDINAL VIEW OF WINDOWS REGISTRY	4
TABLE 2.	INCREASE IN REGISTRY KEYS AND VALUES	4
TABLE 3.	REGISTRY HIVES WITH SUPPORTING FILES	7
TABLE 4.	DESCRIPTION OF REGISTRY ROOT KEYS	9
TABLE 5.	DESCRIPTION OF DIFFERENT REGISTRY VALUES	11
TABLE 6.	PHASES OF DIGITAL EVIDENCE COLLECTION METHODOLOGY	13
TABLE 7.	TOOLS ACQUISITION COMPARISON	14
TABLE 8.	TOOLS ACQUISITION COMPARISON USB DEVICES	15
TABLE 9.	TOOLS ACQUISITION COMPARISON MOBILE DEVICES	15
TABLE 10.	TOOLS COMPARISON BASED ON CHARACTERISTICS	19
TABLE 11.	INSTALLATION ARTIFACTS	29
TABLE 12.	FIRST TIME APP OPENING ARTIFACTS	32
TABLE 13.	APP CLOSING ARTIFACTS	33
TABLE 14.	FILE CREATING ARTIFACTS	35
TABLE 15.	ACCESSING FILE ARTIFACTS	37
TABLE 16.	OFFICE UN-INSTALLATION ARTIFACTS	39

INTRODUCTION

1.1 Background

Rapid development in the field of digitalization has produced a number of digital devices. The overwhelming usage of these devices has contributed handsomely in the routines of individuals. But on the other hand a new era of cyber criminals has evolved in which criminals perform their crime on digital devices rather than old fashion criminal activities. In this modern age, as the word crime is changed to digital crime, similarly, the word investigation is changed to Digital investigation. Digital investigation [1] is a method to deal with digital crimes which are increasing in numbers and severity. Digital investigations are as important as physical investigations to retrieve evidence in such a way that it can be presentable in court of law.

1.2 Significance of Digital Evidence

Significance of digital evidence is same as the significance of evidence of the tool used in a murder case but tempering or even destruction of digital evidence is much easier than physical evidence. The fragile nature of digital evidence [2] makes it prone to alteration, damage or destruction. Therefore negligence in handling digital evidence may lead to swear consequences. Therefore, digital evidence is to be handled with care and for the said purpose a chain of custody is to be maintained honestly.

1.3 Computers and Digital Crimes

Computers are often somehow involved in digital crimes because of being the most popular among digital devices (Figure 1). Instead of treating computers like an item it is treated as a secondary crime scene which often leads to valuable insights of crime.



FIGURE 1. DEVICE DIVERSITY

1.4 Microsoft Windows: The Most Popular Operating System

Among computers, windows is the operating system of choice for being user friendly and popular. Windows operating system has the largest market share [3] with more than 80% of computer users are using windows as shown in Figure 2.

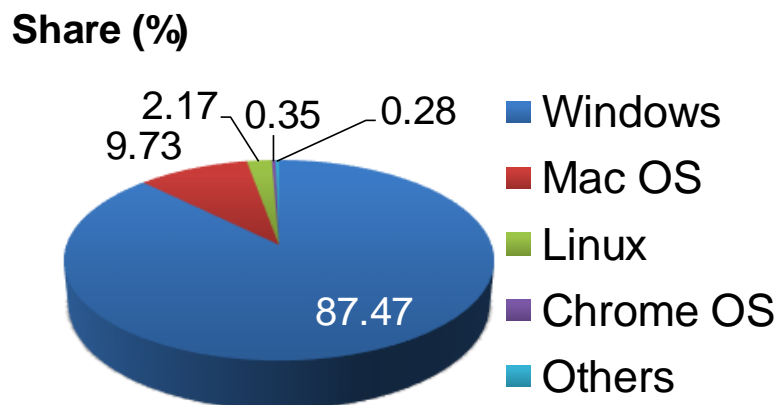


FIGURE 2. OPERATING SYSTEMS MARKET SHARE

A step further, Figure 3 shows the worldwide market share of top used operating system version. Windows 10 is the most widely used OS version and is chosen for the research purpose.

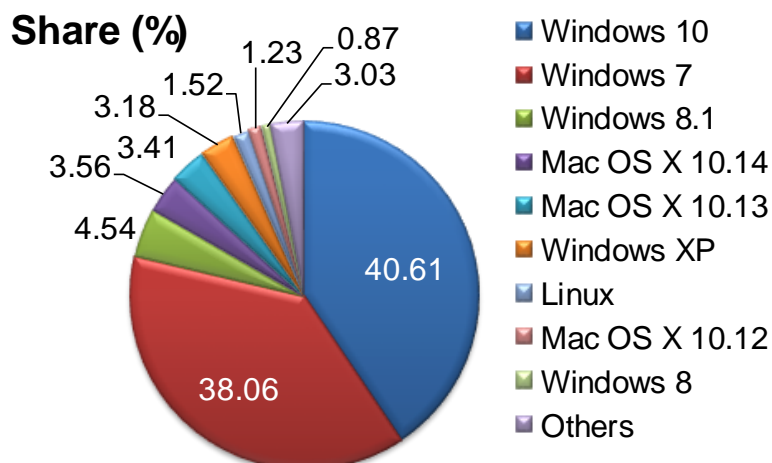


FIGURE 3. OS VERSIONS MARKET SHARE

1.5 Microsoft Windows Registry

Just as Linux contains configuration of every program in files, Windows provides Registry as a central configuration database [4] that different applications use to store their configurations as well as application usage history and other relevant data. More on Windows registry architecture is discussed in Chapter 2.

1.5.1 Forensic Importance of Registry

The data stored in registry can be of great forensic value [5][6] since in some cases even after uninstallation and removal of a program, important artifacts are still retained in the registry. Thus registry analysis can help counter anti-forensic activities [7] and produce forensically sound artifacts. Due to high intensity usages of Microsoft Windows, high number of cybercrimes executed using Windows operating systems. So, most of the computer investigations revolves around Windows operating system registry.

1.6 Registry Evolution

Avinash Singh et al. [8] provided the longitudinal view of Windows registry over the evolution of Windows versions as shown in the Table 1.

TABLE 1. LONGITUDINAL VIEW OF WINDOWS REGISTRY

Hive Files	Windows 95	Windows 98	Windows 2000	Windows XP	Windows VISTA	Windows 7	Windows 8	Windows 10
BCD	-	-	-	-	✓	✓	✓	✓
DRIVERS	-	-	-	-	-	-	-	✓
SAM	-	-	✓	✓	✓	✓	✓	✓
SECURITY	-	-	✓	✓	✓	✓	✓	✓
SOFTWARE	-	-	✓	✓	✓	✓	✓	✓
SYSTEM	-	-	✓	✓	✓	✓	✓	✓
DEFAULT	-	-	✓	✓	✓	✓	✓	✓
COMPONENTS	-	-	-	-	✓	✓	✓	✓
NTUSER.DAT	-	-	✓	✓*	✓*	✓*	✓*	✓*
USRCLASS.DAT	-	-	-	✓*	✓*	✓*	✓*	✓*
SYSTEM.DAT	✓	✓	-	-	-	-	-	-
USER.DAT	✓	✓	-	-	-	-	-	-
Policy.pol	-	✓	-	-	-	-	-	-

Items marked with a (*) may have more than one location for the file

1.7 Registry Complications for Forensic Investigator

Amir Amin et al. [9] mentioned a considerable increase in default registry keys and values of latest Windows operating systems as shown in Table 2.

TABLE 2. INCREASE IN REGISTRY KEYS AND VALUES [9]

Registry Hive	Windows 10		Windows 7		% Difference	
	Keys	Values	Keys	Values	Keys	Values
HKLM	568,162	343,200	354,553	217,193	+160%	+158%
HKCR	187,458	161,053	113,642	94,597	+165%	+170%
HKU	29,505	13,806	7,182	2,554	+411%	+540%
HKCU	10,563	5,237	4,486	1,906	+235%	+275%

1.8 Problem Statement

“In order to solve a Digital Forensic case, acquisition of forensically sound artifacts is very important for attribution. Unplanned and inefficient collection of such artifacts is very cumbersome and time consuming due to the huge volume of registry values and it keeps on increasing with the evolution of Windows and registry versions.”

1.9 Motivation

Different tools exist to perform forensic analysis of a windows registry and each tool has its own capabilities i.e. Timestamps, paths etc. as well as inabilities i.e. garbage values, irrelevant read operations, insufficient data etc. Problem is to use multiple registry forensic tools in a way that puts away their inabilities and consume their capabilities to generate valuable artifacts which will be helpful in digital investigations.

1.10 Solution Description

“To define a methodology which uses capabilities of multiple registry forensic tools in a way to collect forensically sound registry artifacts from a number of windows registry keys and values. It will help researchers working in the field of Windows registry forensic.”

This research is performed by considering the capabilities of multiple registry forensic tools and a methodology is proposed to cope up with the inabilities of these tools by cross comparing the outputs to help Windows registry forensic analysts. A comparison of considered registry tools is also provided to understand the usage of multiple tools. For forensic investigator, registry is a collection of valuable evidence that can solve the puzzle of digital investigation. At the end, a case study is performed to collect results and evaluate the usage of proposed methodology.

1.11 Organization of Thesis

This thesis is distributed into 7 concise chapters. Each chapter covers the different aspect of the research as given below:-

Chapter 1: Introduces the paradigm of the research.

Chapter 2: Provides the brief architecture of Microsoft Windows Registry.

Chapter 3: Gives an overview of related work performed in the same field.

Chapter 4: Elaborates the types and characteristics of registry forensic tools along with comparison of selected tools for this research.

Chapter 5: Proposes a generalized methodology for efficient collection of forensic artifacts from Windows Registry.

Chapter 6: Shows results of a case study performed on Microsoft Office by opting the proposed methodology.

Chapter 7: Concludes the research study including limitations and future possible work in the same field.

WINDOWS REGISTRY ARCHITECTURE

2.1 Introduction

In the 5th edition of Microsoft Computer Dictionary it is defined as [10] a central hierarchical database in Windows which keeps essential configuration information of hardware, software and users. The architecture of Windows Registry is defined in details by Microsoft [11]. It is the main location which keeps the configuration of Windows as well as the applications which are installed, running or even uninstalled from the OS.

2.2 Registry Hives

Registry contains multiple hive files. Each hive file is consist of *keys*, *subkeys* and *values*. A registry includes supporting files which are stored in the *Windows/System32/Config* folder except *NTUser.dat* which is separate for each user and is stored in respective user’s profile. Table 3 shows the Registry Hives along with supporting files against each.

TABLE 3. REGISTRY HIVES WITH SUPPORTING FILES

Registry Hive	Supporting Files
HKEY_LOCAL_MACHINE\SAM	Sam, Sam.log, Sam.sav
HKEY_LOCAL_MACHINE\Security	Security, Security.log, Security.sav
HKEY_LOCAL_MACHINE\Software	Software, Software.log, Software.sav
HKEY_LOCAL_MACHINE\System	System, System.alt, System.log, System.sav
HKEY_CURRENT_CONFIG	System, System.alt, System.log, System.sav, Ntuser.dat, Ntuser.dat.log
HKEY_USERS\DEFAULT	Default, Default.log, Default.sav

2.3 Registry Root Keys

Windows has its own built-in registry editor which can be accessed by typing *regedit* in the command prompt. Registry editor helps windows users to view the current registry configurations and settings. On opening the registry editor, a windows will appear which shows the registry root keys as shown in Figure 4. Table 4 describes the purpose of each of the visible root key [12].

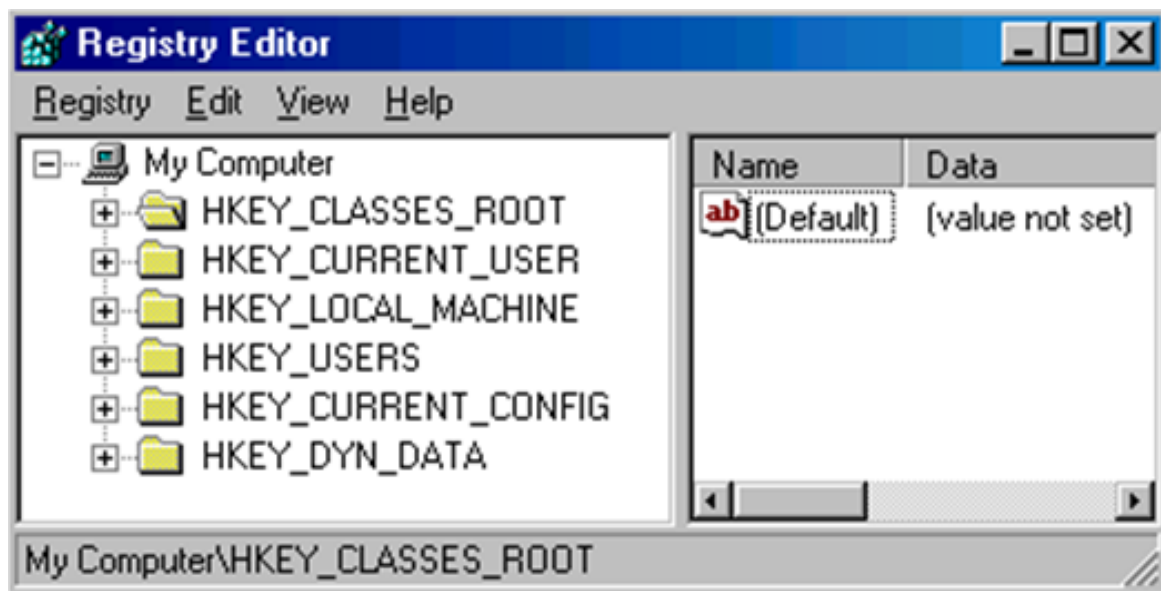


FIGURE 4. WINDOWS REGISTRY EDITOR

TABLE 4. DESCRIPTION OF REGISTRY ROOT KEYS

Root Key	Description
HKCR (HKEY_CLASSES_ROOT)	Describes file type, file extension , and OLE information.
HKCU (HKEY_CURRENT_USER)	Contains user who is currently logged into Windows and their settings.
HKLM (HKEY_LOCAL_MACHINE)	Contains computer-specific information about the hardware installed, software settings, and other information. The information is used for all users who log on to that computer. This key, and its subkeys, is one of the most frequently areas of the registry viewed and edited by users.
HKU (HKEY_USERS)	Contains information about all the users who log on to the computer, including both generic and user-specific information.
HKEY_CURRENT_CONFIG (HKCC)	The details about the current configuration of hardware attached to the computer.
HKDD (HKEY_DYN_DATA)	Only used in Windows 95, 98, and NT, the key contained the dynamic status information and plug and play information. The information may change as devices are added to or removed from the computer. The information for each device includes the related hardware key and the device's current status, including problems.

2.4 Registry Hierarchical Structure

Windows registry is in the form of a hierarchical tree like database which keeps track of the configuration information related to system, users, applications, devices [4]. Figure 5 shows a tree view of series of folders (Hives / Keys), values and data against each value.

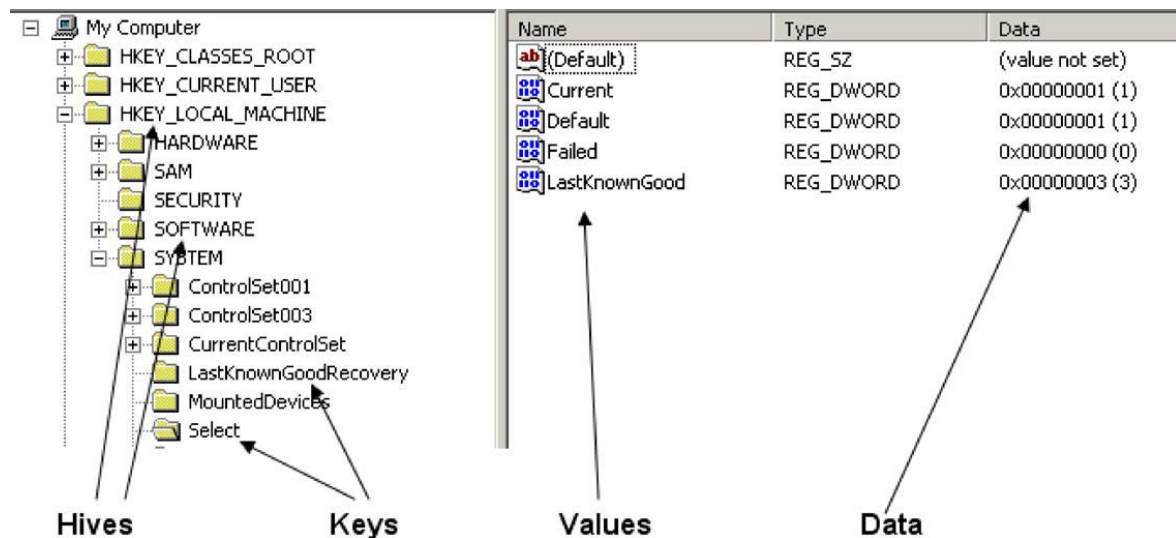










FIGURE 5. WINDOWS REGISTRY STRUCTURE

On installation of Microsoft Windows a number of registry keys and values are by default available which keeps the configuration settings of the Windows OS in use. These keys and values are keep on changing and increasing with the evolution in Windows OS versions. But the most important information is added when the user / suspect starts installing his required applications in the Windows OS. On installation of application, configuration settings are written in the registry and system takes help from those entries whenever some action is performed using that application. To increase the interest, it is important to mention that some of the application's data is remained in the registry even after the uninstallation of the software. Similarly there may be few applications whose data is not even written in the registry or data is of not much of an importance. Individual user activities can also be sorted out from available registry data and values to find out the usage of a particular user against a windows based computer system.

2.5 Registry Values

There are multiple types of values can be seen in the registry editor. Each of these values have different nomenclature as defined with description against each in the Table 5 [12].

TABLE 5. DESCRIPTION OF DIFFERENT REGISTRY VALUES

Icon	Type	Name	Description
		Closed key	Like the folders seen in Windows Explorer. These keys are what contain the registry subkeys mentioned below.
		Open key	When a key is opened, the icon changes to an expanded or open folder and displays all its contents and any additional subkeys.
	REG_SZ	String value	Allows for any string value to be defined on a single line, such as a file path, and is the most commonly found subkey in the registry.
	REG_MULTI_SZ	String array value	Any multi-line string value.
	REG_EXPAND_SZ	Expanded string value	Contains a string with environmental or system variables that need to be expanded. For example, c:\%windir%\example.exe could be the same as C:\windows\example.exe.
	REG_BINARY	Binary value	Allows for attributes to be defined in binary as either on or off (0 or 1).
	REG_DWORD	DWORD value	Similar to the binary value, but capable of values being defined in either 32-bit decimal or hex .
	REG_QWORD	QWORD value	Like the DWORD, but stored as a 64-bit value.

RELATED WORK

3.1 Introduction

With the rapid technology evolution in digital devices and due to the increase number of computer users, most of the data is travelling on the digital devices involving computer with the maximum share of digital data. On the other hand criminals are also adapting the modern technologies and techniques to remain one step ahead by committing digital crimes to surprise the world. Nowadays, digital crimes are growing in numbers and hence making the Digital Forensic a well-known and reputed field. In related word, a detailed literature survey is performed in the field of Windows Registry Forensics. There are a lot of relevant studies carried out in the same context. These studies are divided according to their nature and are discussed in the subsequent paragraphs.

3.2 Digital Forensic: A Separate Field

In 2008, Darek Bem et al. [13] described past, present and future of computer forensics and emphasizes on considering computer forensic as another field in science. He pointed out the unnoticeable gap between a computer crime and its counter measures. To mitigate digital crimes, standards and policies are needed to be defined to help digital investigations. Furthermore, reconstruction of digital crime scene is still needed to be addressed as per some defined standards. In the study, current and future challenges are also discussed including the most important digital forensic requirement of keeping the digital evidence intact.

3.3 Digital Forensic as a Methodology

In 2011, Peter Cisar et al. [14] provided a Digital Forensic analysis methodology to address the issue of multiple methodologies applied in the field of digital investigations. He defined a methodology for each of the three essential processes in the digital forensics field i.e. (1) Preparation / Extraction, (2) Identification and (3) Analysis.

In 2012, Sabah Al-Fedaghi et al. [15] proposes an abstract model of digital investigation which includes six generic internal operations i.e. arrive, accept, process, release, create and transfer. In 2018, Nik Zulkarnaen Khidzir et al. [16] suggested autopsy forensic tools which keeps authenticity of digital evidence by calculating MD5 hashes. The study divided the process into eight phases as shown in Table 6.

TABLE 6. PHASES OF DIGITAL EVIDENCE COLLECTION METHODOLOGY

No. of Phase	Digital Evidence Collection Methodology
Phase 1	Confirmation
Phase 2	System Explanation
Phase 3	Proof Acquisition
Phase 4	Timeline Evaluation
Phase 5	Mass Media Artifact Evaluation
Phase 6	Sting Byte Search
Phase 7	Data Collection
Phase 8	Reporting Result

However all of these proposed methodologies or models do not define any procedure for windows registry forensics to perform a research on the basis of activities of a user or an application.

3.4 Windows Registry Forensics

In 2013, Raihana Md Saidi et al. [17] investigates windows 7 registry for illegal activities of an attacker using Virtual Network Computing (VNC) and Keylogger applications. As a result of the study, they provided the registry values as evidence of VNC accessing a Windows 7 computer in different scenarios. Yet Another Registry Utility (YARU) is used by the researchers to analyze the data and to find the footprints on VNC activities in targeted Windows registry. In 2013, Sriram Raghavan et al. [18] provided a study on a variety of

forensic and analysis tools which also include a detailed comparison of these tools related to computer forensics. Encase Forensic, FTK, RegRipper and Wireshark are few of the famous tools used in the study. But windows registry forensics are not focused in the study. In 2018, Muhammad Nur Faiz et al. [19] performed a comparison (shown in Table 7) of acquisition software for the purpose of digital investigations in the field of live forensics. Live forensics are performed on the volatile data available in the RAM which is vanished as the operating system is shutdown or restarted.

TABLE 7. TOOLS ACQUISITION COMPARISON

Tools	Memory Usage (Mb)	Processing Time (second)	Registry Key	DLL
FTK Imager	117	198.65	59	270
Belka RAM Capturer	18	186.22	9	56
Magnet RAM Capture	76	220.24	98	285
Dumplt	10	185.6	4	44
Memoryze	13	184.54	7	71

In abovementioned studies, tools comparisons are discussed but usage of multiple tools in a way to collect filtered results and that too in Windows registry domain is not provided.

3.5 Latest Studies in Registry Forensic

Keeping in view the latest research work on windows registry forensics, in 2017, Ayesha Arshad et al. [20] performed USB and Mobile device forensics on different versions of Windows i.e. 7, 8 and 10 by gathering registry and event logs data on performing 3 activities on USB i.e. before insertion, during insertion and after removal. Table 8 showing the presence of artifacts in case of USB devices and Table 9 showing the artifacts in case of Mobile devices. But study only tracks USB relevant registry values.

TABLE 8. TOOLS ACQUISITION COMPARISON USB DEVICES

	Key/Subkey	Win 7	Win 8	Win 10
First insertion time from DeviceClasses key in System Hive	10497b1b-ba51-44e5-8318-a65c837b6661	✓	✓	✓
	53f56307-b6bf-11d0-94f2-00a0c91efb8b	✓	✓	✓
	53f5630d-b6bf-11d0-94f2-00a0c91efb8b	✓	✓	✓
	65a9a6cf-64 cd-480b-843e-32c86e1ba19f	✓	✓	✓
	6ac27878-a6fa-4155-ba55-f95f491d4f33	✓	✓	✓
	7f108a28-9833-4b3b-b780-2c6b5fa5c062	✓	✓	✓
	7fcc86c-228a-40ad-8a58-f590af7bfdce	✓	✓	✓
	a5dcbf10-6530-11d2-901f-00c04fb951ed	✓	✓	✓
	EEC5AD98-8080-425f-922A-DABF3DE3F69A	✓	✓	✓
First insertion time from System Hive Under USBSTOR Property Key	f33fdc04-d1ac-4e8e-9a30-19bbd4b108ae	✓	✓	✓
	0003	✓	✓	✓
	000A	✓	✓	✓
	0064	✓	✓	✓
Last insertion time from System Hive Under USBSTOR Property Key	0065	✓	✓	✓
	0066	✓	✓	✓
Last removal time from System Hive Under USBSTOR Property Key	0067	✓	✓	✓
	0067	✓	✓	✓
First insertion time from System Hive under USB Key	VID_[VendorID]&PID_[ProductID]\{SerialNo}\DeviceParameters	✓	✓	✓
	\\e5b3b5ac-9725-4f78-963f-03dfb1d828c7	✓	✓	✓
Last insertion time from System Hive under USB Key	VID_[VendorID]&PID_[ProductID]\{SerialNo}\	✓	✓	✓
	Microsoft\Windows Portable Devices\Devices	✓	✓	✓
First insertion time from Software Hive	\\SWD#WPDBUSENUM#_??_ USBSTOR#DISK&VEN_[VenderName]	✓	✓	✓
	&PROD_[ProductName]&REV_PMAP#{SerialNo}#{53F56307-B6BF-11D0-94F2-00A0C91EFB8B}	✓	✓	✓

TABLE 9. TOOLS ACQUISITION COMPARISON MOBILE DEVICES

	Key/Subkey	Win 7	Win 8	Win 10
First insertion Time from DeviceClasses key in System Hive	10497b1b-ba51-44e5-8318-a65c837b6661	✓	✓	✓
	6ac27878-a6fa-4155-ba55-f95f491d4f33	✓	✓	✓
	6bdd1fc6-810f-11d0-bec7-08002be2092f	✓	✓	✓
	a5dcbf10-6530-11d2-901f-00c04fb951ed	✓	✓	✓
	EEC5AD98-8080-425f-922A-DABF3DE3F69A	✓	✓	✓
	f33fdc04-d1ac-4e8e-9a30-19bbd4b108ae	✓	✓	✓
First insertion time from System Hive Under USB Property Key	0003	✓	✓	✓
	0007	✓	✓	✓
	0008	✓	✓	✓
	0009	✓	✓	✓
	000A	✓	✓	✓
	0064	✓	✓	✓
	0065	✓	✓	✓
Last insertion time from System Hive Under USB Property Key	0066	✓	✓	✓
	0067	✓	✓	✓
Last removal time from System Hive Under USB Property Key	0067	✓	✓	✓
	0067	✓	✓	✓
First insertion time from System Hive under USB Key	VID_[VendorID]&PID_[ProductID]\{SerialNo}\DeviceParameters	✓	✓	✓
	\\e5b3b5ac-9725-4f78-963f-03dfb1d828c7	✓	✓	✓
Last insertion time from System Hive under USB Key	VID_[VendorID]&PID_[ProductID]\{SerialNo}\	✓	✓	✓
	Microsoft\Windows Portable Devices\Devices	✓	✓	✓

Study of Hasan Binjuraid et al. [21] in 2018, targets the changes in Windows 10 registry by performing analysis on the basis of two different cybercrime cases i.e. Use of BitTorrent

clients for downloading illegal or copyrighted data and data theft using USB devices. Due to increased use of virtualization, Virtual Machine forensic analysis has become an important area of research from forensic point of view. In 2018, Erfan Wahyudi et al. [22] provides forensic analysis of virtual machine to get digital evidence and even recovery of deleted virtual machine with the help of forensic artifacts. They collected the registry artifacts by using *Regshot* registry forensic tool by comparing snapshots before installation of *VirtualBox* and after deletion of created virtual machine. Figure 6 shows the proposed methodology from the study.

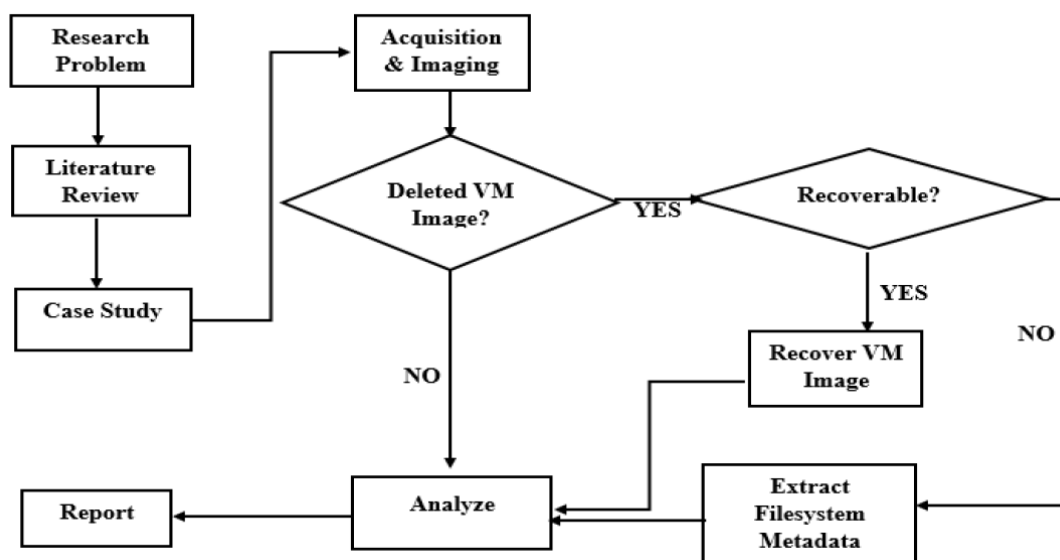


FIGURE 6. VIRTUAL MACHINE FORENSICS METHODOLOGY

3.6 Overview of Related Work

Literature review reveals that a lot of work has been performed in the field of digital forensics to help digital investigations. However, a generalized methodology for analysis of windows registry is not yet proposed to help digital forensic researchers. Keeping in view the importance of Windows registry artifacts, a simple way is introduced in this research to collect filtered and cross validated Windows registry values while performing an activity. Proposed technique will help future studies on monitoring registry activities of different applications or users.

REGISTRY FORENSIC TOOLS

4.1 Introduction

This chapter provides a detailed overview of registry forensic tools while keeping in view their types and characteristics. Depending upon the requirements of forensic analysis, selection of an ideal tool is an important task which will make the rest of the analysis much easier and will help in fulfilling the purpose of the research. Ideal tool(s) can be selected on the basis of their types and characteristic as explained in the subsequent paragraphs. A comparison of selected tools is also provided in this chapter.

4.2 Types of Registry Forensic Tools

There are two types of Registry Forensic Tools; Live Monitoring tool and Registry Snapshot tool.

4.2.1 Live Monitoring tool

It has the capability of monitoring live activities performed by a running process in windows. Process activities make changes in the registry which is monitored live by these type of tools. Results can be saved for forensic analysis.

4.2.2 Registry Snapshot tool

Taking snapshot of Windows registry means current state of windows registry is captured (which includes all the available registry configurations) at a point in time. Usually snapshot tools compare images of the whole registry before and after performing an activity and helps in identifying changes made in the registry due to the performed activity.

4.3 Characteristics of Registry Forensic Tools

There can be many characteristics of a registry forensic tool but few important characteristic which helps in selecting the tool are defined as:

4.3.1 Launch Process from Tool

It is the capability of tool to launch the process required to be analyzed. This feature will start monitoring the process from the scratch. This characteristic is the specialty of Live Monitoring tools.

4.3.2 Time Stamps

It gives the time stamps of registry keys and values i.e. previous modified time and new modification time etc. It is an important feature of any type of forensic tool.

4.3.3 Relevant Data Filtering

Relevant data filtering means that the tool is capable of monitoring and filtering out the activities performed by inspected process only. Mostly included in Live monitoring tools.

4.3.4 Previous Values

These are the old registry values before inspected process performed modification. These values are produced by the tools having snapshot capability.

4.3.5 Result Export Formats

Results are the key in registry forensics. Results can be exported in the form of reports in different formats i.e. XML, HTML, CSV etc. This characteristic belongs to the report formats supported by the forensic tool.

On the basis of types and characteristics of registry forensic tools, a comparison can be performed which helps in selecting an appropriate tool which is useful in collecting forensic artifacts.

4.4 Tools Comparison

Plenty of windows registry tools are available for examining windows registry. In this research, renowned windows registry forensic tools are utilized i.e. *ProcMon* [23], *RegFromApp* [24], *RegShot* [25] and *RegChangeVwr* [26]. First two connect with Process ID (PID) and monitor process activities performed in registry, later two takes snapshot of the registry before and after performing activity and collects the changes after comparing both snapshots. These tools have some of the characteristics as mentioned above. Table 10 shows the characteristics comparison of each of these four tools:-

TABLE 10. TOOLS COMPARISON BASED ON CHARACTERISTICS

	<i>ProcMon</i>	<i>RegFromApp</i>	<i>RegShot</i>	<i>RegChangeVwr</i>
Live Monitor	✓	✓	✗	✗
Registry Snapshot	✗	✗	✓	✓
Launch Process from Tool	✗	✓	✗	✗
Time Stamps	✓	✗	✗	✓
Relevant Data Filtering	✓	✓	✗	✗
Previous Values	✗	✗	✓	✓
Result Export Formats	CSV, PML, XML	REG	TXT, HTML	CSV, TXT, HTML, XML, REG

From the comparison table it is obvious that *ProcMon* and *RegFromApp* tools have the capability of live monitoring which will give us results specific to the application or process we are monitoring. On the other hand *Regshot* and *RegChangeVwr* works on the basis of taking complete snapshot of the registry at a given time and comparison of two snapshots will include

registry changes not only made by our desired application or process but also the changes made by other applications or processes running at the same time for example system processes. Time stamps are very important capability which helps a lot in temporal analysis. It is available in *ProcMon* and *RegChangeVwr* which makes these tools more valuable. *ProcMon* and *RegChangeVwr* allow users to save its results in CSV file format, which will be helpful in further filtration process. While HTML format offered by *Regshot* is good for viewing results in a browser. The results of *ProcMon*, *Regshot* and *RegChangeVwr* provides the valuable information about registry changes in form of six different categories i.e. *Keys Added*, *Values Added*, *Keys Modified*, *Values Modified*, *Keys Deleted* and *Values Deleted*. But *RegFromApp* results does not include any valuable information i.e. Timestamps, keys added or deleted or modified etc. which makes it difficult to understand its results and so comparison with other tools is unjustified.

After detailed discussion on the characteristics and capabilities of above mentioned tools, an efficient methodology can be introduced to define a mechanism for using mixture of these tools to produce filtered and valuable Windows registry forensic artifacts.

PROPOSED METHODOLOGY

5.1 Introduction

Keeping in view the huge amount of registry data scattered under the hierarchical tree of Windows registry, it is important to define a simple way, which should be generic in nature so that it can be applied on any type of research performed on Windows registry forensics. Therefore, a methodology is proposed in this chapter which works around the capabilities of each tool discussed in the previous chapter. The results produced are cross-validated between tools and thus filtered out to collect relevant forensically sound artifacts.

5.2 Proposed Methodology

Keeping in view the activities of a normal user who is working on a Windows OS, a systematic methodology is opted to monitor and record the changes produced as a result of such activities. Routine operations of a user may involve using Microsoft Office, watching videos on Windows Media Player etc. Figure 7 shows a sequential diagram which depicts the way information is collected about registry operations of a single activity. The operations are divided into five major categories i.e. Before Activity, After Activity, Comparison, Filtration and Validation.

5.2.1 Before Activity Operations

In order to test the impact of a single activity on Windows Registry keys and values, it is important to perform before activity operations. It includes starting of live monitoring tools i.e. *ProcMon* and *RegFromApp* and taking snapshot using *RegShot* and *RegChangeVwr*. Now launch the required *Application* (to be analyzed) in *RegFromApp* tool console. After starting the *Application*, monitor the activities of *Application's PID* in *ProcMon*. Next step is to perform the activity which requires to be analyzed.

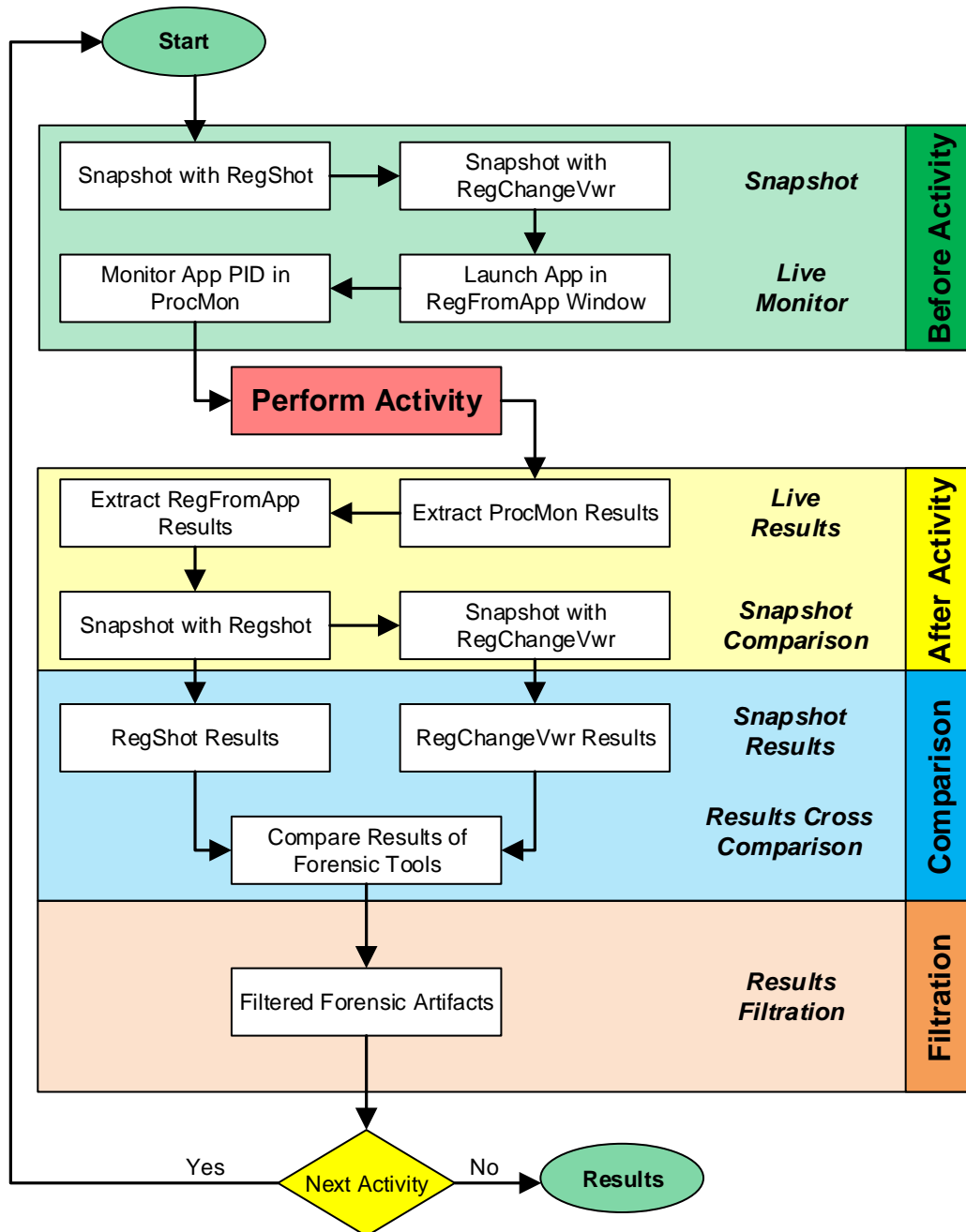


FIGURE 7. FLOW CHART FOR COLLECTING REGISTRY CHANGES OF A SINGLE ACTIVITY

5.2.2 After Activity Operations

The impact of the completed activity is recorded in Windows Registry. After performing the activity first of all extract the results of *ProcMon* and *RegFromApp* tools and then take the after activity snapshot of the registry using *RegShot* and *RegChangeVwr* tools.

5.2.3 Comparison

Compare the two snapshots taken before and after the activity using both *RegShot* and *RegChangeVwr* tools. A delta is generated showing changes made due to the activity performed between snapshots. It is necessary to mention that extra values (noise) are added due to the activities of other processes i.e. system processes etc. By now we have extracted results from all four tools and are in a position to compare those results to filter out the required ones.

5.2.4 Filtration

In the comparison phase, the results of the mentioned tools are compared to find out the forensically sound artifacts which will be helpful in forensic investigations. Finding such artifacts can be done by filtering out the ones which shows attribution to something i.e. attribution to person, things, date, event, file locations etc. Forensically sound artifacts are separated during the process of filtration.

Figure 8 illustrates column wise separate functioning of these tools and shows how each tool is contributing in collecting forensically sound registry artifacts. Any tool can be used to perform the registry forensic operation single handedly i.e. *ProcMon* can be used alone to perform an activity and monitor its changes using its PID. But *ProcMon* also shows results of Application Programming Interface (API) calls of read operations, which are not related to changes made on performing the activity. So, such values are not important. Similarly, *Regshot* and *RegChangeVwr* compares the registry snapshots taken just before and immediately after an action is performed by the user, but still it may contain noise due to the activities of parallel processes. To eliminate such irrelevant forensic artifacts multiple tools are used and results of each tool are cross compared to filter out the relevant forensic artifacts.

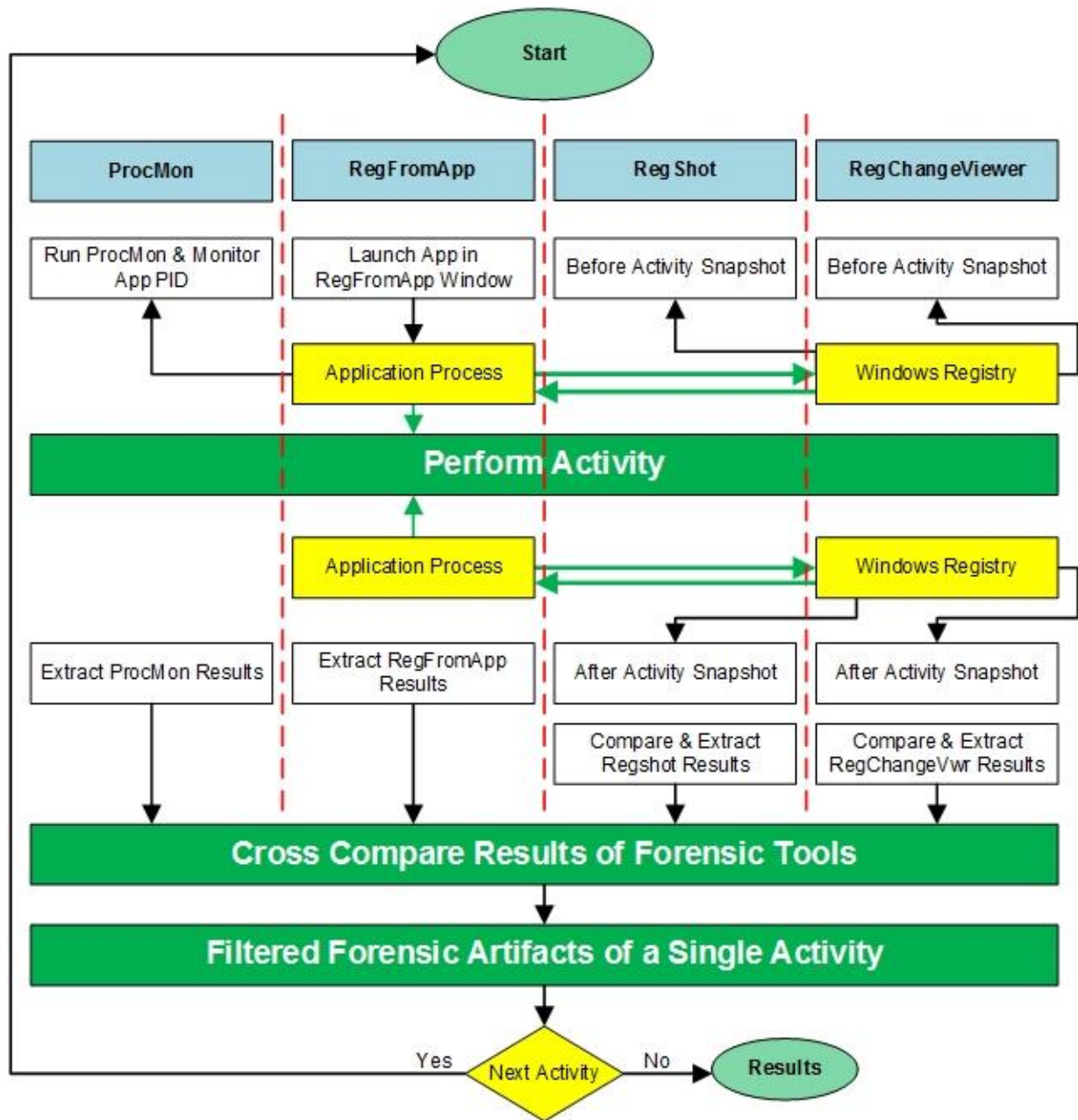


FIGURE 8 COLUMN WISE SEPARATE WORKING OF EACH TOOL

5.2.5 Validation

In order to find out the originality of collected artifacts, validation is necessary. Figure 9 shows the process of cross checking the filtered registry values with current registry by exporting relevant registry hive files by means of *FTK Imager* tool and then hives can be viewed in *Registry Viewer* software to check validity and produce rigger.

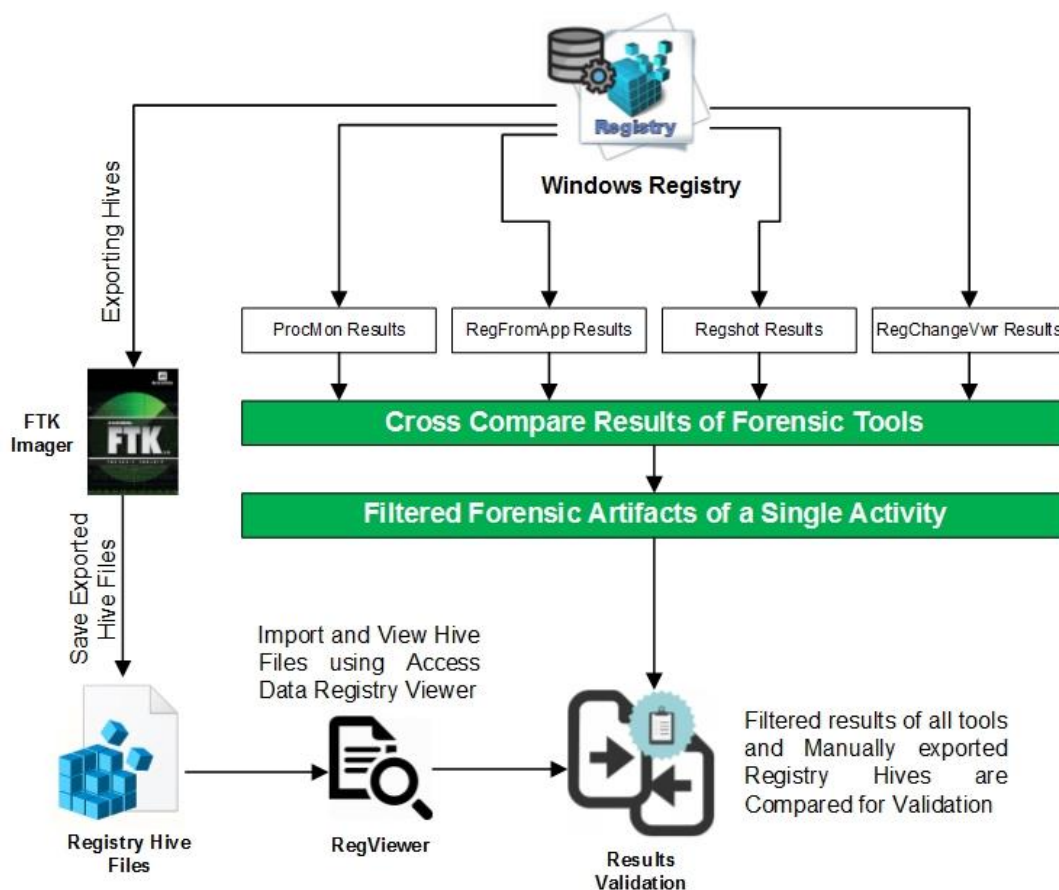


FIGURE 9 VALIDATION PROCESS

Furthermore, to elaborate the working of proposed methodology a case study is performed on Microsoft Office as a test case. Most commonly used office applications i.e. Word, Excel and Power Point are selected to perform different activities. These are the daily routine activities a normal user carries out. These operations are characterized as:-

- Microsoft Office Suite Installation
- Opening Word, Excel and Power Point
- Closing Microsoft Word, Excel and Power Point
- Creating File
- Accessing File
- Modifying File
- Closing File
- Microsoft Office Suite Un-installation

By following the proposed procedure and by using tools mentioned in this paper, filtered and validated results are collected for each of the operation mentioned above. These results are the proof of our proposed methodology which may be adopted in digital forensic investigations involving Windows registry.

CASE STUDY: MICROSOFT OFFICE RESULTS

6.1 Introduction

The primary objective of the research is to propose a methodology and then associate its effectiveness by performing a case study to collect filtered and validated results while performing different activities in Microsoft Word, Excel and Power Point as illustrated in Figure 10. Results will help to find answers of what, when, whom, where and how [27] in digital investigations.

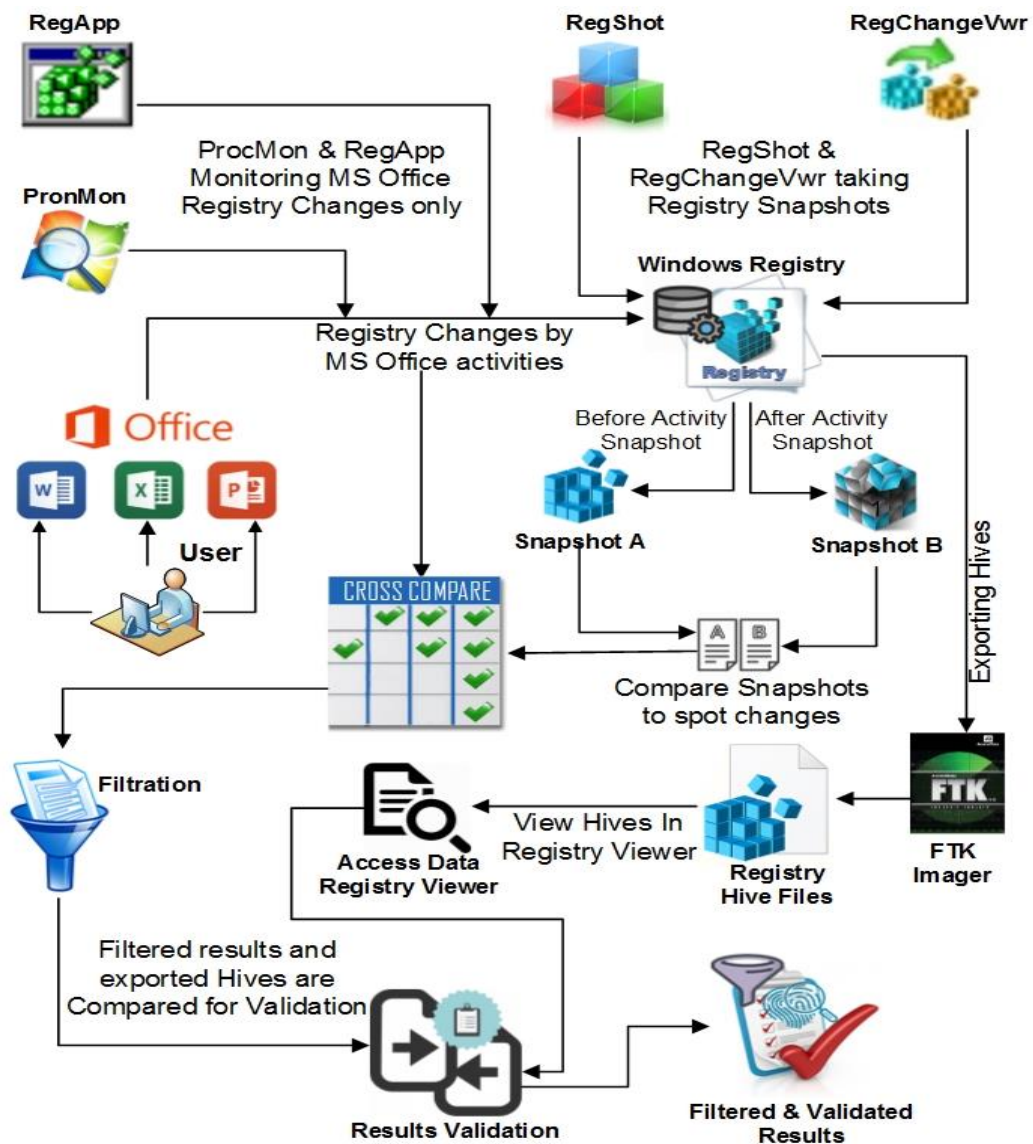


FIGURE 10 MS OFFICE – ARTIFACTS COLLECTION PROCESS

6.2 **Results**

By using proposed methodology, a number of keys and values are added, modified or deleted. As mentioned in Chapter 4, we have 6 different categories of registry changes i.e. *Keys Added, Values Added, Keys Modified, Values Modified, Keys Deleted, Values Deleted*. Out of these categories *Regshot* and *RegChangeVwr* show results of all categories except *Keys Modified* while *ProcMon* depicts results of each category. Meanwhile *RegFromApp* only shows results in the form of keys and values but does not depict whether these values are added, modified or deleted. However on comparing its results with other tools it can be seen that *RegFromApp* shows those values which are either added or modified. Thus, results of *RegFromApp* shown separately in Figure 11 and results depict that Microsoft Word has huge number of values against activities of closing word, accessing file and modifying file. It is because of 260+ word fonts are added or modified in the registry while performing mentioned activities. On the other hand activity wise registry changes against rest of the tools are mentioned below as overall results:

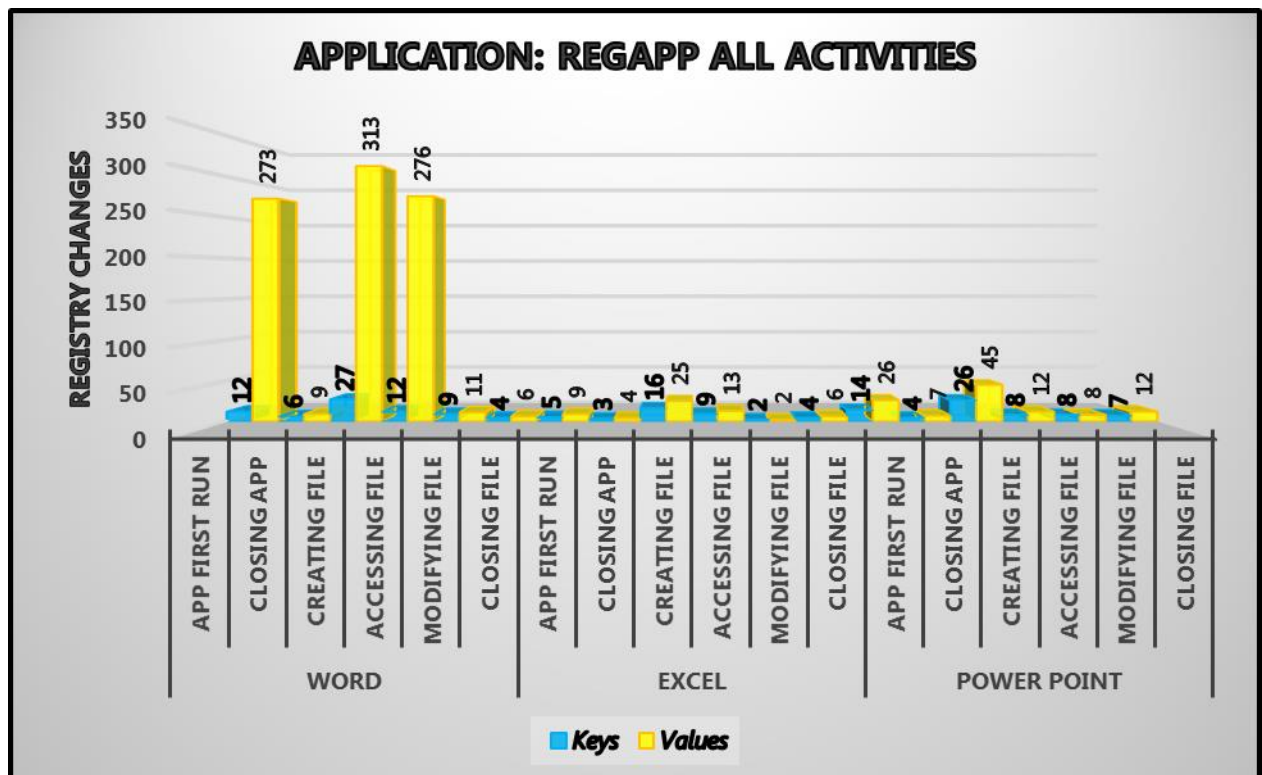


FIGURE 11 REGFROMAPP RESULTS

6.2.1 Microsoft Office Suite Installation

A number of registry changes occurred against each tool when installation of Microsoft Office Suite takes place, as shown in Figure 12.

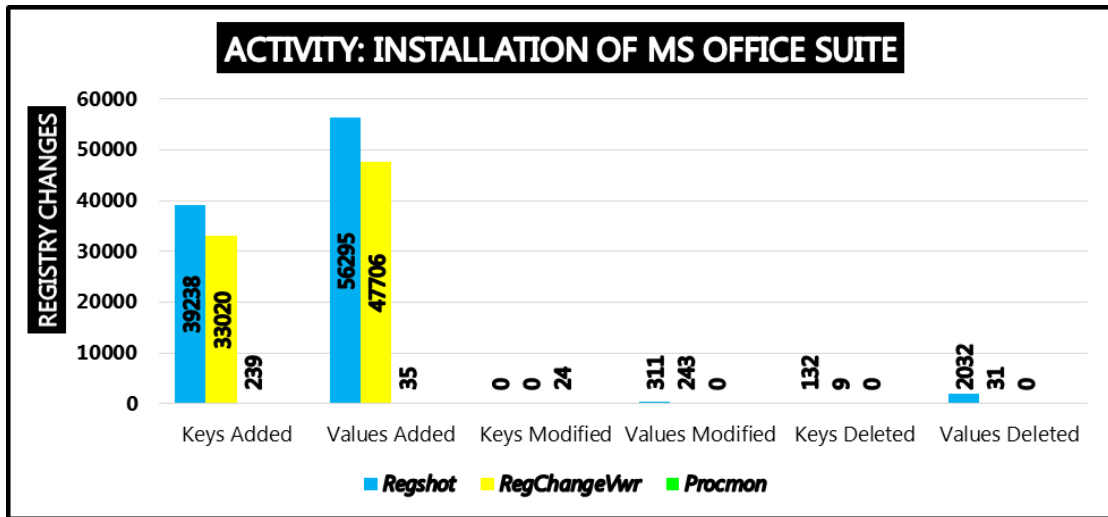


FIGURE 12. MS OFFICE INSTALLATION - REGISTRY CHANGES

Keeping in view the nature of the activity, it is obvious that most the registry changes belongs to the *Keys Added* and *Values Added*. *ProcMon* has less number of values due to only monitoring the relative process’s process ID, so noise is not added in the results. After applying the proposed methodology while performing the installation activity filtered and validated results are generated and shown in Table 11.

TABLE 11. INSTALLATION ARTIFACTS

Artifact	Registry Path
Timestamp	HKEY_LOCAL_MACHINE\System\ControlSet001\Services\bam\State\UserSettings\ {User SID}
Product Name	HKEY_LOCAL_MACHINE\Software\Microsoft\Office\15.0\Registration\{064383FA-1538-491C-859B-0ECAB169A0AB}\ProductName
Install Locations	HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-

	18\Products\00005109810090400100000000F01FEC\InstallProperties
Installation Packages	HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\PackageInstallation\Microsoft.Windows.AppRep.ChxApp_cw5n1h2txyewy
Installation Folders	HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Installer\Folders
Templates & Configurations	HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Components\
Setup Duration	HKEY_LOCAL_MACHINE\Software\Microsoft\Office\Office version\Common\Config\{90150000-0011-0000-1000-0000000FF1CE }
Install Count	HKEY_LOCAL_MACHINE\Software\Microsoft\Office\15.0\Common\InstallRoot

On running the setup installer of MS Office Suite, Background Activity Moderator (BAM) entry is added which discloses the execution *Timestamp* of the executable as shown in Figure 13. *Product Name* field shows the name of the office product and *Install Location* provides the path on computer where Office Suite and all office products are installed.

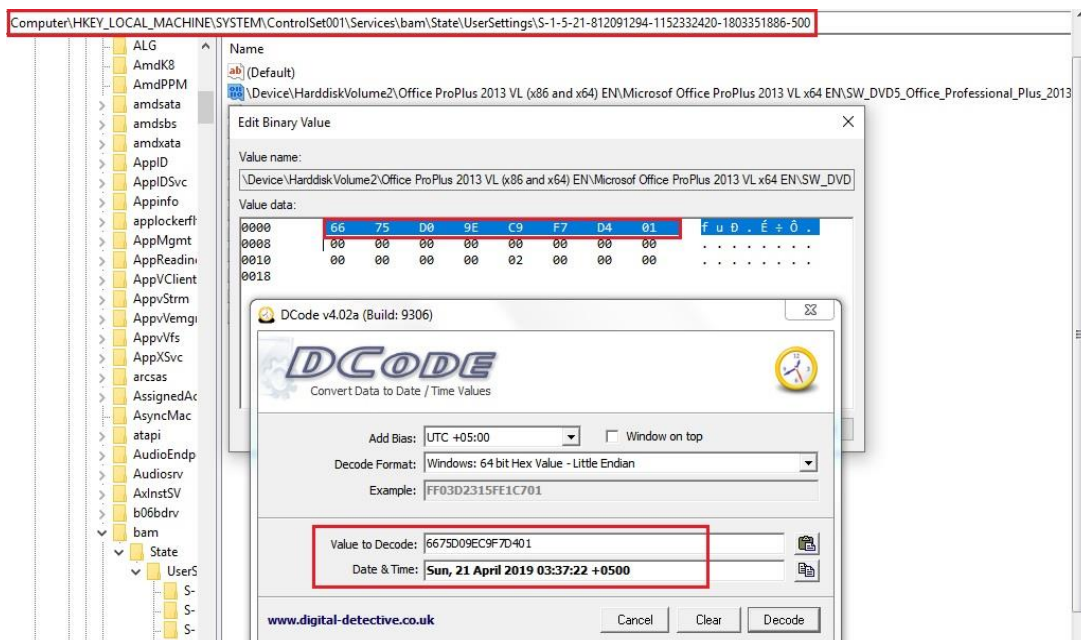


FIGURE 13 INSTALLER TIMESTAMP IN REGISTRY

Installation Packages provides the information about packages used by the windows installer to install Office Suite. *Installation Folders* provides the list of Paths created to store office information on computer. *Templates & Configurations* provides the default templates & configurations name and path in lieu of all office applications. It contains the paths of the templates and configurations related to office applications. *Setup Duration* depicts the total time (in sec) taken for installation of office as shown in Figure 14. While *Install Count* is the counter which shows how many times office suite installer is executed. Whether setup is used for installation or uninstallation of Office, Install Count counter will be incremented by one.

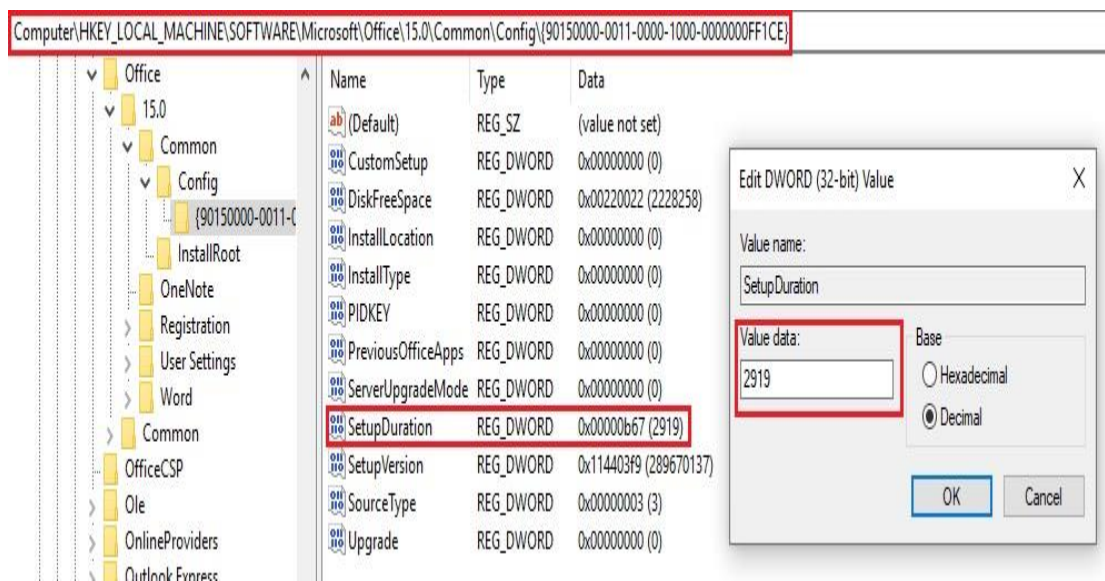


FIGURE 14 SETUP DURATION TIME (SEC)

6.2.2 Opening Microsoft Word, Excel and Power Point

Opening artifacts are recorded on first time running each of the 3 applications and number of registry changes are depicted in Figure 15. Graph shows the higher number of values in *ProcMon* results specifically in case of Word and it is because of a number of Word Fonts are added on first time starting the application. The filtered artifacts extracted on first time launching Microsoft Word, Excel and Power Point are mentioned in Table 12. Most of the artifacts revealed the same location separately for each application with the application name. So, similar locations are mentioned as *Common* rows. As mentioned earlier, Background Activity Moderator (BAM) keeps

track of all the executables accessed on the system that includes information of each user who accessed an executable along with executable *Path and Accessed Time*. If the product is not activated then each time running of an application will give activation alert. So, *Product Activation* value in each application’s folder shows the status of the activation. A *Migration* key is added for each office application which keeps migration data in an event of migrating office to the later version. Similar to the results of *RegFromApp*, *ProcMon* also captures additional changes occurred in case of accessing Word as it adds 260+ *Font* values in the registry. *Font* values are not seen in case of Excel and Power Point.

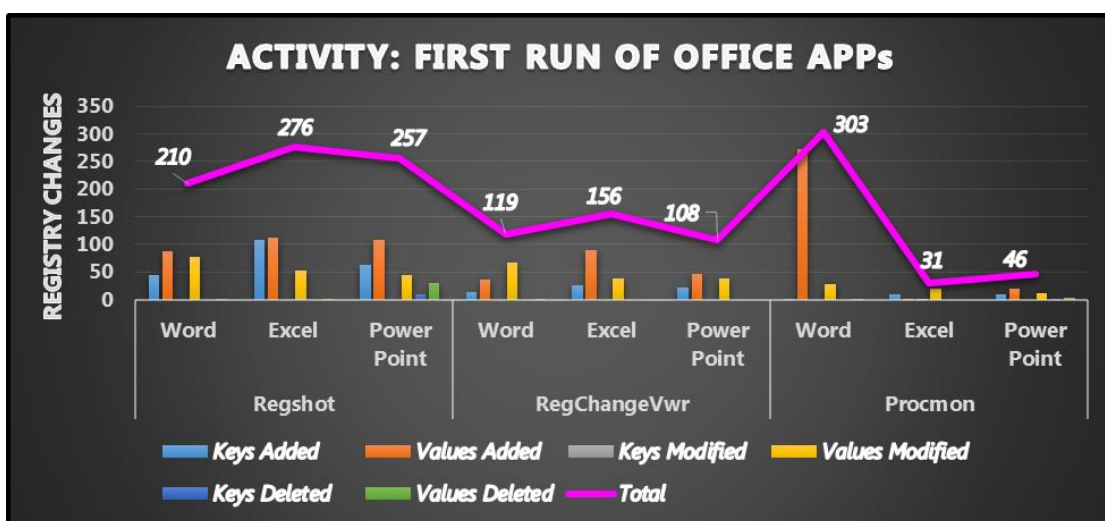


FIGURE 15 FIRST RUN REGISTRY CHANGES

TABLE 12. FIRST TIME APP OPENING ARTIFACTS

App	Artifact	Registry Location
Common	Access Time	HKEY_LOCAL_MACHINE\System\ControlSet001\Services\bam\State\UserSettings\{User SID}
	App Path	
	Username	
	Product Activation	HKEY_CURRENT_USER\Software\Microsoft\Office\15.0\AppName
	Migration	HKEY_CURRENT_USER\Software\Microsoft\Office\15.0\Common\Migration\AppName

Word	Fonts	HKCU\Software\Microsoft\Office\15.0\Common\Math Fonts\
-------------	--------------	--

6.2.3 Closing Microsoft Word, Excel and Power Point

Figure 16 shows a relatively linear graph of registry changes made on closing applications. *RegShot* results were recorded after collecting results of all of the other tools that caused addition of extra values and shows higher numbers.

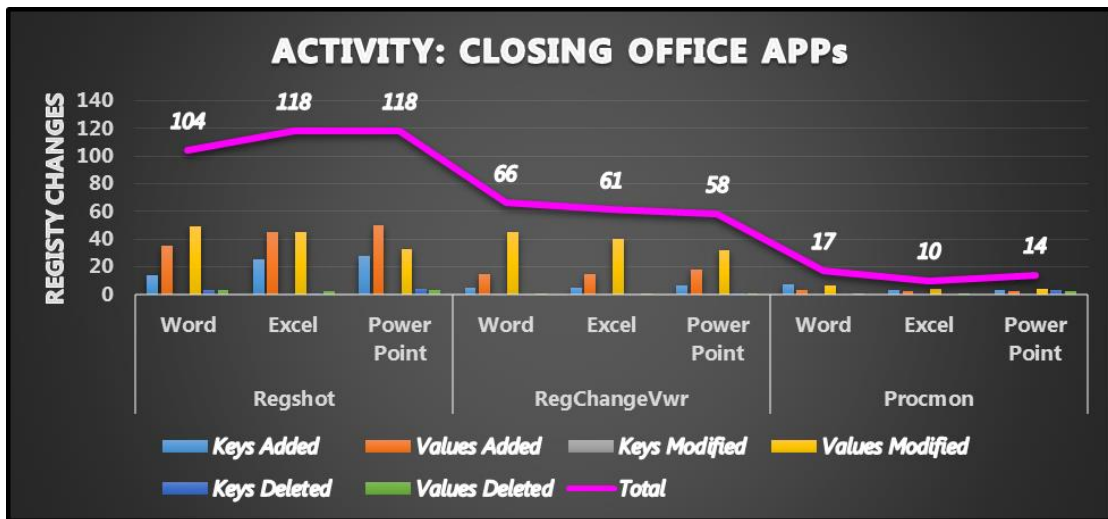


FIGURE 16 CLOSING REGISTRY CHANGES

Upon closing of office application the most important artifact is the closing *Timestamp* of the application, which is filtered out from registry key of BAM state and is shown in Table 13. In case of different values Word, Excel and Power Point store their current screen *Position* in options of their specific folder. Each have different way of storing screen positions i.e. *AppWindowPos* for Word, *Pos* for Excel and (*Left, Right, Top, Bottom*) values for Power Point.

TABLE 13. APP CLOSING ARTIFACTS

App	Artifact	Registry Location
Common	Last Access Time	HKLM\System\CurrentControlSet\Services\bam\State\UserSettings\S-1-5-21-812091294-1152332420-
	App Path	

	Username	1803351886-500\\Device\HarddiskVolume2\Program Files\Microsoft Office\Office15\AppDataName.EXE
Word	Position	HKCU\Software\Microsoft\Office\15.0\Word\Options\AppWindowPos
Excel	Position	HKCU\Software\Microsoft\Office\15.0\Excel\Options\Pos
Power Point	Position	HKCU\Software\Microsoft\Office\15.0\PowerPoint\Options\

6.2.4 Creating File

On creating a new file, most of the changes are produced in categories of *Keys Added*, *Values Added* and *Values Modified* as shown in Figure 17. Obviously, creation of new file will contribute more towards keys and values added or modified as compared to keys and values deleted. Creating a new digital document is always very important in digital investigations because document's name, path and timestamp help in attribution.

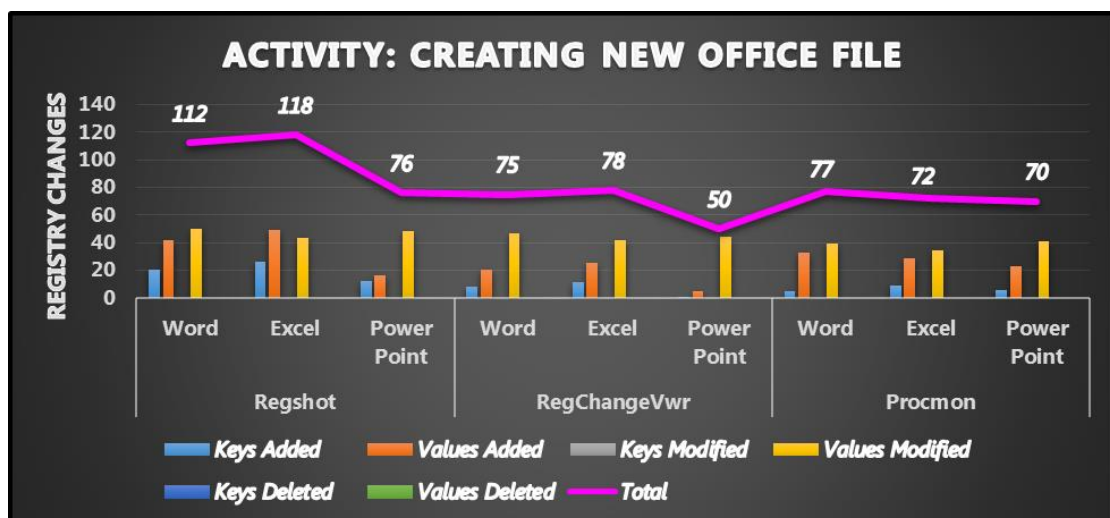


FIGURE 17. CREATING FILE REGISTRY CHANGES

Artifacts in Table 14 are belongs to *NTUser.dat* which makes it easier to relate it to a single user. *File MRU* gives information about the file (Figure 18) and *Place*

MRU gives recent saving locations on windows. Meanwhile whenever a user is working on an opened document, a *Document Recovery* key is created and that document is periodically triggered for auto saving, which makes it protected in case of abrupt shutdown. So, on malfunctioning of Office or Computer, *Resiliency* values will be kept in the registry and will help in auto recovering the document when the next time office runs as shown in Figure 19.

TABLE 14. FILE CREATING ARTIFACTS

App	Artifact	Registry Location
Common	Filename	
	File Path	HKEY_CURRENT_USER\Software\Microsoft\Office\
	Time stamps	15.0\AppName\File MRU
	Recent Places	HKEY_CURRENT_USER\Software\Microsoft\Office\
	Document Recovery	15.0\AppName\Resiliency\DocumentRecovery
	Jump List Data	HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Search\JumplistData



FIGURE 18 FILE MRU DESCRIPTION

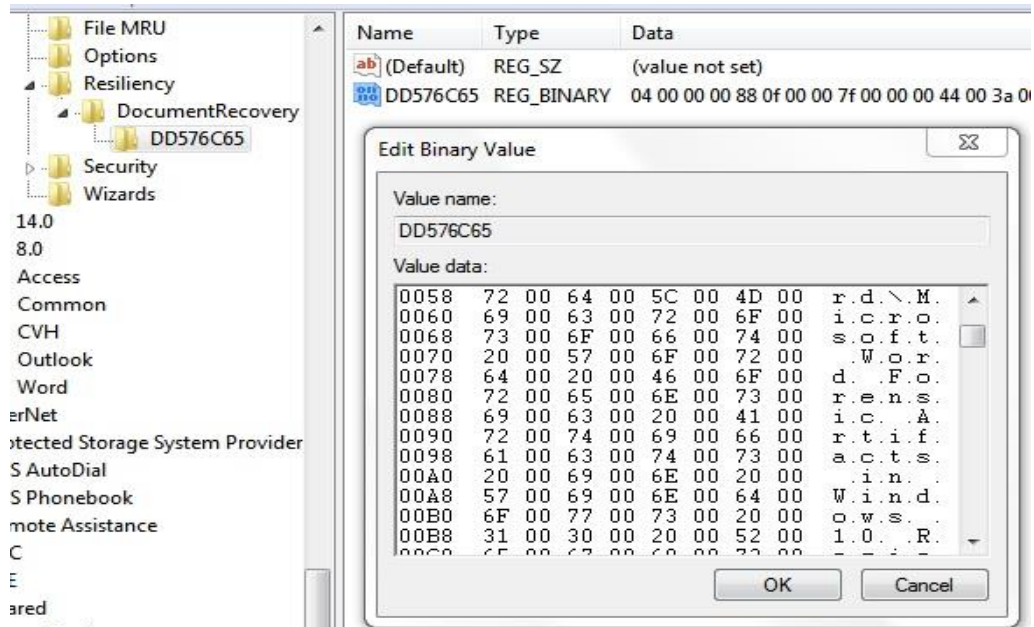


FIGURE 19 DOCUMENT RECOVERY KEY

On the other hand *Document Recovery* key is deleted whenever there is a normal closure of application takes place. *JumplistData* is saved in registry which keeps the record of recent user activities in Windows 10 i.e. File creation in this case.

6.2.5 Accessing File

On accessing the already created file, most of the values are just modified as shown in Figure 21. The filtered results show that there is not much added to the previous values i.e. *File MRU* will keep the recently accessed document at Item 1 location, *Place MRU* timestamp will be modified, *Document Recovery* is again created and *BAM* state will be maintained if the application is not already running. There is only one value of *Reading Location* (which was modified last time the file is closed) will be added in this activity as shown in Table 15. *Reading Location* keeps the last cursor location of the closed document as shown in Figure 20.

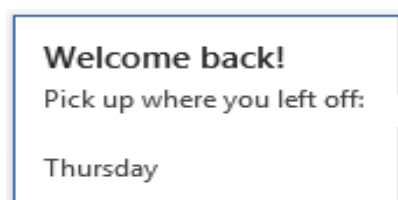


FIGURE 20 READING LOCATION CURSOR

TABLE 15. ACCESSING FILE ARTIFACTS

App	Artifact	Registry Location
Common	Reading Location	HKEY_CURRENT_USER\Software\Microsoft\Office\15.0\AppName\Reading Locations\

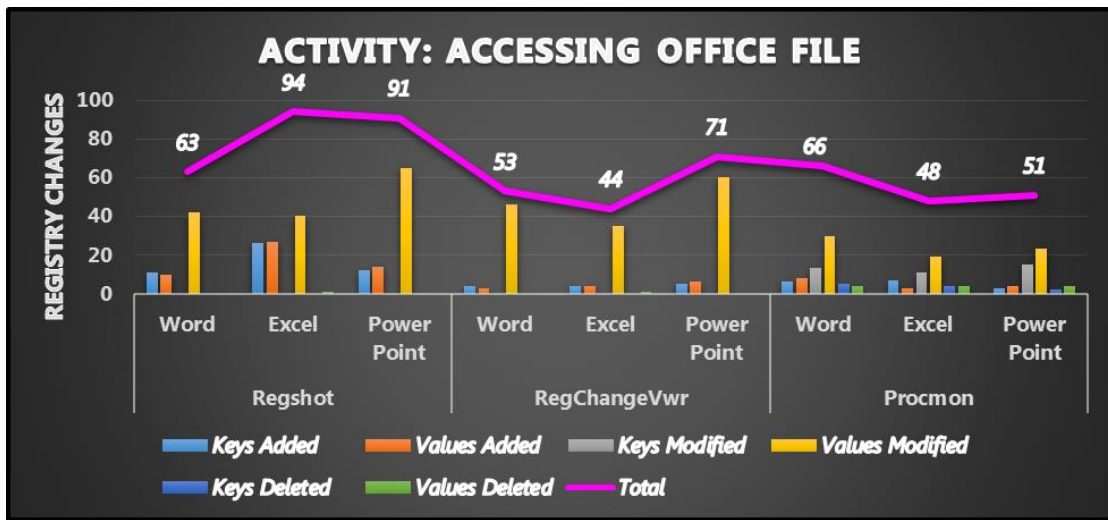


FIGURE 21 ACCESSING FILE REGISTRY CHANGES

6.2.6 Modifying File

As per the name of the activity most of the values are modified and are depicted in relatively linear graph shown in Figure 22. Upon modifying and saving the file same values i.e. *File MRU*, *Place MRU*, *Document Recovery* and *Reading Location* are modified according to the new modifications.

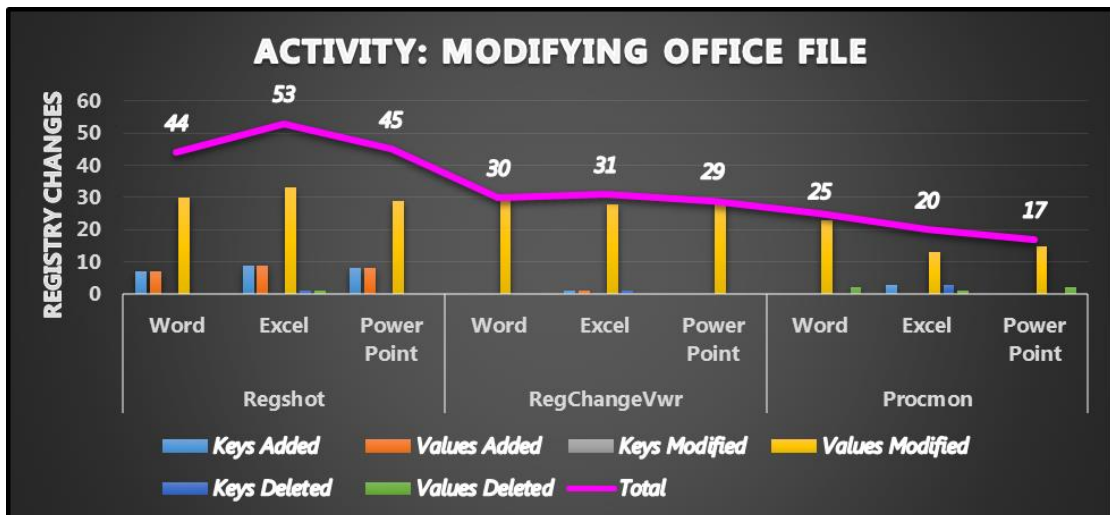


FIGURE 22 MODIFYING FILE REGISTRY CHANGES

6.2.7 Closing File

Similarly on closing the file most of the values are again modified and are shown in Figure 23. Upon closing the file latest cursor location is stored in *Reading Location* value and document is removed from the *Document Recovery* key. Moreover, if the document is the last one such that the application is also closed the *Resiliency* key will be removed and *BAM* state of the application will be maintained too.

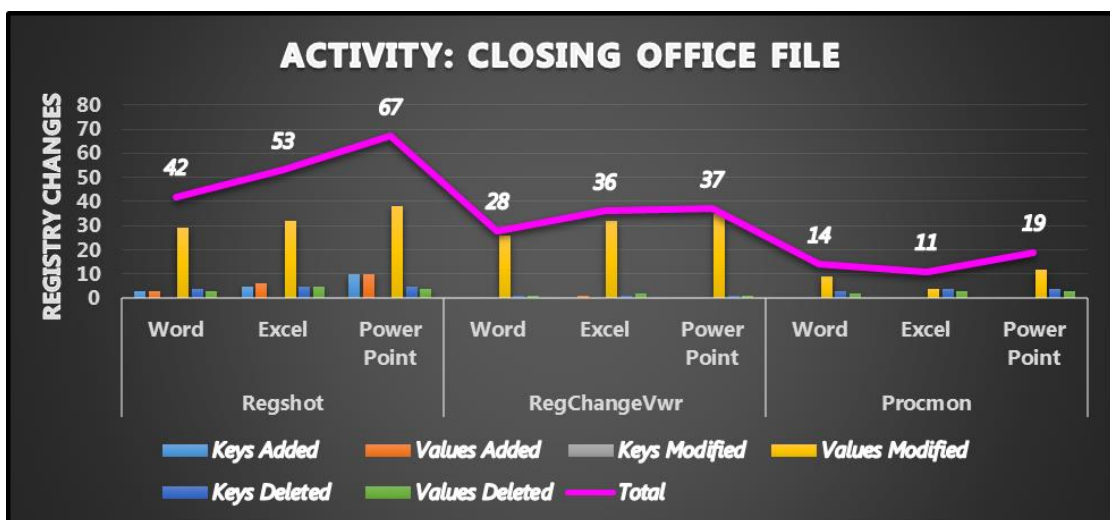


FIGURE 23 CLOSING FILE REGISTRY CHANGES

6.2.8 Microsoft Office Suite Un-installation

The nature of the un-installation activity in normal circumstances will delete many of the registry keys and values that were added upon installation of Microsoft Office. Same can be seen in Figure 24 that the huge values are recorded in the fields of *Keys Deleted* and *Values Deleted*. There are thousands of keys and values related to the installation of Microsoft Office applications are removed but few important one are shown in Table 16. On installation, Office applications are registered in *Registered Applications* registry path, for instance value *PowerPoint.Application.15* is Application registration of Microsoft Power Point and is removed upon un-installation of Microsoft Office Suite. *Product Code* value is placed in Windows registry only if the Office is installed and value is deleted on un-installation of Office Suite. *Installer Folders* location shows a number of folder paths which were added in registry values during installation and are removed upon un-installation.

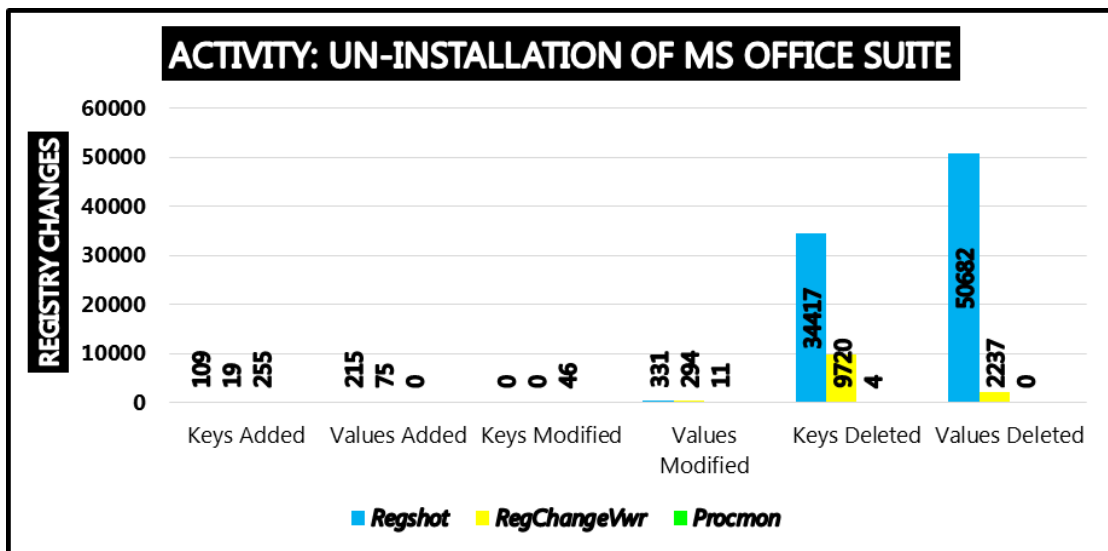


FIGURE 24 OFFICE UN-INSTALLATION REGISTRY CHANGES

TABLE 16. OFFICE UN-INSTALLATION ARTIFACTS

Artifact	Registry Location
Registered Applications	HKEY_LOCAL_MACHINE\Software\RegisteredApplications
Product Code	HKEY_LOCAL_MACHINE\Software\Microsoft\Office\15.0\Registration\{2B88C4F2-EA8F-43CD-805E-4D41346E18A7}

Installer Folders	HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Installer\Folders
--------------------------	--

The abovementioned results depicts that proposed methodology can be applied on the basis of activities performed on windows operating system and then registry results of multiple tools may be filtered out and cross compared to get the validated and forensically sound artifacts.

6.3 How Proposed Methodology is Beneficial?

As already discussed in Chapter 1 that each tool has its own capabilities i.e. Relevant data filtering, Timestamps, paths etc. as well as inabilities i.e. garbage values, irrelevant read operations, insufficient or huge data etc. Therefore, while proposing the methodology for performing research analysis on Windows registry, multiple registry forensic tools are used in a way which abandons their inabilities and considers their capabilities to focus only on valuable artifacts. For example, *RegShot* and *RegChangeVwr* include additional but useful registry keys and values related to *Shellbags* (Stores information about recently used folder locations, positions, icons etc) which are not monitored by *ProcMon* and *RegFromApp*. It is because *ProcMon* and *RegFromApp* connect with the Process ID and only monitor changes produced by the Process under observation. *Shellbags* entries are the result of different process i.e. *explorer.exe*, and are depicted in *RegShot* and *RegChangeVwr* results because they work on before and after activity snapshots comparison. The proposed methodology is focusing solely on extracting the valuable artifacts by filtering out the garbage. Extracted valuable artifacts are produced by taking intersection of results of different tools. During proposed methodology, intersection is obtained in the process of filtration by the method of cross comparison and sanitization. Hence, the methodology will be beneficial for performing research in the field Windows registry forensics and in this way, produced results will be helpful in digital investigations.

LIMITATIONS, CONCLUSION AND FUTURE REQUIREMENTS

7.1 Limitations and Future Work

Due to a huge number of windows registry keys and values changes observed and that too across multiple tools, it was quite difficult and consumed a lot of time in filtration and validation of the results. In order to climb the mountain of huge number of registry changes which are keep on increasing with the evolution of Windows operating system, it is important to develop a tool which include Machine Learning based technique to automatically collect forensically sound artifacts. However, the same methodology can be applied to monitor activities other than windows registry. For this purpose different tools will be used according to the requirement of the activities to be monitored. Though proposed study may be used in future to conduct research works related to Windows Registry Analysis.

7.2 Conclusion

Due to the high amount of evidence storage, Windows Registry is always important from digital investigations point of view and helps in attribution. This paper proposes a methodology for Windows Registry forensic Analysis by describing a way to incorporate multiple registry tools in a way which allows the researcher to filter and cross validate the results. In this way, obtained outcome will produce clean and authentic artifacts which will help digital investigations. The generalized methodology is introduced using registry live monitoring and Registry snapshot tools to collect forensically sound artifacts. A case study is carried out on Microsoft Office Suite to check the effectiveness of the methodology and resulted registry changes are elaborated in the form of graphs. Furthermore, filtered and cross validated artifacts are discussed in details to complete the research work. The research will simplify the Registry Forensic Analysis and will help researchers in collecting trustworthy pieces of evidence from registry.

REFERENCES

- [1] P. Čisar and S. M. Čisar, “General Directions of Development in Digital Forensics,” *Acta Tech. Corviniensis - Bull. Eng.*, vol. 5, no. 2, pp. 87–92, 2012.
- [2] I. O, D. Chris, and D. David, “A New Approach of Digital Forensic Model for Digital Forensic Investigation,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 2, no. 12, pp. 175–178, 2011.
- [3] Netmarketshare, “Operating Systems Market Share.” [Online]. Available: <https://netmarketshare.com/operating-system-market-share.aspx>.
- [4] H. Carvey, “The Windows Registry as a forensic resource,” *Digit. Investig.*, vol. 2, no. 3, pp. 201–205, 2005.
- [5] S. B. Lee, J. Bang, K. S. Lim, J. Kim, and S. Lee, “A stepwise methodology for tracing computer usage,” *NCM 2009 - 5th Int. Jt. Conf. INC, IMS, IDC*, pp. 1852–1857, 2009.
- [6] K. S. Lim, S. B. Lee, and S. Lee, “Applying a stepwise forensic approach to incident response and computer usage analysis,” *Proc. 2009 2nd Int. Conf. Comput. Sci. Its Appl. CSA 2009*, no. December 2009, 2009.
- [7] T. D. Morgan, “Recovering deleted data from the Windows registry,” *DFRWS 2008 Annu. Conf.*, vol. 5, pp. 33–41, 2008.
- [8] A. Singh, H. S. Venter, and A. R. Ikuesan, “Windows registry harnesser for incident response and digital forensic analysis,” *Aust. J. Forensic Sci.*, vol. 00, no. 00, pp. 1–17, 2018.
- [9] A. Amin, F. Shabbir, S. Saleem, M. Waheed, and Z. Khan, “Microsoft Word Forensic Artifacts in Windows 10 Registry,” in *2019 International Conference on Applied and Engineering Mathematics, ICAEM 2019 - Proceedings*, 2019.
- [10] L. Bruno, “濟無No Title No Title,” *Journal of Chemical Information and Modeling*, 2019. [Online]. Available: <https://support.microsoft.com/en-us/help/256986/windows-registry-information-for-advanced-users>.
- [11] Microsoft, “No Title.” [Online]. Available: <https://docs.microsoft.com/en>

- us/windows/win32/sysinfo/registry.
- [12] ComputerHope, “No Title.” [Online]. Available: <https://www.computerhope.com/jargon/r/registry.htm>.
- [13] D. Bem, F. Feld, E. Huebner, and O. Bem, “Journal of Information Science and Technology www,” 2008.
- [14] P. Čisar and S. M. Čisar, “Methodological frameworks of digital forensics,” *SISY 2011 - 9th Int. Symp. Intell. Syst. Informatics, Proc.*, pp. 343–347, 2011.
- [15] S. Al-Fedaghi and B. Al-Babtain, “Modeling the forensics process,” *Int. J. Secur. its Appl.*, vol. 6, no. 4, pp. 97–108, 2012.
- [16] N. Z. Khidzir and M. Ahmed, “Towards Fact-Based Digital Forensic Evidence Collection Methodology,” *SSRN Electron. J.*, 2018.
- [17] R. M. Saidi, S. A. Ahmad, N. M. Noor, and R. Yunos, “Windows registry analysis for forensic investigation,” in *2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering, TAECE 2013*, 2013.
- [18] S. Raghavan and S. V. Raghavan, “A study of forensic & analysis tools,” in *Int. Workshop Syst. Approaches Digit. Forensics Eng., SADFE*, 2014.
- [19] M. N. Faiz and W. A. Prabowo, “Comparison of Acquisition Software for Digital Forensics Purposes,” *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, no. 1, p. 37, 2018.
- [20] A. Arshad, W. Iqbal, and H. Abbas, “USB Storage Device Forensics for Windows 10,” *J. Forensic Sci.*, vol. 63, no. 3, pp. 856–867, 2018.
- [21] H. Binjuraid and M. Mat Din, “Case Based Interpretation of Windows 10 Registry Forensics,” *Int. J. Innov. Comput.*, vol. 8, no. 1, pp. 43–47, 2018.
- [22] E. Wahyudi, I. Riadi, and Y. Prayudi, “Virtual Machine Forensic Analysis And Recovery Method For Recovery And Analysis Digital Evidence,” *International J. Comput. Sci. Inf. Secur.*, vol. 16, no. 2, pp. 1–7, 2018.
- [23] Microsoft, “Process Monitor.” [Online]. Available: <https://docs.microsoft.com/en-us/sysinternals/downloads/procmon>.
- [24] Nirsoft, “RegApp.” [Online]. Available:

- https://www.nirsoft.net/utils/reg_file_from_application.html.
- [25] SourceForge, “RegShot.” [Online]. Available: <https://sourceforge.net/projects/regshot/>.
- [26] Nirsoft, “RegChangeView.” [Online]. Available: https://www.nirsoft.net/utils/registry_changes_view.html.
- [27] E. Casey, *Chapter1 of Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet*, Third Edit. Elsevier, 2011.