2013

# Journey through the impact of the recovery artifacts in Windows 8

WENDELL Kenneth JOHNSON
*Iowa State University*

# Journey through the impact of the recovery artifacts in Windows 8

by

**Wendell Kenneth Johnson**

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Co-majors: Computer Engineering; Information Assurance

Program of Study Committee:
Yong Guan, Major Professor
Doug Jacobson
Jennifer L. Davidson

Iowa State University
Ames, Iowa
2013

## DEDICATION

This Thesis is dedicated to my family Jessica, Savannah and Brady. Without your unrelenting support and sacrifices I would not have been able to follow my educational and career dreams.

To Lee Adams, while you will never see the finished work, your guiding light and compassion shown to me helped create the person I am today. My drive to succeed and to share my success comes from watching you give so much of your compassion to others.

# TABLE OF CONTENTS

LIST OF FIGURES

## LIST OF TABLES

# NOMENCLATURE

| | |
|---|---|
| HIVE | File structure that registry values are stored in |
| ROOT KEY | Top level key of the registry database, starts with HKEY |
| SUBKEY | Registry directory structure |
| DWORD | Type of registry key, 32-bit unsigned intiger |
| HKLM | HKEY_LOCAL_MACHINE |
| HKU | HKEY_USER, unique user |
| HKCU | HKEY_CURRENT_USER, current signed in user |
| VSS | Volume Shadow Copy Services |

## ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. Yong Guan, and my committee members, Dr. Doug Jacobson, and Dr. Jennifer Davidson, for their guidance and support throughout the course of this research.

In addition, I would also like to thank my friends, colleagues, the department faculty and staff for making my time at Iowa State University a wonderful experience. I want to also offer my appreciation to those who were willing to provide feedback on my research, without whom, this thesis would not have been possible.

In addition I would also like to thank my family for their encouragement, to my wife, Jessica, and children, Savannah and Brady, their hours of patience, respect and love. Thank you for believing in me and pushing me to succeed.

Finally, thanks to Leocadia Adams for your encouragement and always seeing the best in me. The compassion and life lessons that you taught me will always be appreciated.

## ABSTRACT

One of the most difficult processes of digital forensics is understanding how new technology interacts with current technology and how digital forensic analysts can utilize current Digital Forensics technologies and processes to recover and find information hidden. Microsoft has released their new Operating System Windows 8, with this new release Microsoft has added some features to the Operating System that will present some interesting complications to digital forensics.

Since the initial release of the Windows 8 Release Candidates there have been some research released that focus primarily on the new user created artifacts and a few artifacts that have been added by the operating system that might contain valuable information. In this paper I will look at the new recovery options that have been introduced in Windows 8, and the impact that have on the artifacts.

The first thing that I plan to look at is the artifacts discovered by the research of Amanda Thomson. Once I have analyzed these artifacts and then verify the locations on the disk I will create a baseline dataset to compare the impact of the recovery options on these artifacts. I will also use artifacts of new features that I have researched for this baseline.

The second thing that I will look at is how the various recovery options impact the artifacts that are found on the operating system. This will be done by

installing Windows 8 in a Virtual Machine environment and taking snapshots of a base image and then utilizing the various recovery methods.

The final thing that I will include in this paper is a detailed walk through on where the artifacts will reside on the machine after a recover option has been completed. I will examine the locations on a live machine as well as on a forensic copy. I will show what artifacts are easily recoverable, what artifacts need a little time to recovery and what artifacts that will not be recoverable.

# CHAPTER I

## INTRODUCTION: WINDOWS 8 FORENSICS

One of the most difficult processes of digital forensics is understanding how new technology interacts with current technology and how digital forensic analysts can utilize current Digital Forensics technologies and processes to recover and find information hidden. Microsoft has released their new Operating System Windows 8, with this new release Microsoft has added some features to the Operating System that will present some interesting complications to digital forensics.

Over the last year there has been plenty of research released that focus on the new user created artifacts and a few artifacts that have been added by the operating system that might contain valuable information. In this paper I will look at the new recovery options that have been introduced in Windows 8, and the impact that have on the artifacts.

The first thing that I plan to look at is the artifacts discovered by the research of Amanda Thomson. Once I have analyzed these artifacts and then verify the locations on the disk I will create a baseline dataset to compare the impact of the recovery options on these artifacts. I will also use artifacts of new features that I have researched for this baseline.

The second thing that I will look at is how the various recovery options impact the artifacts that are found on the operating system. This will be done by installing Windows 8 in a Virtual Machine environment and taking snapshots of a base image and then utilizing the various recovery methods. Once the recovery method has been successful I will take the Virtual Machine and mount it into FTK and Encase for analysis.

The final thing that I will include in this paper is a detailed walk through on where the artifacts will reside on the machine after a recover option has been completed. I will examine the locations on a live machine as well as on a forensic copy. I will show what artifacts are easily recoverable, what artifacts need a little time to recovery and what artifacts that will not be recoverable.

# CHAPTER 2

# GATHERING OF INFORMATION

## Windows 8 Forensic Overview

Amanda Thomson released documentation on various artifacts that the new Metro interface of the Windows 8 operating system creates (Thomson, 2012). I will look into the communication apps, internet explorer 10, File History, registry values and user created documents. This will create a baseline of data to search for and extract from.

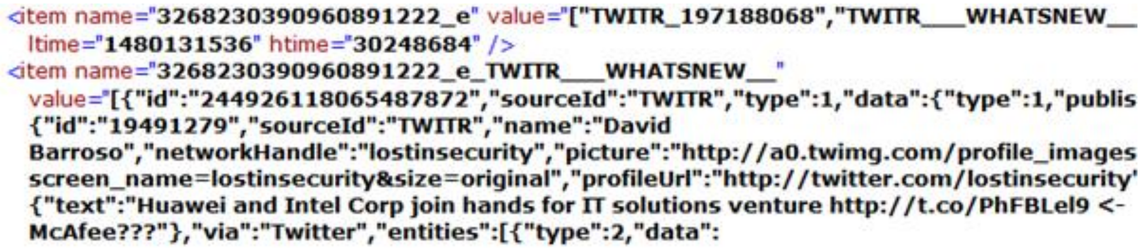
### Communication Application Artifacts

Some of the more useful artifacts that are included with Windows 8 as the Metro App known as the Communication App. Amanda Thomson went into great details in her research on the artifacts that can be recovered. She showed that analysts are able to extract artifacts that will show who and how a user interacts with various online organizations and people. (Thomson, 2012)

From within the Communication App there are few locations that will provide artifacts for analysis; Thomson touched on the Cache and the Mail locations. I will discuss the following artifacts that I discovered in the application; AddressBook and Me directories. While there may be more of value, these two locations open up more insight on the users contacts and the accounts being used by the user.

From the cache files analysts are able to extract the user online contacts details. These details include email, twitter handle, profile pictures, and pictures that were shared by the users contact. From the cookies files analysts can extract conversation, email, email attachments, twitter communications and other communication transactions between users.

From the cookies files analysts are able to extract user messages that have appeared in the communication app. Thomson's research showed how email messages appeared in the cookies directory, this example shows how a Twitter stream would appear in the cookies directory. In this example we can see the username that posted to tweet, the content of the tweet and the associated url that was included in the tweet.



```
<item name="3268230390960891222_e" value="["TWITR_197188068","TWITR___WHATSNEW__
  ltime="1480131536" htime="30248684" />
<item name="3268230390960891222_e_TWITR___WHATSNEW__"
  value="[{"id":"244926118065487872","sourceId":"TWITR","type":1,"data":{"type":1,"publis
  {"id":"19491279","sourceId":"TWITR","name":"David
  Barroso","networkHandle":"lostinsecurity","picture":"http://a0.twimg.com/profile_images
  screen_name=lostinsecurity&size=original","profileUrl":"http://twitter.com/lostinsecurity'
  {"text":"Huawei and Intel Corp join hands for IT solutions venture http://t.co/PhFBLel9 <-
  McAfee???"},"via":"Twitter","entities":[{"type":2,"data":
```

Figure 1

From the mail files analysts can recover information about the emails the sender has sent, received, stored or even ads that have populated the inbox of the live account tied to the communication app. According to Thomson’s research the file path to the Mail directory is the users windows live account. This is no longer the case as it is now a random string. I have verified that by having the same windows live account across multiple machines and this 16 alpha numeric character directory does not share the same name across the machines.

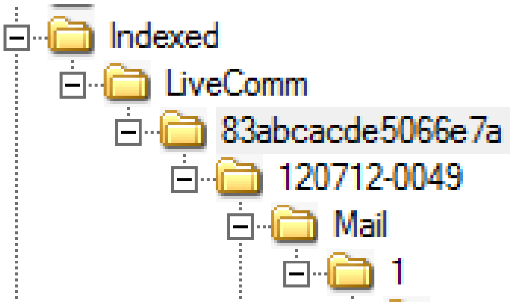
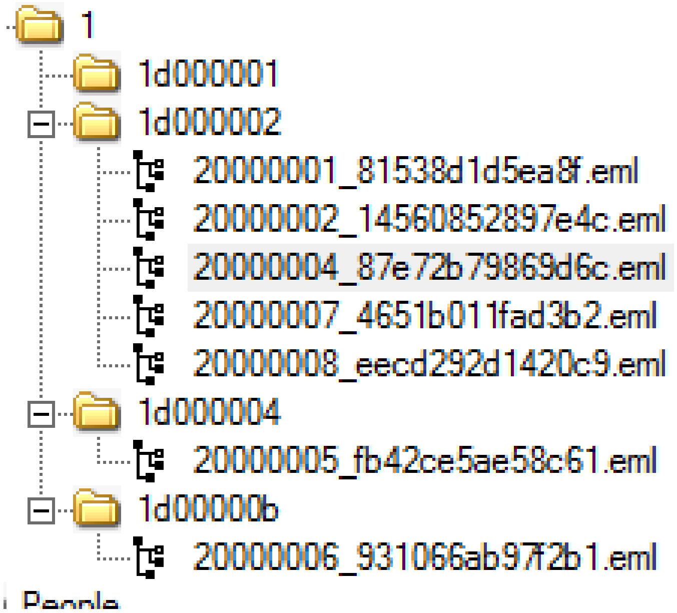


Figure 2

Within the Mail directory are subdirectories that hold various files. The subdirectories on my machines met the following naming standards 1d00000# while in Thomson’s research these values were 1200000#, it appears that this naming standard follows the similar hex pattern, although at this time it is unknown what the meaning of the pattern is. The files in the subdirectories in my testing followed the naming standard of:

2000000#_##############.eml.OECustomProperty

(14 alpha numeric characters)



```
1
  1d000001
  1d000002
    20000001_81538d1d5ea8f.eml
    20000002_14560852897e4c.eml
    20000004_87e72b79869d6c.eml
    20000007_4651b011fad3b2.eml
    20000008_eecd292d1420c9.eml
  1d000004
    20000005_fb42ce5ae58c61.eml
  1d00000b
    20000006_931066ab97f2b1.eml
People
```

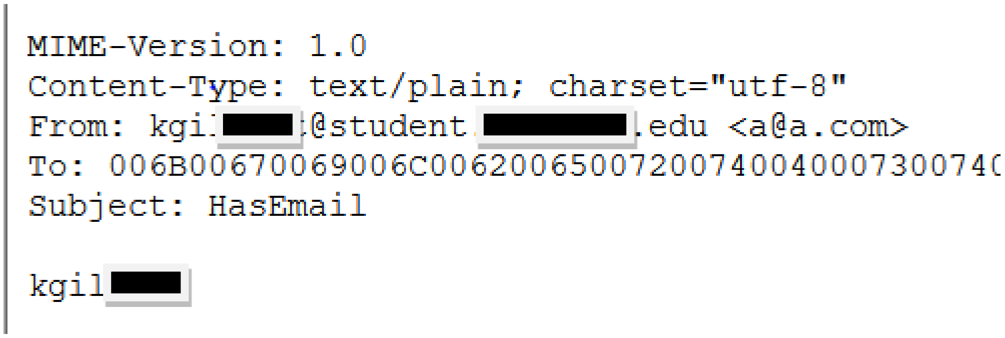
Figure 3

The different subdirectories under the mail directory appear to be different directories in the Windows Live email system. In my case directory 1d000002 appears to be my inbox, the directory 1d000004 appears to have been my sent folder, and the 1d00000b appears to be my draft folder.

From the AddressBook file analysts are able to gather username of the contacts.

All entries will contain a From field which will list the Contact name, at the end of the entry there will also be a list of all alias's tied to that account. These alias can be email usernames, full names, first name, last name or even another screen name.

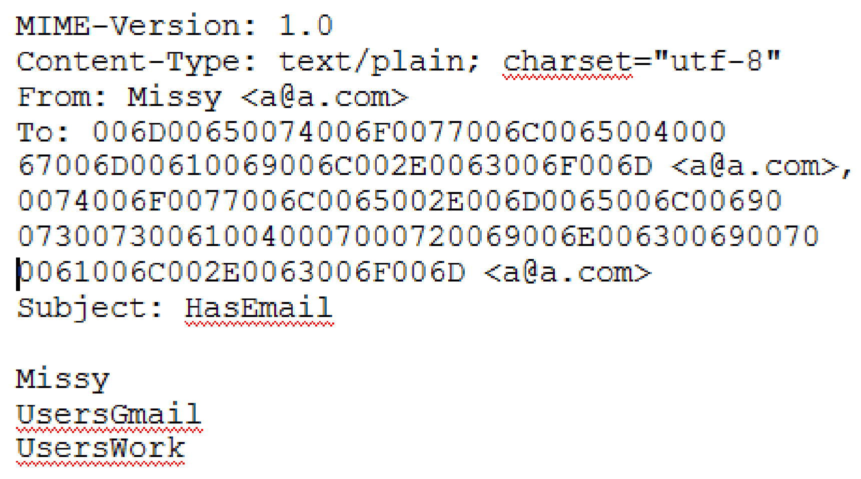
For a contact with an associated email the entry will contain a Subject line that will have te value HasEmail if this account is tied to an email address. In figure X we can see a user with only one email account, and in figure two we can see a user with multiple email accounts. In the To; field there will be a hex character string that is the associated email addresses for this contact.



```
MIME-Version: 1.0
Content-Type: text/plain; charset="utf-8"
From: kgi[redacted]@student[redacted].edu <a@a.com>
To: 006B00670069006C00620065007200740040007300740
Subject: HasEmail

kgil[redacted]
```

Figure 4

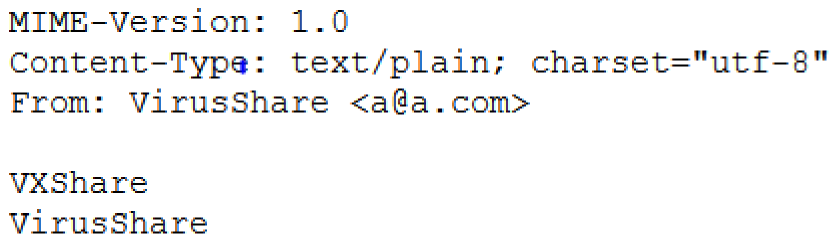


```
MIME-Version: 1.0
Content-Type: text/plain; charset="utf-8"
From: Missy <a@a.com>
To: 006D00650074006F0077006C0065004000
67006D00610069006C002E0063006F006D <a@a.com>,
0074006F0077006C0065002E006D0065006C00690
073007300610040007000720069006E006300690070
0061006C002E0063006F006D <a@a.com>
Subject: HasEmail

Missy
UsersGmail
UsersWork
```

Figure 5

For a Twitter contact the two fields at the bottom of the artifact will list the screen name first and the first and last name that was entered into the product. Since this is a twitter account and no email is associated with it there is no subject that shows the HasEmail value.

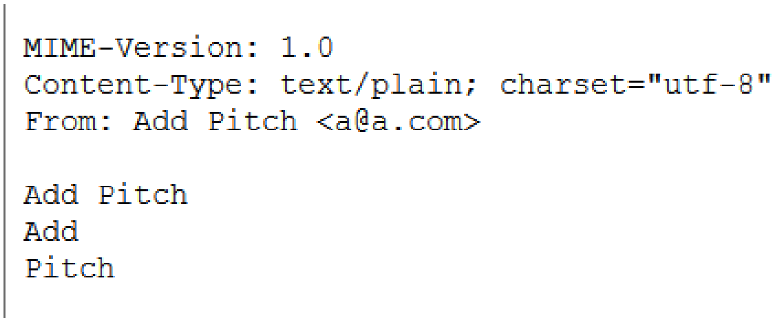


```
MIME-Version: 1.0
Content-Type: text/plain; charset="utf-8"
From: VirusShare <a@a.com>

VXShare
VirusShare
```

Figure 6

For a Facebook contact the artifact will contain the full name in the from field, and at the end of the artifact the first line will again be the full name, the second line will be the first name associated with the account and the last line will be the last name associated with the account.

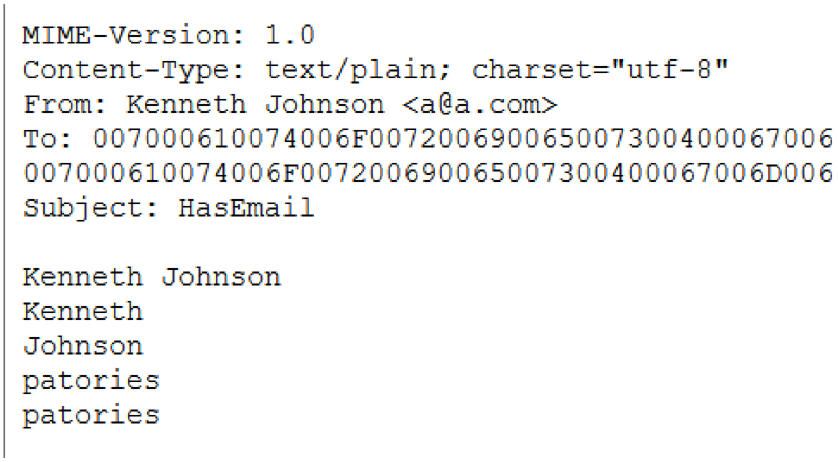


```
MIME-Version: 1.0
Content-Type: text/plain; charset="utf-8"
From: Add Pitch <a@a.com>

Add Pitch
Add
Pitch
```

**Figure 7**

The Me folder will contain an artifact entry that will contain all the accounts that the user has connected to the communication app. This artifact follows the same structure as previous examples; containing the hex string of the email address, user name and the associated full name.

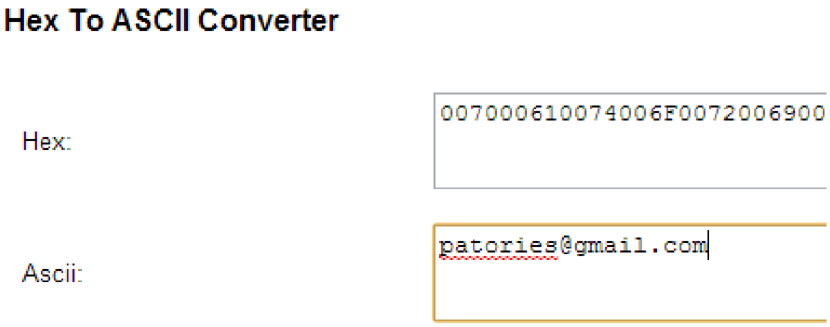


```
MIME-Version: 1.0
Content-Type: text/plain; charset="utf-8"
From: Kenneth Johnson <a@a.com>
To: 007000610074006F007200690065007300400067006
007000610074006F007200690065007300400067006D006
Subject: HasEmail

Kenneth Johnson
Kenneth
Johnson
patories
patories
```

Figure 8

Figure 9 shows the hex being converted into a readable Ascii string showing the email



Hex To ASCII Converter

Hex: 007000610074006F0072006900

Ascii: patories@gmail.com

Figure 9

In the table below I have updated the artifact list to include other artifacts that should be of interest to an analyst. I have included the location of the Cache and Mail artifacts as well as introducing the Adressbook and the ME locations.

Table 1

| Artifact Type | Artifact Location | Purpose |
|---|---|---|
| Cache | %Root%\Users\%User%\AppData\Local\Packages\microsoft.windowscommunicatisapps_8wekyb3d8bbwe\AC\INetCache | Contains contacts email, screen name, or images the user has viewed. |
| Cookies | %Root%\Users\%User%\AppData\Local\Packages\microsoft.windowscommunicatisapps_8wekyb3d8bbwe\AC\INetCookies | Copy of messages that have shown up in the Communication App. |
| Mail | %Root%\Users\%User%\AppData\Local\Packages\microsoft.windowscommunicatisapps_8wekyb3d8bbwe\LocalState\Indexed\LiveComm\%randomString%\%randomString\Mail | Copy of users emails, these will contain sender, recipient, subject, body and attachments. |
| Address Book | %Root%\Users\%User%\AppData\Local\Packages\microsoft.windowscommunicatisapps_8wekyb3d8bbwe\LocalState\Indexed\LiveComm\%randomString%\%randomString\People\Address | Contains username and screen name of contacts. If account has email address then email address is also stored in hex value. |

| | | |
|---|---|---|
| ME | %Root%\Users\%User%\AppData\Local\Packages\microsoft.windowscommunicatisapps_8wekyb3d8bbwe\LocalState\Indexed\LiveComm\%randomString%\%randomString\People\Me | Contains username and screen name of users accounts and all associated email addresses is also stored in hex value. |

Volume Shadow Copy Services

Microsoft introduced the Volume Shadow Copy Service (VSS) as a framework to allow for volume backups while the applications on the system could continue to write to the volumes. (Microsoft)

VSS was initially implemented in Windows XP, and since that time a lot of research has been released that deals with the artifacts that are created through the utilization of VSS. The primary use of VSS on previous versions of Windows was to create System Restore Points and Previous Versioning of documents and files. Within these shadow copies analysts can find data that has been removed from the system or that data that might have been modified. (Larson)

With the release of Microsoft Windows 8, the primary use of VSS has changed. While it is still used for Restore Points it has now also being utilized for

FileHistory Services which replaces the previous versioning on documents and files. As of initial research it appears that Volume Shadow Copies can still be accessed with current technology.

As part of my research I will reevaluate the artifacts created and their behavior to see if anything has changed. I will also look at the integration of the Volume Shadow Copy services with File History Services and what artifacts will be useful for analysis.

FileHistory Services

Within the Microsoft Windows 8 Operating System, they have introduced file history backup, which changes the way backups were previously used. In previous versions windows could only maintain and restore backups using the default system. With Windows 8, Microsoft has implemented a solution that is more robust and allows backups to be stored both on removable media and remote network shares. By default File History will back up the following folders: Music, Documents, Videos, Pictures, Desktop, Contacts and Favorites. (Serban)

There are a few artifacts that are established when the File History is turned on these inclue the file history folder and registry values.

The file history folder can be found in the following path: ***C:\Users\<USERID>\AppData\Local\Microsoft\Windows\Filehistory***

This directory is also written to the backup location. Within in that directory there are two folders named Data, and Configuration. The data folder contains the files and folders that are tagged for backup using the file history. The configuration folder contains files that are both EDB and XML files. File names for the EDB follow the naming conventions of Catalog#.edb, and file names for the XML files are Config#.

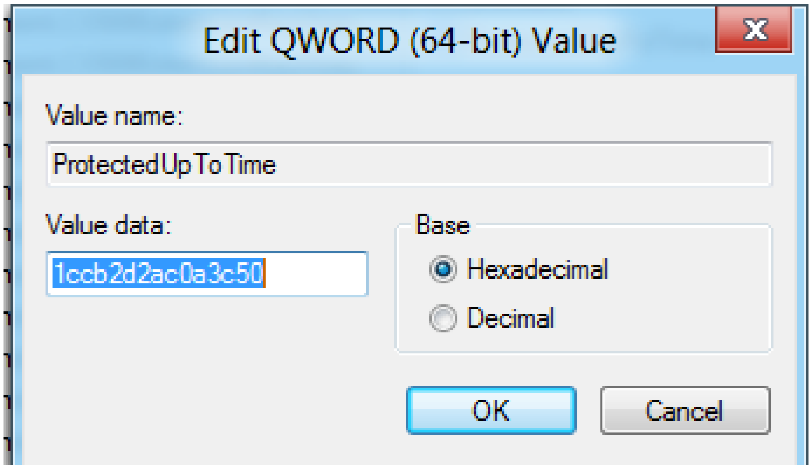
Within the config files there is various bits of data that will be of interest to the digital forensic analysts. These data values include username, computer name and the File History options configuration. These values include:

**Table 2**

| File History Config Values | **What does it mean** |
|---|---|
| UserName | **The user account that this is configed for.** |
| Friendly Name | **First/Lastname tied to account. Inherited from Windows LiveID if configured** |
| PC Name | **Name of the computer** |
| UserType | **What type of user this is** |

| | |
|---|---|
| Library | **Directories that are being backed up** |
| UserFolder | **User Specific Directories being backed up. Sub Value of Library** |
| LocalCatalogPath | **Filepath to the Catalog.edb and the config files** |
| Retention Policy | **How long data is retained** |
| DPFrequency | **How often is the data backed up** |
| Target | **The backup location** |
| TargetName | **Name of the Location used for backup** |
| TargetDriveType | **The options Local, Remote, and Removable are for the drive type that the backup is sent to.** |

If the File History option has been turned on, there will be registry keys created in the HKU keys of the users that have this option turned on. This key can be found in the Software\Microsoft\Windows\CurrentVersion\FileHistory. Within this directory there is a key names ProtectedUpToTime which is a 64 Bit Hex Value – Big Endian, which can be deciphered by utilizing the DCode application. In this case my value is:



Edit QWORD (64-bit) Value

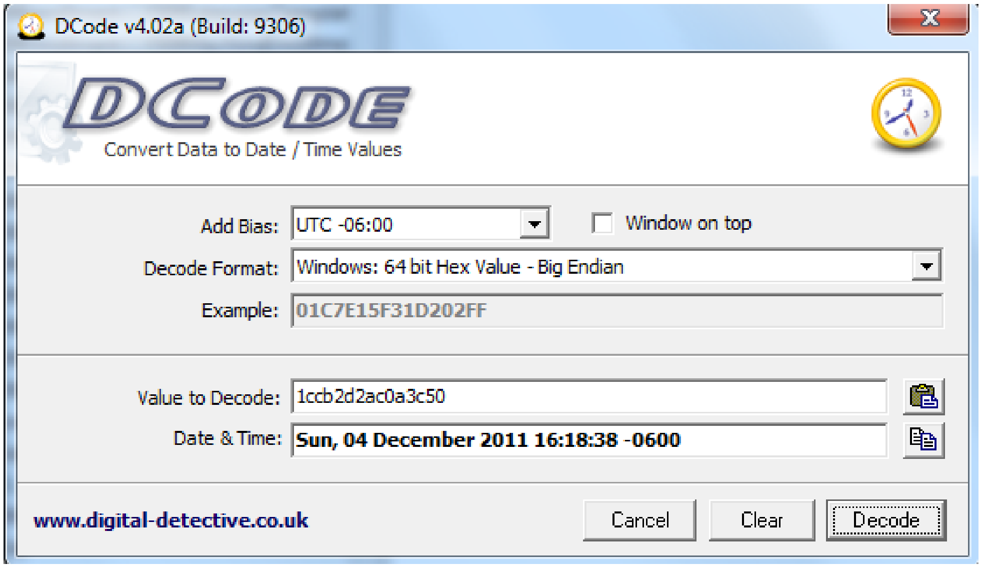
Value name: ProtectedUpToTime

Value data: 1ccb2d2ac0a3c50

Base: ● Hexadecimal ○ Decimal

Figure 10

The ProtectedUpToTime value after it has been submitted and processed through the DCode application.



DCode v4.02a (Build: 9306)

Add Bias: UTC -06:00

☐ Window on top

Decode Format: Windows: 64 bit Hex Value - Big Endian

Example: 01C7E15F31D202FF

Value to Decode: 1ccb2d2ac0a3c50

Date & Time: Sun, 04 December 2011 16:18:38 -0600

www.digital-detective.co.uk

Figure 11

This value represents the last time the system pushed an update to the file history system.

Another area in the registry that may containt keys of importance is HKLM\System\Controlset001\Services\fhsvc. Within this Key, there is a paramater key that shows the location of the configurations values.

Another area to look at in gathering File History information is within the System Events. The following Event Sources provide us with information related to the File History: FileHistory-Catalog, FileHistory-ConfigManager, FileHistory-Core, FileHistory-Engine, FileHistory-EventListener, and FileHistory-Service. As of this research the event logs being parsed are the onces that show errors, and one that fires off with each successful backup, but claims something is missing and can't be parsed. Until this operating system is further along, this issue might remain.

Digital forensic analysts can also utilize the Jump Lists for the File History to gather more information on it. I was able to pull from the file history jump list, the various drives I used for my back up locations. This will be beneficial if the user modifies their backup location.

If a Windows 8 machine has the File History Service turned on, it will persist over a system Refresh. These files can be found in the original directory.

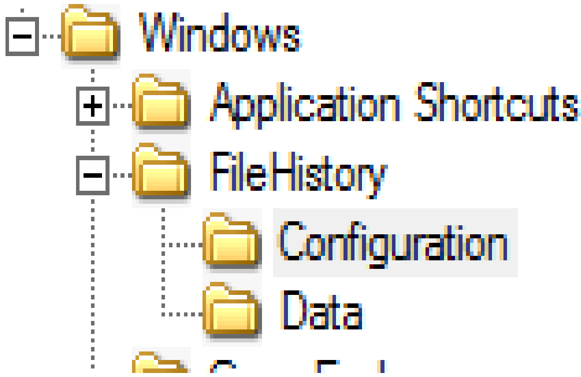


Figure 12

## Jump Lists

The Jump List is a feature that was introduced in Windows 7 that allows for quick access to recently used files, or files that a user could attach to the list. There are two types of files that are created when applications perform certain actions these files are: (4n6k, 2011)

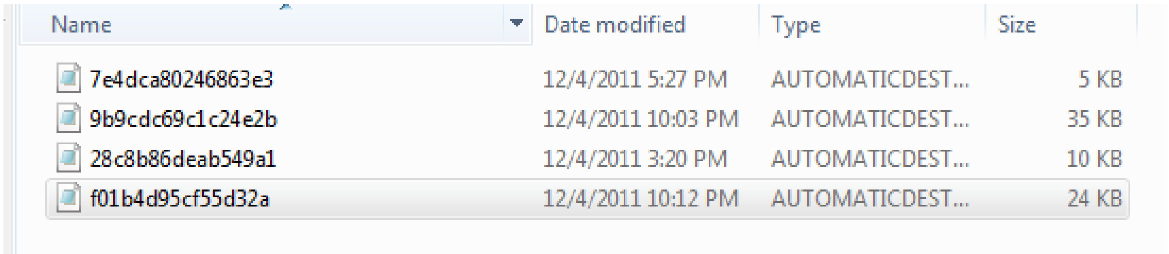
- automaticDestinations-ms files

***C:\Users\<userid>\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations***

- customDestinations-ms files

***C:\Users\<userid>\AppData\Roaming\Microsoft\Windows\Recent\CustomDestinations***

When first analyzing jump lists it is important to know that the jumplists found in the User Appdata directory path do not share a user friendly name. As shown in figure 4, the jumlist name is an alpha numeric file name. This makes it difficult for the analyst to just pick out the jumplist in question, they will need to know what each jumplist ApplicationID corresponds to on the computer. The four files in Figure 4, represent Media Player, Notepad, File History and Windows Explorer applications.



| Name | Date modified | Type | Size |
|---|---|---|---|
| 7e4dca80246863e3 | 12/4/2011 5:27 PM | AUTOMATICDEST... | 5 KB |
| 9b9cdc69c1c24e2b | 12/4/2011 10:03 PM | AUTOMATICDEST... | 35 KB |
| 28c8b86deab549a1 | 12/4/2011 3:20 PM | AUTOMATICDEST... | 10 KB |
| f01b4d95cf55d32a | 12/4/2011 10:12 PM | AUTOMATICDEST... | 24 KB |

**Figure 13**

When an analyst looks at the jump lists from within the user interface they can see the recent and most accessed files associated with that jump list. When I looked at the jump list tied to my Windows Explorer application I was able to see the locations on my computer that I frequently accessed. As shown in figure 5, my frequent access locations included Desktop, Documents, multiple Configuration directories and a few other locations on my computer. The only reason why an item would appear on the jumplist is if the user had interacted with the application at some point.

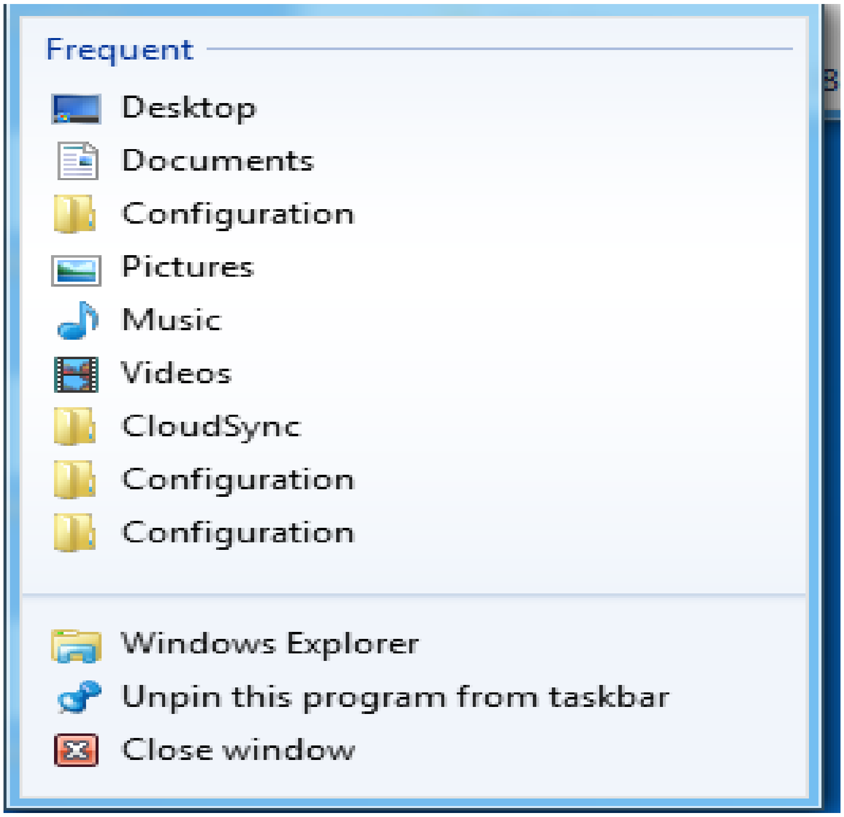


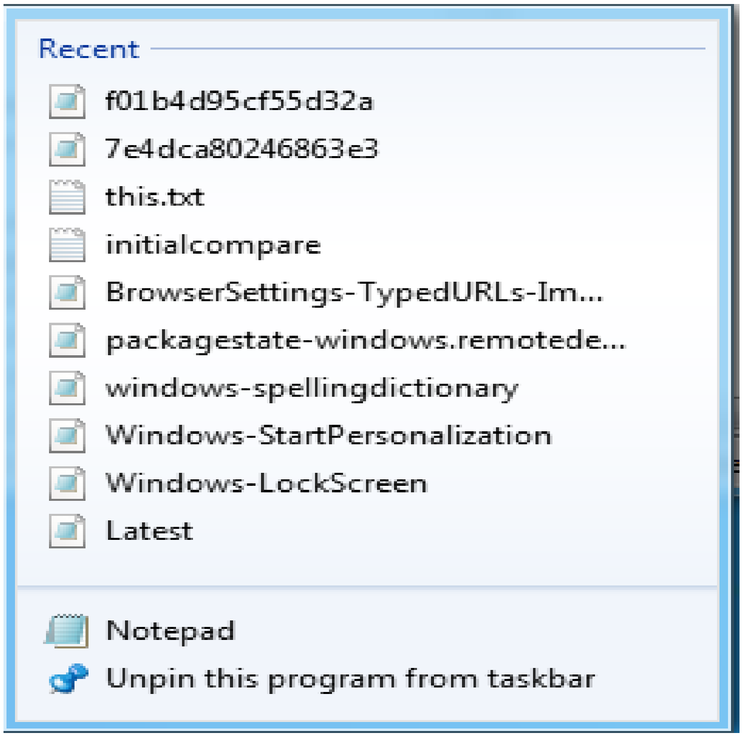
Figure 14

The reason that these lists are important is that it allows digital forensic analysts to see the following:

- Lists of Most Recently Used or Most Frequently Used files opened by the user/application

- List of Most Recently Used or Most Frequently Used by the user/application and how the application was used
- List of most recently or frequently accessed website URLs
- If an application was installed or used/run

Depending on the application that is utilizing the jump list they can utilize different functions. For example the Folder Icon that is on my task bar above shows the most recent or frequent directories I have accessed. While the notepad jump list below shows the most recent files I have accessed with the notepad application.



Recent

f01b4d95cf55d32a

7e4dca80246863e3

this.txt

initialcompare

BrowserSettings-TypedURLs-Im...

packagestate-windows.remotede...

windows-spellingdictionary

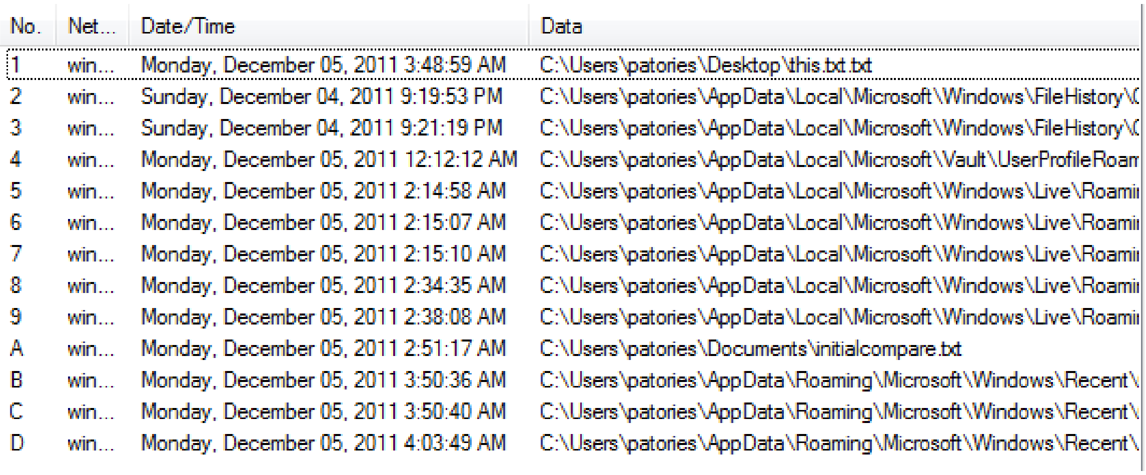
Windows-StartPersonalization

Windows-LockScreen

Latest

Figure 15

When using the JumpLister application I can pull following information back on the files that are listed in my Notepad jump list:



| No. | Net... | Date/Time | Data |
|---|---|---|---|
| 1 | win... | Monday, December 05, 2011 3:48:59 AM | C:\Users\patories\Desktop\this.txt.txt |
| 2 | win... | Sunday, December 04, 2011 9:19:53 PM | C:\Users\patories\AppData\Local\Microsoft\Windows\FileHistory\ |
| 3 | win... | Sunday, December 04, 2011 9:21:19 PM | C:\Users\patories\AppData\Local\Microsoft\Windows\FileHistory\ |
| 4 | win... | Monday, December 05, 2011 12:12:12 AM | C:\Users\patories\AppData\Local\Microsoft\Vault\UserProfileRoam |
| 5 | win... | Monday, December 05, 2011 2:14:58 AM | C:\Users\patories\AppData\Local\Microsoft\Windows\Live\Roami |
| 6 | win... | Monday, December 05, 2011 2:15:07 AM | C:\Users\patories\AppData\Local\Microsoft\Windows\Live\Roami |
| 7 | win... | Monday, December 05, 2011 2:15:10 AM | C:\Users\patories\AppData\Local\Microsoft\Windows\Live\Roami |
| 8 | win... | Monday, December 05, 2011 2:34:35 AM | C:\Users\patories\AppData\Local\Microsoft\Windows\Live\Roami |
| 9 | win... | Monday, December 05, 2011 2:38:08 AM | C:\Users\patories\AppData\Local\Microsoft\Windows\Live\Roami |
| A | win... | Monday, December 05, 2011 2:51:17 AM | C:\Users\patories\Documents\initialcompare.txt |
| B | win... | Monday, December 05, 2011 3:50:36 AM | C:\Users\patories\AppData\Roaming\Microsoft\Windows\Recent\ |
| C | win... | Monday, December 05, 2011 3:50:40 AM | C:\Users\patories\AppData\Roaming\Microsoft\Windows\Recent\ |
| D | win... | Monday, December 05, 2011 4:03:49 AM | C:\Users\patories\AppData\Roaming\Microsoft\Windows\Recent\ |

Figure 16

Knowing the location of the files accessed can prove a vital role in analysis of files and activity on a computer. If the file or folder would have been on another machine the second column would have been blank, or listed the NetBIOS Name of the machine, and the Data Column would include the mapped drive of the artifact.

## Windows 8 Registry Overview

The Windows Registry is a hierarchical database that stores configuration settings and options on the Windows operating systems. It contains settings for low-level operating system components as well as the applications running on the platform. The registry is split up into different files called Hives. The following is a brief description of each hive. (Wikipedia)

*HKEY_CLASSES_ROOT* (HKCR): This key contains information about registered applications, such as file associations, and OLE Object Class IDs, that tie them to the applications that will handle them. (Wikipedia)

HKEY_CURRENT_USER (HKCU): This key will only be found on a live machine, and will only reflect the HKU data for the user that is currently logged in.

HKEY_LOCAL_MACHINE (HKLM): The settings that are stored in the HKLM are specific to the local computer.

HKEY_USERS (HKU): This key is comprised of multiple subfolders for each account this on the machine, and their settings.

*HKEY_CURRENT_CONFIG (HKCC):*This key contains information gathered at runtime, and contains the current configuration settings for the machine.

### *A. Understanding the Registry Values (Carvey, Windows Registry Forensics, 2011)*

There are four core system registry hives within windows that are located in the Windows\system32\config directory. These files are the SAM, Security, System, and software hives. Windows 8 continues with the BCD hive location within the BOOT directory which is stored on the System Reserved Partition. In Windows 7 Microsoft had started moving away from the NTUSER.dat file and replaced it with the USRCLASS.dat hive file, this appears to remain the same, as both files are still available in Windows 8.

### *B. New Registry Values within Windows 8*

With Windows 8 there have been some new registry keys that have been introduced. These registry keys correspond with the new Metro Applications that Microsoft has been promoting within Windows 8, these applications maintain registry entries.

One of these entries which is for IE 10, can be found in the HKU\Software\Microsoft\Internet Explorer\TypedURLsTime. This key stores the time stamp that the corresponding website in the TypedURLs key was visited. This data is saved as a Filetime Object. By having this value retained it adds another location that investigators can compare time stamps for sites visited. This data can be removed if the user clears their browsing history.

With Windows 8, Microsoft is promoting the ability to allow information to be shared across the cloud. Microsoft is talking to application developers about utilizing three types of cloud based storage for their apps:

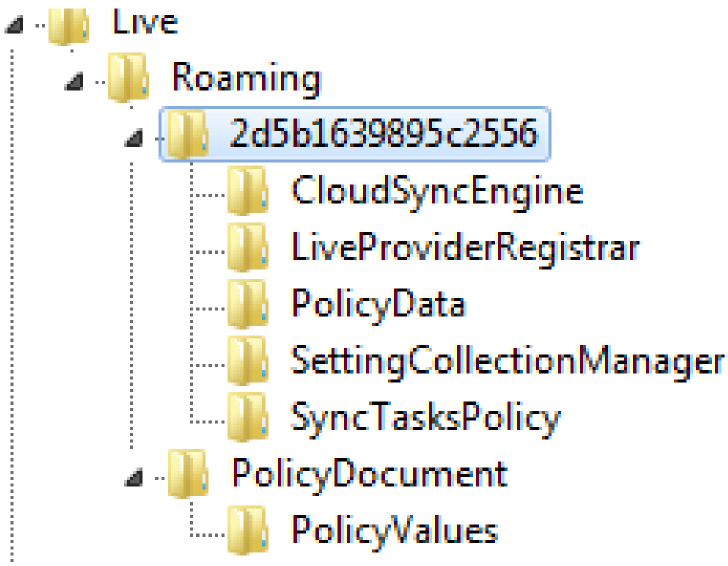
- Windows Live SkyDrive
- Windows Azure Storage
- SQL Azure Database

This fits in well with the ability to create a user account that is tied to your Windows Live ID. (Wegner, 2011) Looking at files within Windows 8 we can see the following directory for a user with a Live ID as their sign-in:

***C:\Users\<user>\AppData\Local\Microsoft\Windows\Live\Roaming\2d5b1639895c2556\CloudSync***

Within the Cloudsync Directory we see multiple SDF files. Some of these files are interesting because of names with Immersive Browser.

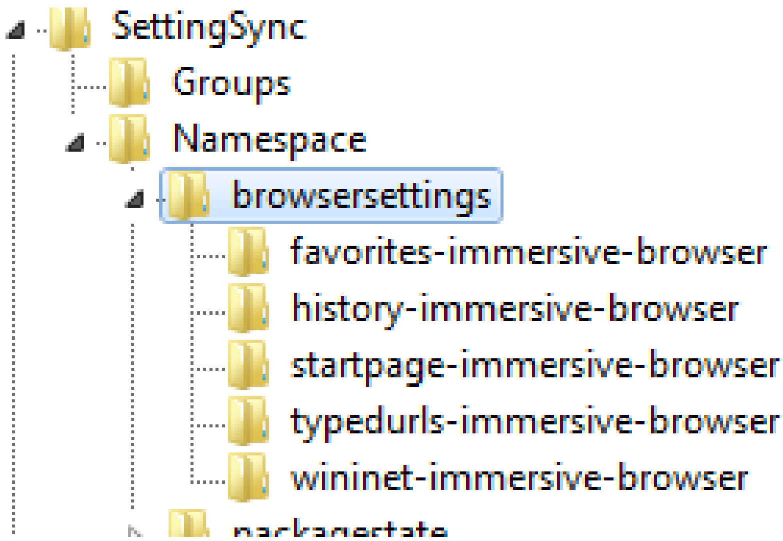
From the registry we can see the following directory and folders that relate to the CloudSync files.



```
Live
  Roaming
    2d5b1639895c2556
      CloudSyncEngine
      LiveProviderRegistrar
      PolicyData
      SettingCollectionManager
      SyncTasksPolicy
    PolicyDocument
      PolicyValues
```

Figure 17

We can see the following registry keys for the Immersive Browser which should be able to provide more information on what actions the user has done, although through testing it appears that these values are redirected into the normal locations. This may change as this operating system becomes more refined and closer to production.



```
SettingSync
    Groups
    Namespace
        browsersettings
            favorites-immersive-browser
            history-immersive-browser
            startpage-immersive-browser
            typedurls-immersive-browser
            wininet-immersive-browser
        packagestate
```

Figure 18

**Windows 8 Recovery Options Overview**

System Restore Points

System Restore Points are created three different ways within Windows 8. Like previous versions of Windows these can be created via System Initiated process, and user initiated process. Within Windows 8, two new registry values have been created that allows applications to initiate the request for systems restore point creation.

I will utilize the artifacts discovered from Thomson's research as well as research of my own to compare between what was expected and how it can be recovered. As with previous versions of windows previous version copies are stored in the Volume Shadow Copies and van be recovered by mounting the drive and extracting the data.

## System Refresh

Windows 8 has introduced the ability for users to recover from malware infection or stability issues by including the refresh option in the operating system. There are two options with the System Refresh that users can utilize. The first option is a default refresh which will revert the operating system back to a factory default setting and a custom refresh that allows the user to define the snapshot scope to revert back to.

In gathering information on what artifacts are impacted and retained from utilizing the System Refresh I will analyze both the Custom and Default against my baseline and compare the difference. I will also look for any new artifacts that are created on the hard drive from this process.

## System Recovery

Windows 8 has introduced the ability for users to quickly reinstall the Operating System from a GUI for the user. The system recovery offers a few options for reinstallation; these options are Quick and Thorough Recovery. The differences between these two options are the ease of which data can be recovered. The quick recovery still allows for easy extraction of files on the machine; while the thorough recovery makes data recovery difficult.

I will look at the impact on the baseline dataset artifacts when I utilize these two recovery options. I will also point out new artifacts that are created when these are run.

# CHAPTER 3

# Experimental Evaluation

## Lab Specification

The lab system that I used for testing was a Windows 7 HP laptop. I ran my Windows 8 systems in a virtualized environment and used freeware tooling to extract my data for analysis, table X lists the software that was utilized for this research. I created multiple snapshots of my Windows 8 VM instance and ran the various recovery methods against them.

Table 3

| Software Name | Purpose |
|---|---|
| VM Workstation 8 & 9 | Virtual Environment for testing |
| FTK Imager | Mounting VM images for analysis |
| FTK ToolKit | Forensic analysis |
| RegRipper | Parsing  registry values from hive files |
| DCode | Conversion of FileTime Format |
| SANS SIFT 2.14 | Alternative tool for file carving and data extraction |

### User Created Dataset

For my user created data sets I utilized various file types and placed them in different locations on the drives. This allowed for the simulation of normal usage and file locations a user might utilize in their daily activities on the system. I have included a sampling of some of my user created data sets in the table below.

Table 4

| File Name | File Type | Original Location |
|---|---|---|
| **DSC_0478** | Image file | My pictures folder |
| **DSC_0481** | Image file | My pictures folder |
| **DSC_0497** | Image file | My pictures folder |
| Tools Directory | Directory | desktop |
| DSC_0499 | Image file | My pictures folder |
| Odbg201d.zip | Zip file | desktop |

### System Created Dataset

For my system created datasets I utilized the artifacts that were created by the system when an application was installed and ran or when a service was initialized. These locations by default should be in the same location across installs. I have recorded the ones that I looked at in the following table and included their original file locations.

Table 5

| File Name | File Type | Original Location |
|---|---|---|
| Config1.xml | File history config file | User Profile, appdata, filehistory |
| Ntuser.dat | Registry hive | User profile |

# CHAPTER 4

# JOURNEY THROUGH THE RECOVERY ARTIFACTS

## System Restore Points

Windows 8 maintains the traditional Restore Points that have been seen in previous versions of windows with a few new tweaks. System Restore points automatically monitors and records key system changes on the computer. It allows users to undo a change that may have caused a problem with the system, or to revert to a day when the system might was preforming optimally. (Microsoft)

Within Windows 8 a new registry key was added that enables application developers to change the frequency of restore-point creation. If the key does not exist then when an application calls the **SRSetRestorePoint** function to create a restore point, Windows skips creating this new restore point if any restore points have been created in the last 24 hours.

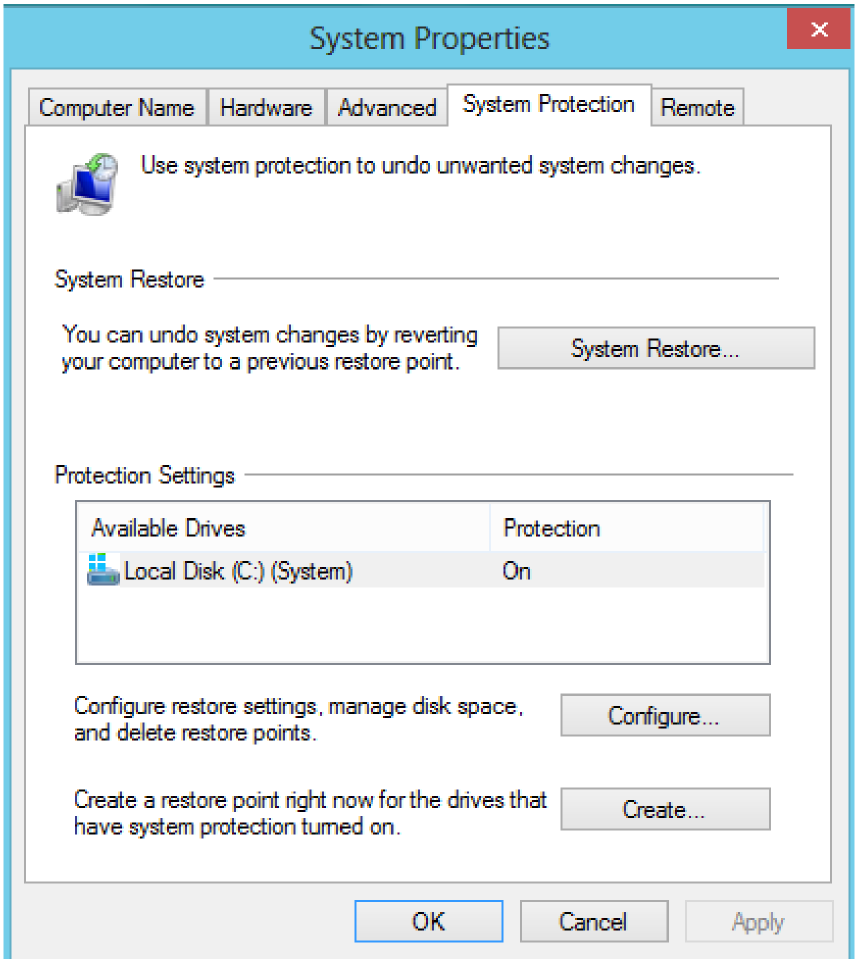
With this new registry key, developers can write applications that create the **DWORD** value **SystemRestorePointCreationFrequency** under the registry key **HKLM\Software\Microsoft\Windows NT\CurrentVersion\SystemRestore**. The value of this registry key can change the frequency of the restore point creation.

When the application calls **SRSetRestorPoint** to create a restore point, and the registry key value is 0, system restore does not skip creating a new restore point. If the application tries to create a restore point, and the registry value is the integer N, than system restore skips creating a new restore point if any restore points were created in the previous N minutes.

System Restore running on Windows 8 monitors files in the boot volume that are relevant for system restore only. Snapshots of the boot volume created by System Restore running on Windows 8 may be deleted if the snapshot is subsequently exposed by an earlier version of Windows.

Developers can write applications that create the **DWORD** value **ScopeSnapshots** under the registry key **HKLM\Software\Microsoft\Windows NT\CurrentVersions\SystemRestore**. If this value is 0, System Restore creates snapshots of the boot volume in the same way as earlier versions of Windows. If this value is deleted, System Restore running on Windows 8 resumes creating snapshots that monitor files in the boot volume that are relevant for the system restore only.

When a user initiates a System Restore Point creation they are greeted with a menu that offers a chance to Create, Configure or to proceed with a System Restore Recovery.



Use system protection to undo unwanted system changes.

You can undo system changes by reverting your computer to a previous restore point.

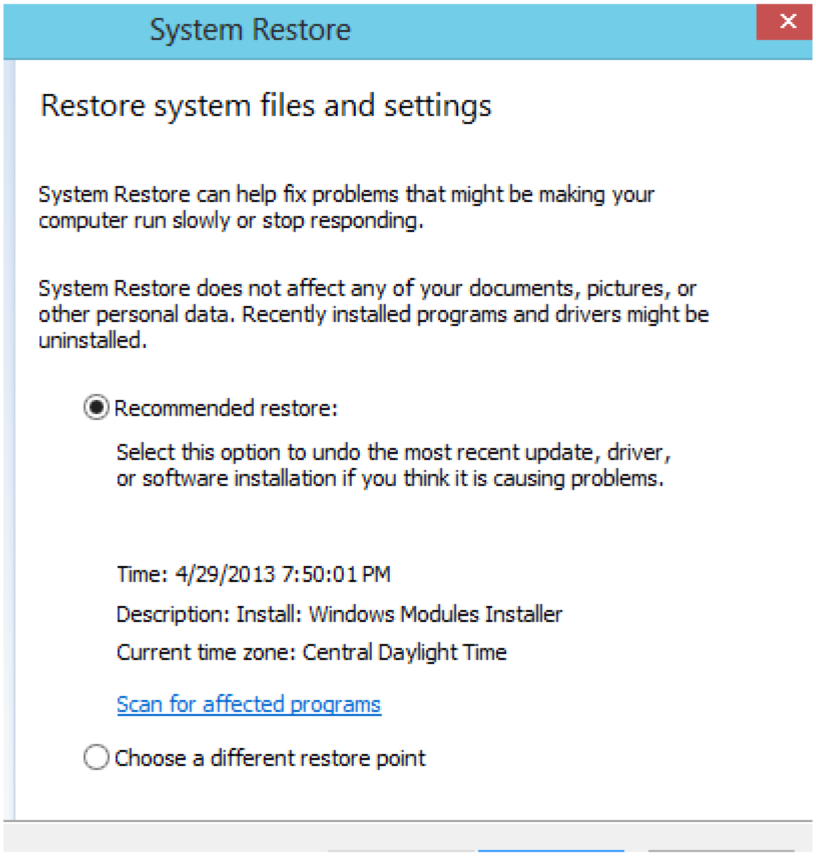
| Available Drives | Protection |
|---|---|
| Local Disk (C:) (System) | On |

Configure restore settings, manage disk space, and delete restore points.

Create a restore point right now for the drives that have system protection turned on.

Figure 19

If the user proceeds with a system restore they can choose to use a recommended restore, select a different restore point or scan the restore point to see affected programs.



Restore system files and settings

System Restore can help fix problems that might be making your computer run slowly or stop responding.

System Restore does not affect any of your documents, pictures, or other personal data. Recently installed programs and drivers might be uninstalled.

● Recommended restore:

Select this option to undo the most recent update, driver, or software installation if you think it is causing problems.

Time: 4/29/2013 7:50:01 PM

Description: Install: Windows Modules Installer

Current time zone: Central Daylight Time

Scan for affected programs

○ Choose a different restore point

Figure 20

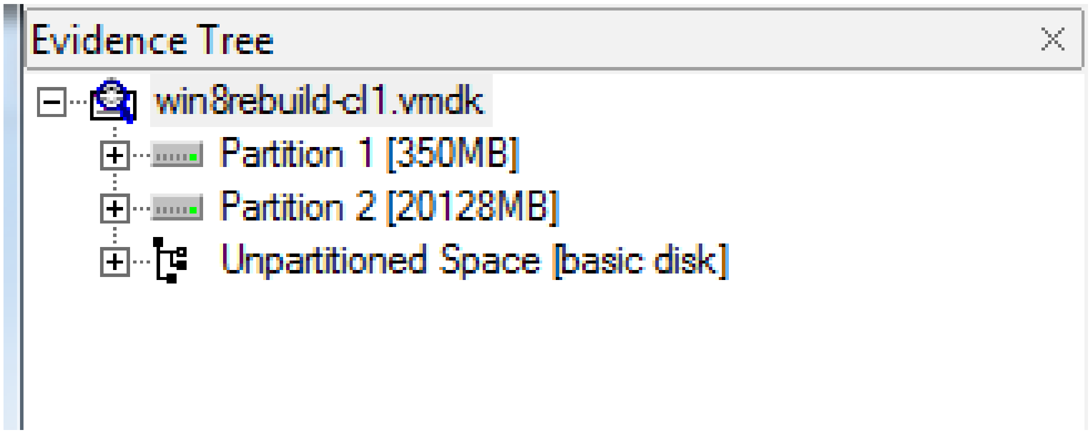
### System Refresh Points

Windows 8 introduces two new options for system recovery, these options are: Refresh Points and System Recovery. Within Refresh Point there are two options; you can utilize the default refresh point or a custom refresh point.

Both Refresh options can be utilized by Windows 8 to remove malicious files and corrupted entries into the operating system. When using Refresh it is important to understand that the operating system creates a Recovery Image that makes a backup of the Windows System Files. For the default recover these Windows System Files are from when Windows 8 was first installed. When the Custom Refresh option is used than the Windows System Files are from the date that the Custom Refresh was created, the Custom Refresh also will contain the desktop applications that you have installed. Refresh Images **DO NOT** contain your Metro-style apps, documents, personal settings or user profiles, this is because that information is preserved at the time you refresh your PC.

The System Recover option in Windows 8 will return the Operating system back to the factory default. While using the System Recover there will be options on Using Recover with Multiple Drives, and how personal files are removed.

When looking at an image of the Windows 8 operating system from within the AccessData FTK Imager there are three things that are quickly noticed. There are two partitions and an unpartitioned space. This is similar to previous versions of Windows.



Evidence Tree

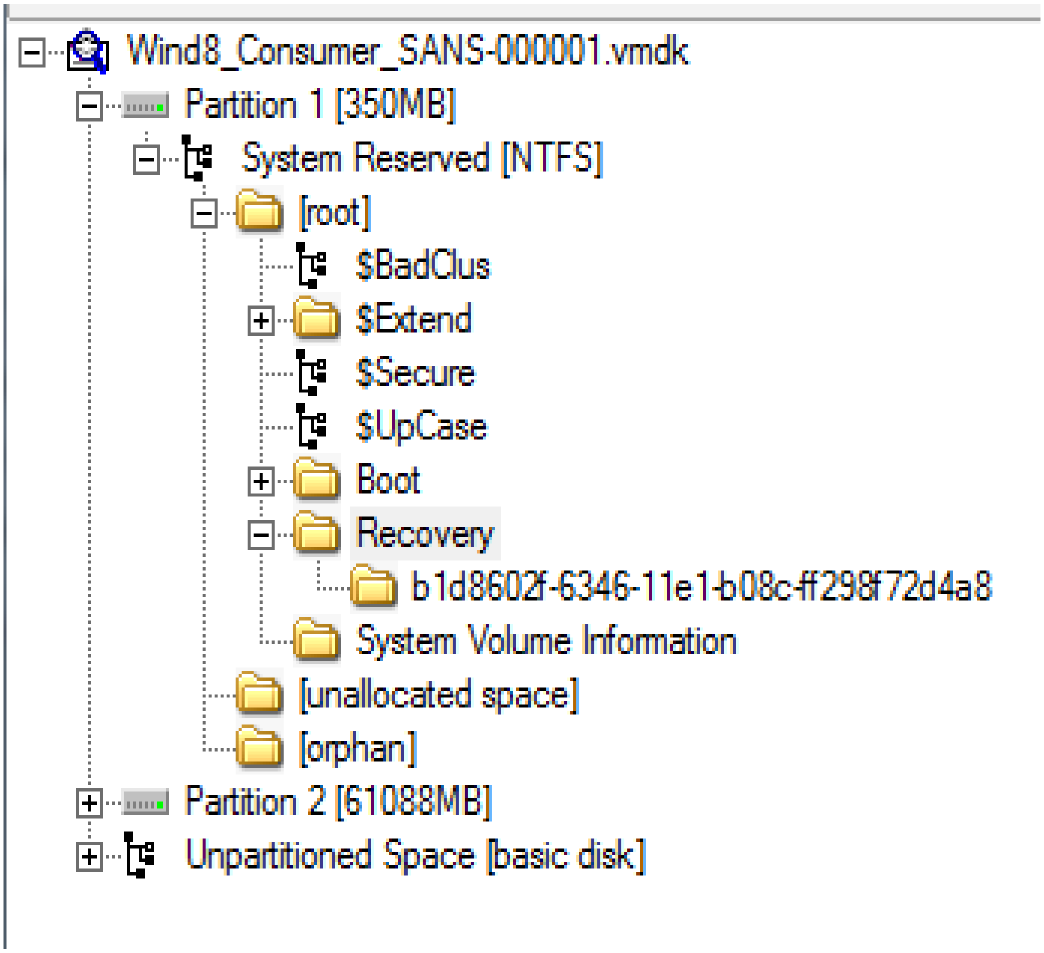
- win8rebuild-cl1.vmdk
  - Partition 1 [350MB]
  - Partition 2 [20128MB]
  - Unpartitioned Space [basic disk]

Figure 21

Windows system recovery partitionis a 350MB partition, and is dedicated to the basic root of the operating system. This will contain information related to the refresh process of the machine. I noticed that from the Developer Preview to the Final Release some of the Directory Artifacts locations were changed. Both Versions contained a recovery folder, but the Developer Preview contained some more artifacts for comparison. Both the System Refresh and the System Recover will leave their artifacts in similar locations.

One of the more interesting files that can be found in this partition is the Reload.xml. This file is found in the Root/Recovery/Logs and contains information for the OS to refresh.

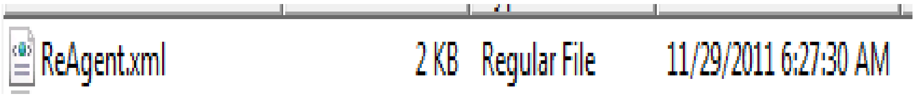
These next two images are the Windows system recovery partition from the same machine but at different points in the recovery process. The first one is taken from a machine that has not had refresh or recover ran on it. The second one is after a System refresh has been ran. As you can see when a refresh option is ran on a Windows 8 machine there is a creation of a directory called Log in the recovery partition.



```
Wind8_Consumer_SANS-000001.vmdk
    Partition 1 [350MB]
        System Reserved [NTFS]
            [root]
                $BadClus
                $Extend
                $Secure
                $UpCase
                Boot
                Recovery
                    b1d8602f-6346-11e1-b08c-ff298f72d4a8
                System Volume Information
            [unallocated space]
            [orphan]
    Partition 2 [61088MB]
    Unpartitioned Space [basic disk]
```

Figure 22

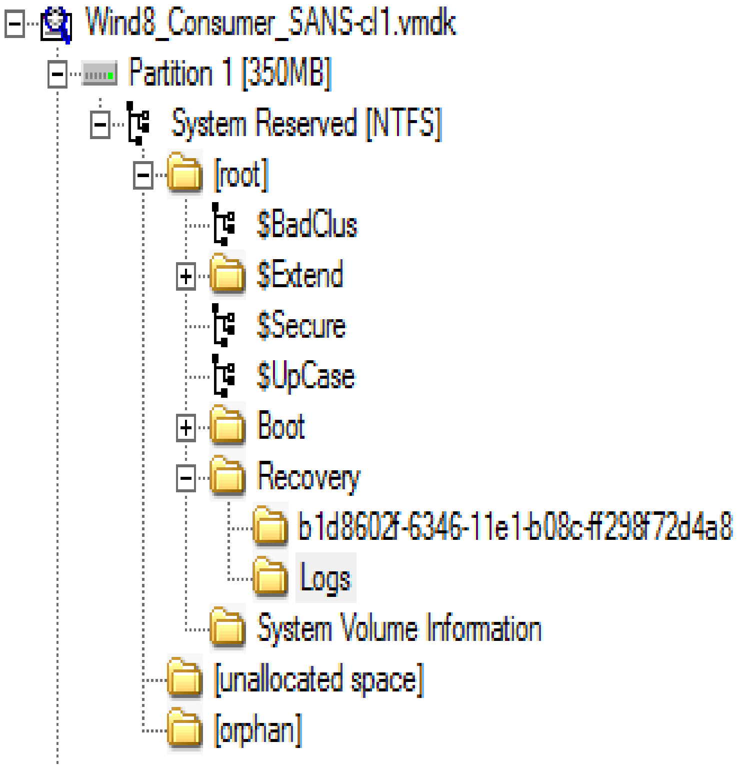
This figure is what the recovery partition appears before a recovery option is ran on the system. In this case the directory called **b1d8602f-6346-11e1-b08c-ff298f72d4a8** will contain a file called ReAgent.XML. This file will contain the current configuration data for the Refresh or Recovery process to run.



ReAgent.xml 2 KB Regular File 11/29/2011 6:27:30 AM

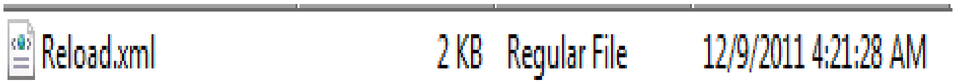
Figure 23

The next figure includes the creation of the directory called logs. This directory will contain a file called Reload.xml. This file will contain the configuration data that was used to refresh the system. Since this can be different from the ReAgent.xml it might provide more clues on what was configured when.



```
Wind8_Consumer_SANS-cl1.vmdk
  Partition 1 [350MB]
    System Reserved [NTFS]
      [root]
        $BadClus
        $Extend
        $Secure
        $UpCase
        Boot
        Recovery
          b1d8602f-6346-11e1-b08c-ff298f72d4a8
          Logs
        System Volume Information
      [unallocated space]
      [orphan]
```

Figure 24



```
Reload.xml                2 KB   Regular File   12/9/2011 4:21:28 AM
```

Figure 25

Both the Reload.xml and the ReAgent.xml contains time stamps, the Reload indicates when the Refresh process was started, while the ReAgent.xml indicates when it was finished, or if it was modified after the Refresh.

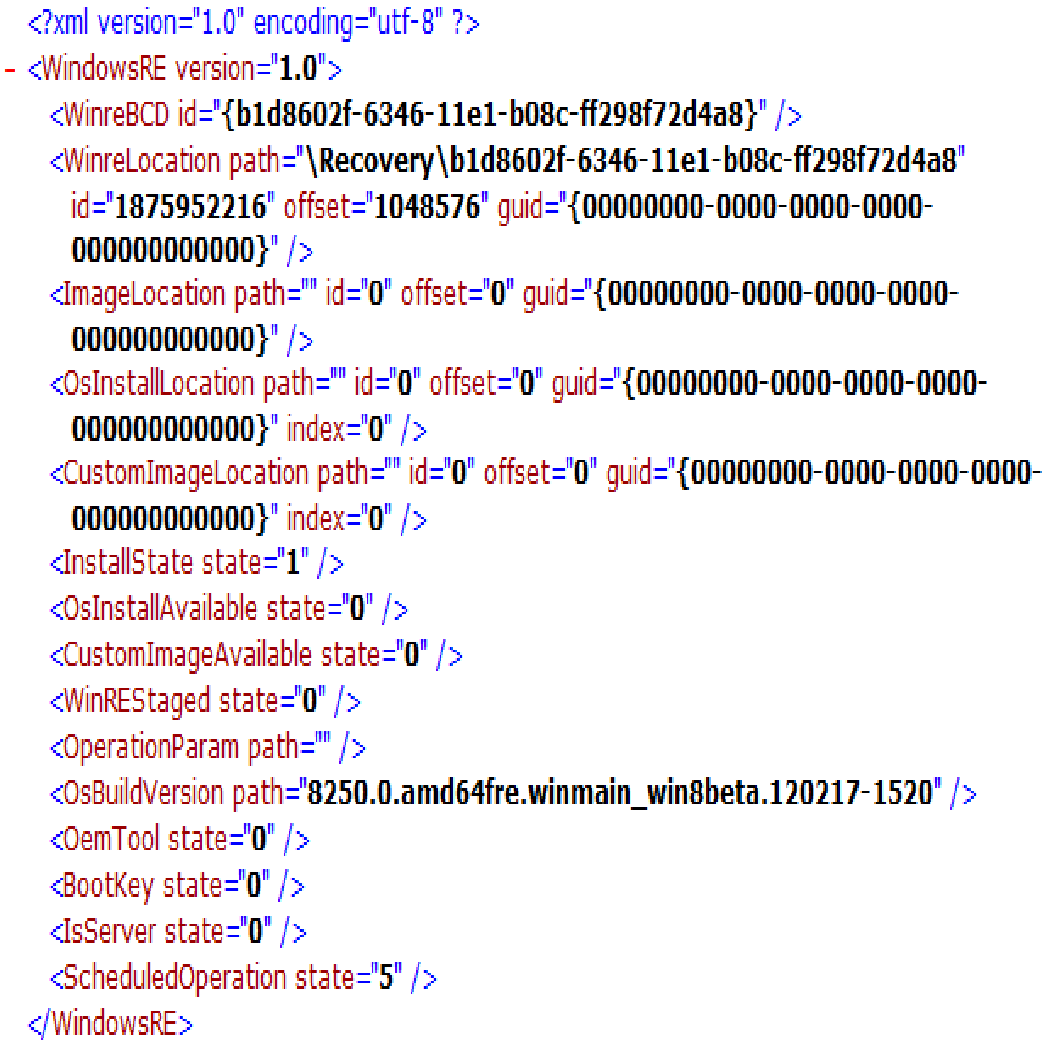
Within the ReAgent.xml we can see various configuration options for the System Refresh as well as see if it is a default or custom refresh. The first screen shot is the default refresh, the second one is from a custom refresh stored locally, and the final one is a custom refresh stored on a remote drive.

As we can see the images share similar data between them. This includes the following:

| Option | Description |
|---|---|
| WinreBCD ID | is the same identifier of the folder on the system drive |
| WinRE Location Path | Where the WIMRE.wim file is found on the system drive. |
| CustomImageLocation | Where the Custom Refresh Image is stored |

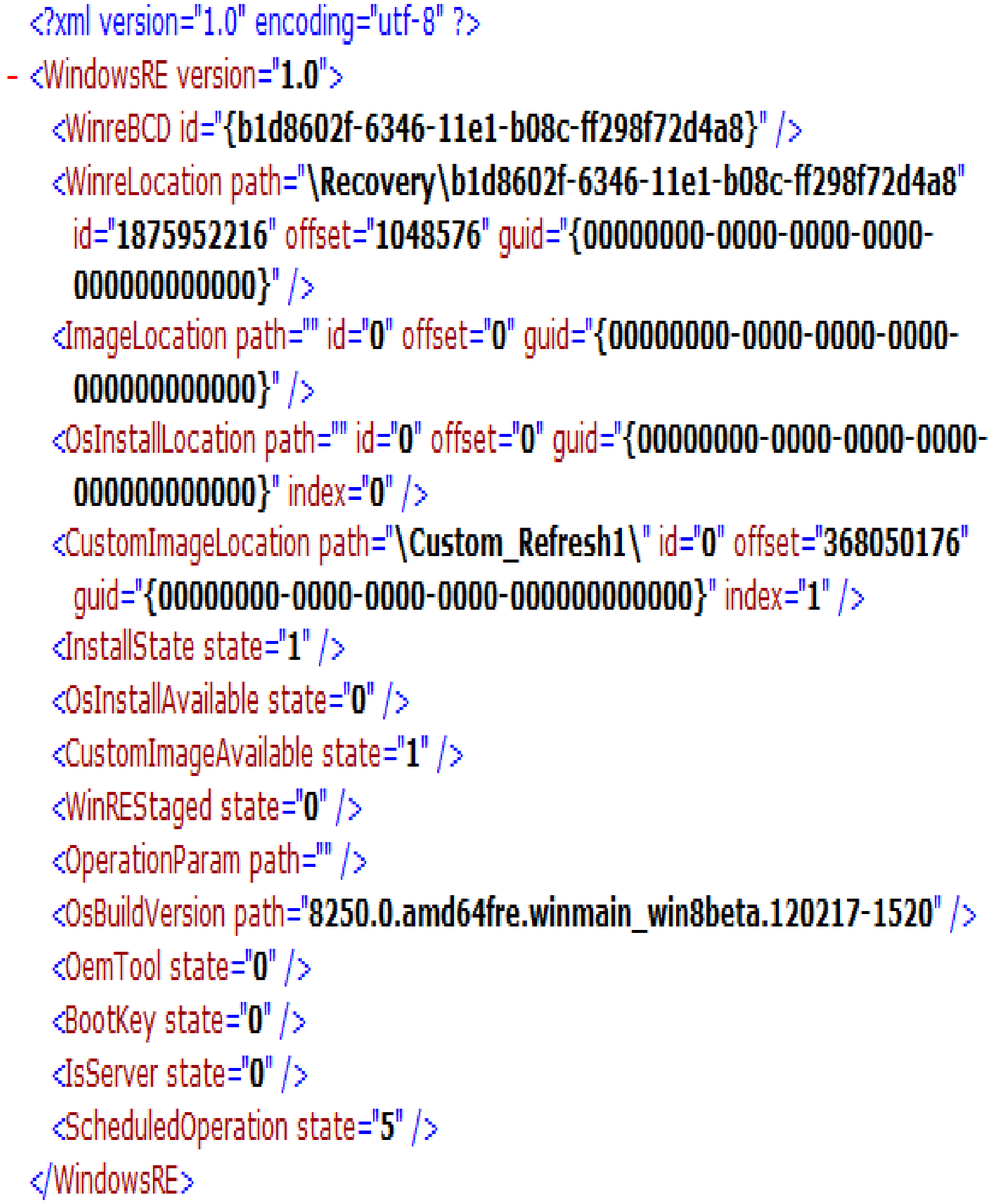
Table 6

The first image is the ReAgent.xml from a Refresh Point stored on the C: drive, while the second image is a Refresh Point stored on a 2nd hard drive. As you can see the harddisk location is not mentioned in the CustomImageLocation, and the offset value has changed. Currently I am trying to figure out what the offset value means.



```
  <?xml version="1.0" encoding="utf-8" ?>
- <WindowsRE version="1.0">
    <WinreBCD id="{b1d8602f-6346-11e1-b08c-ff298f72d4a8}" />
    <WinreLocation path="\Recovery\b1d8602f-6346-11e1-b08c-ff298f72d4a8"
      id="1875952216" offset="1048576" guid="{00000000-0000-0000-0000-
      000000000000}" />
    <ImageLocation path="" id="0" offset="0" guid="{00000000-0000-0000-0000-
      000000000000}" />
    <OsInstallLocation path="" id="0" offset="0" guid="{00000000-0000-0000-0000-
      000000000000}" index="0" />
    <CustomImageLocation path="" id="0" offset="0" guid="{00000000-0000-0000-0000-
      000000000000}" index="0" />
    <InstallState state="1" />
    <OsInstallAvailable state="0" />
    <CustomImageAvailable state="0" />
    <WinREStaged state="0" />
    <OperationParam path="" />
    <OsBuildVersion path="8250.0.amd64fre.winmain_win8beta.120217-1520" />
    <OemTool state="0" />
    <BootKey state="0" />
    <IsServer state="0" />
    <ScheduledOperation state="5" />
  </WindowsRE>
```

**Figure 26**

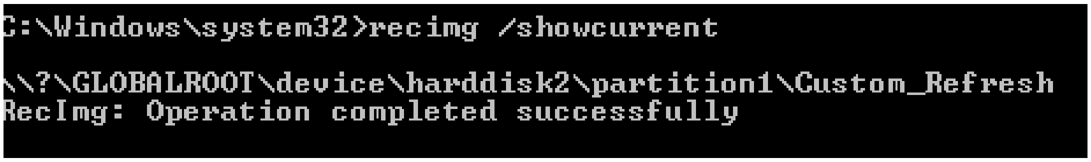


```
  <?xml version="1.0" encoding="utf-8" ?>
- <WindowsRE version="1.0">
    <WinreBCD id="{b1d8602f-6346-11e1-b08c-ff298f72d4a8}" />
    <WinreLocation path="\Recovery\b1d8602f-6346-11e1-b08c-ff298f72d4a8"
      id="1875952216" offset="1048576" guid="{00000000-0000-0000-0000-
      000000000000}" />
    <ImageLocation path="" id="0" offset="0" guid="{00000000-0000-0000-0000-
      000000000000}" />
    <OsInstallLocation path="" id="0" offset="0" guid="{00000000-0000-0000-0000-
      000000000000}" index="0" />
    <CustomImageLocation path="\Custom_Refresh1\" id="0" offset="368050176"
      guid="{00000000-0000-0000-0000-000000000000}" index="1" />
    <InstallState state="1" />
    <OsInstallAvailable state="0" />
    <CustomImageAvailable state="1" />
    <WinREStaged state="0" />
    <OperationParam path="" />
    <OsBuildVersion path="8250.0.amd64fre.winmain_win8beta.120217-1520" />
    <OemTool state="0" />
    <BootKey state="0" />
    <IsServer state="0" />
    <ScheduledOperation state="5" />
  </WindowsRE>
```

Figure 27

To create a custom recovery image, you need to use the recimg.exe. When you create a custom recovery image, recimg will store it in the specified directory, and it is set as the active image. If a custom recovery image is set as the active recovery image, Windows will use it when you refresh your PC. All recovery images have the filename install.wim. If no install.wim file is fund in the active recovery image directory, Windows will fall back to the default image.

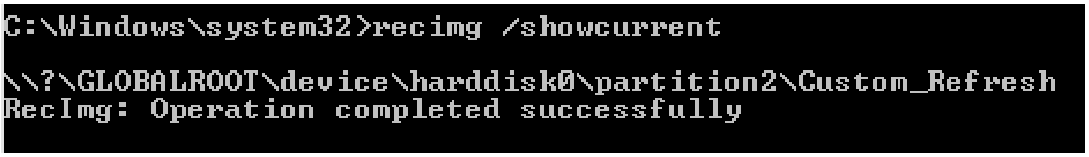
After you have created a custom image, or changed active recovery images you will see the following showing what the current recovery image is. The first image is the default location, the second is on a removable drive, and the final one is located on the C:\ drive of the computer.



```
C:\Windows\system32>recimg /showcurrent

\\?\GLOBALROOT\device\harddisk2\partition1\Custom_Refresh
RecImg: Operation completed successfully
```

Figure 28



```
C:\Windows\system32>recimg /showcurrent

\\?\GLOBALROOT\device\harddisk0\partition2\Custom_Refresh
RecImg: Operation completed successfully
```

Figure 29

Artifacts that can be found on Partition1 after a refresh includes the current refresh as well as the most recent refresh that happened.

Windows system partition is the actual system partition. On a refresh this partition contains the following folders: root, orphan and unallocated space. It also contains the files called backup boot sector and file system slack. The root folder contains the files and settings used within the operating system. Some key information that can be found in this partition Registry Hives, File History configuration and data, contacts, documents, windows.old and other locations. Depending on how recent the refresh happened there might be a HTML file on the desktop that lists what applications are removed.

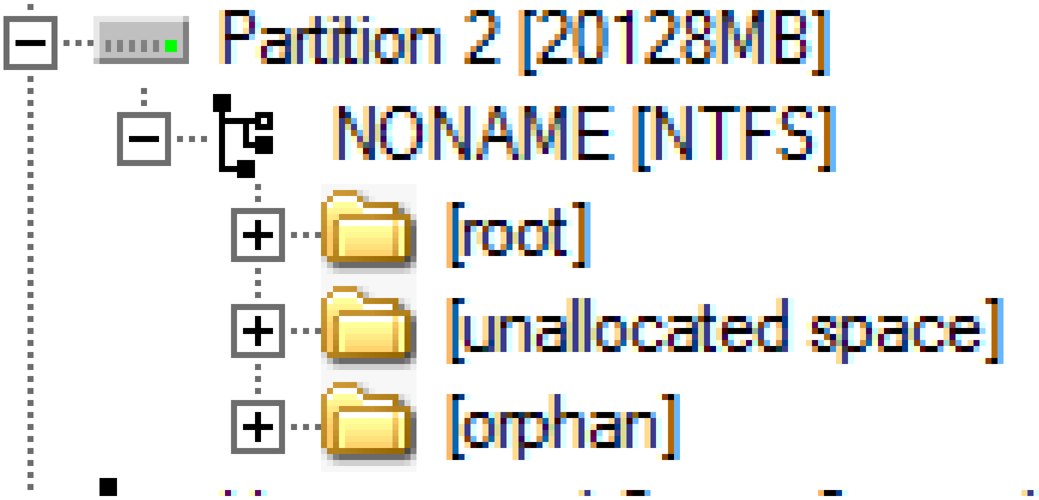
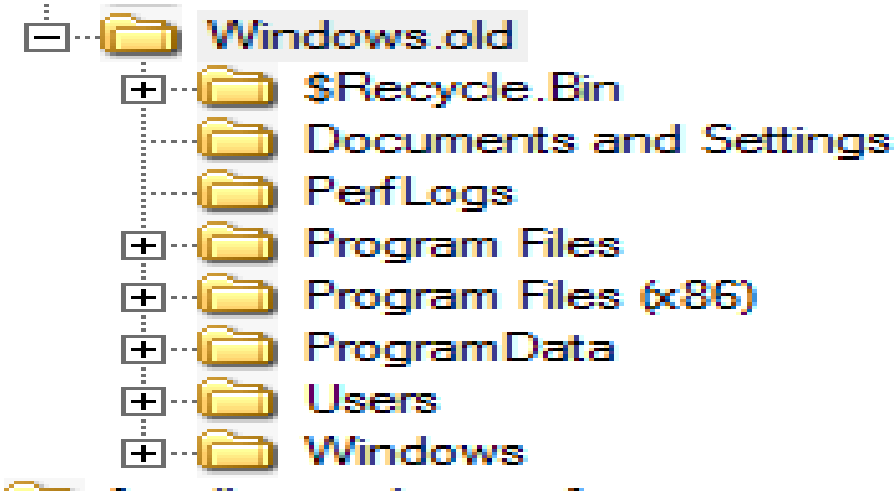


Figure 30

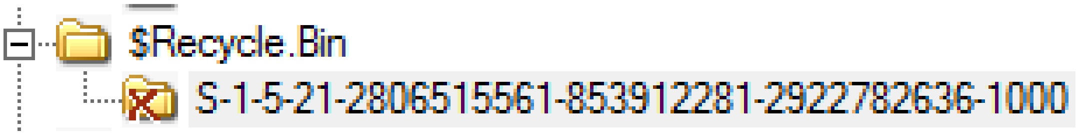
The windows.old folder, which is located in the root directory holds artifacts of what files and programs were installed on the machine before the refresh. From this information it appears that on a refresh nothing is uninstalled or physically deleted from the hard drive but marked to allow the space being over written, you can still access them through devices such as FTK Imager.



Windows.old
$Recycle.Bin
Documents and Settings
PerfLogs
Program Files
Program Files (x86)
ProgramData
Users
Windows

Figure 31

When utilizing FTK Imager those files that are marked for deletion have a Red X over the icon.



$Recycle.Bin
S-1-5-21-2806515561-853912281-2922782636-1000

Figure 32

The files in Windows system partition that are marked for deletion can still be accessed and analyzed as normal. I was able to pull the Registry Hive files and run them through RegRipper and Registry Decoder for data analysis.

The unpartitioned Space, contains unallocated file space, while there might be information stored there that will have value, I did not find anything in this research scope. The Windows system partition contains a wealth of information that should be interesting to an investigator.

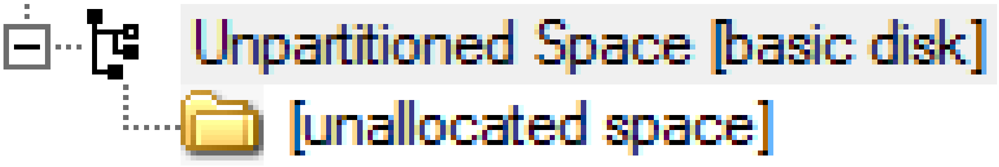
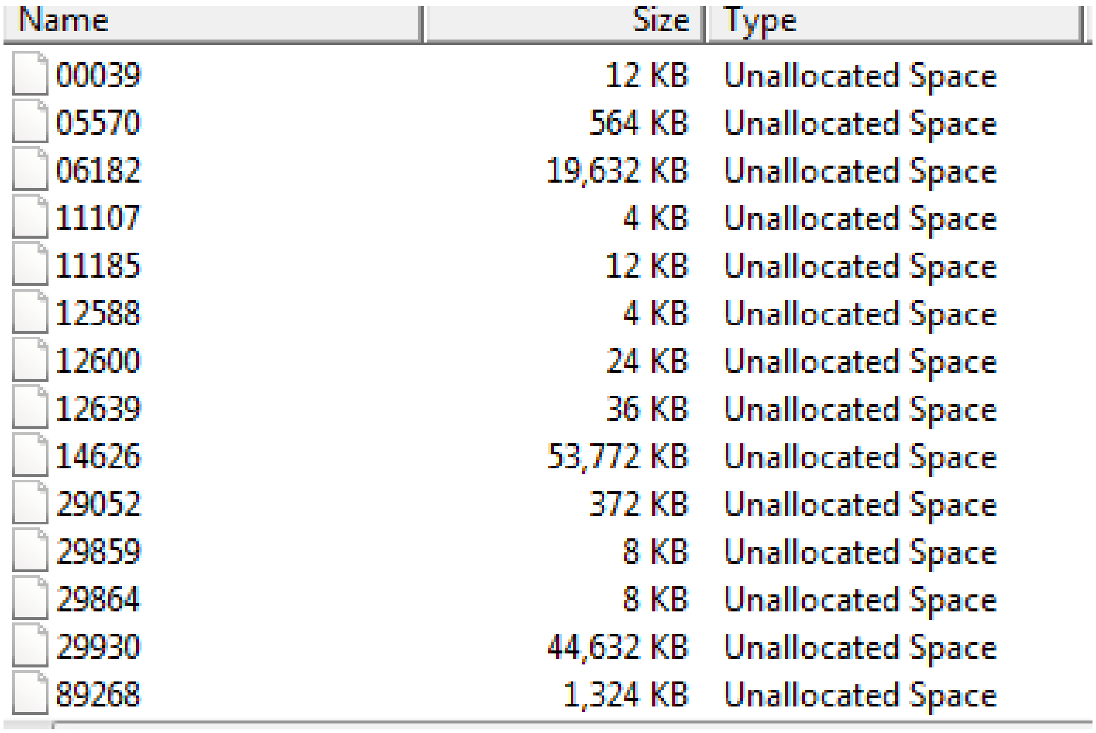


Figure 33

## System Reset Artifacts

When analyzing a disk that has had System Reset used on to restore back to factory default it is advised to keep in mind that this process will completely wipe everything. Looking at the disk image within FTK Imager it can be seen that there are 3 sections, the Windows system recovery partition, the Windows system partition, and the unpartitioned space.

Windows system recovery partition is a 350MB partition, and is dedicated to the basic root of the operating system. This will contain information related to the operation of the machine. One of the more interesting files that can be found in this partition is the Reload.xml which can be found in the Root/Recovery/Logs and the ReAgent.xml file which is found in the Root/Recovery/ file path both of these files contain information utilized by the operating system for the restore or refresh action. The unallocated space in Windows system recovery partition contains numerous files of unallocated space in various sizes. More research will need to be done to understand what is on those unallocated clusters, if anything.

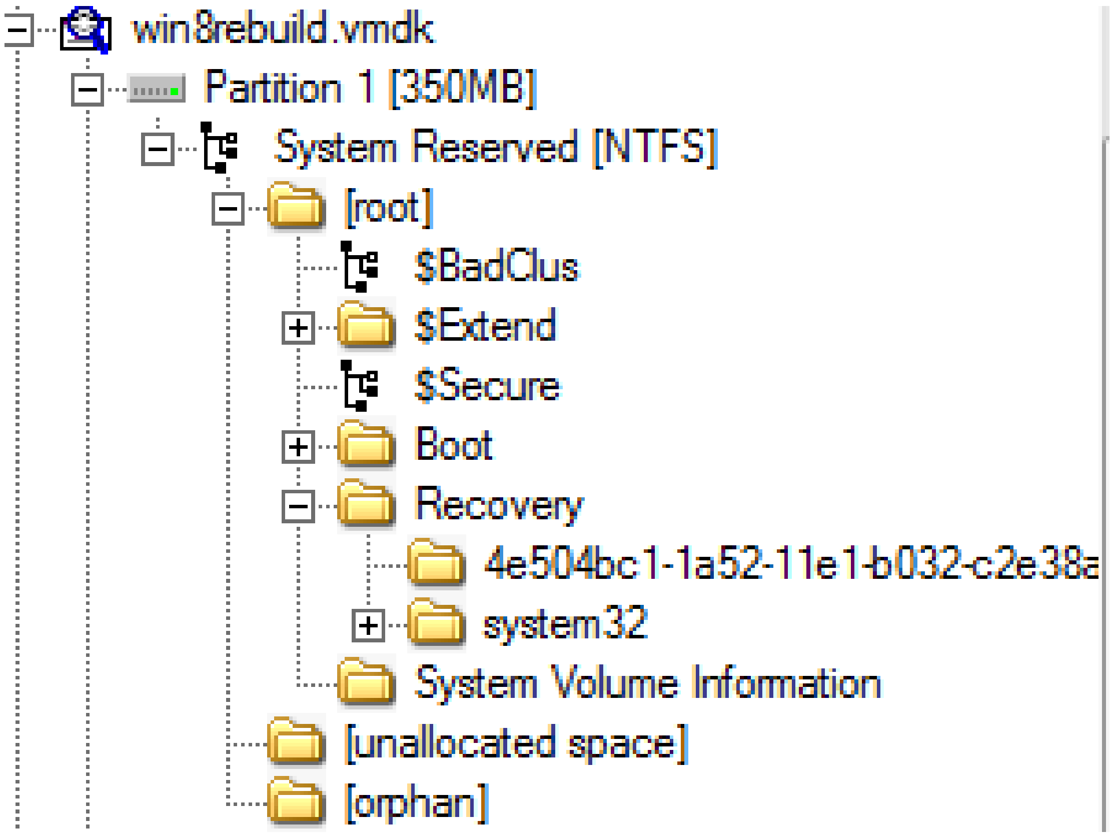


| Name | Size | Type |
|---|---|---|
| 00039 | 12 KB | Unallocated Space |
| 05570 | 564 KB | Unallocated Space |
| 06182 | 19,632 KB | Unallocated Space |
| 11107 | 4 KB | Unallocated Space |
| 11185 | 12 KB | Unallocated Space |
| 12588 | 4 KB | Unallocated Space |
| 12600 | 24 KB | Unallocated Space |
| 12639 | 36 KB | Unallocated Space |
| 14626 | 53,772 KB | Unallocated Space |
| 29052 | 372 KB | Unallocated Space |
| 29859 | 8 KB | Unallocated Space |
| 29864 | 8 KB | Unallocated Space |
| 29930 | 44,632 KB | Unallocated Space |
| 89268 | 1,324 KB | Unallocated Space |

Figure 34

Windows system partition in a freshly restore image contains three primary folders, Root, Unallocated Space and Orphan. Looking into the Orphan and Unallocated space files we can see that the orphan folder is empty, but the unallocated space contains numerous unallocated spaces of varying sizes. Looking through the files contained within Windows system partition, there appears to be little forensic evidence that the system has been recently restored, save for the lack of personalized data that is stored. In order to see evidence of a restore on the machine we actually need to return our investigation to the Windows system recovery partition.

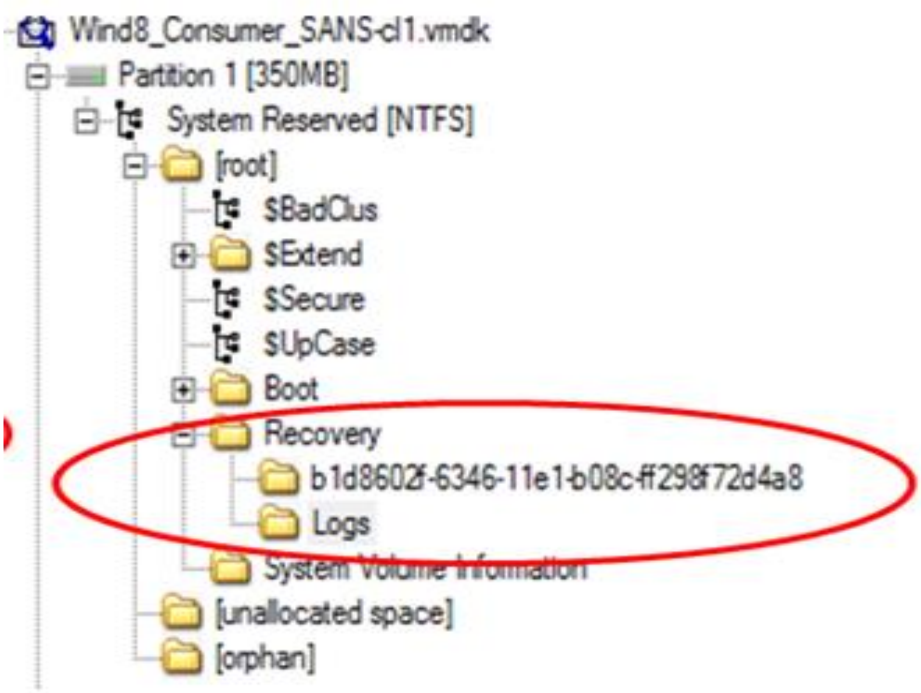
To find this proof we need to understand the folders that are initially created in this space, as the image below shows these are the following folders within Windows system recovery partition before a restore or refresh is done.



```
win8rebuild.vmdk
  Partition 1 [350MB]
    System Reserved [NTFS]
      [root]
        $BadClus
        $Extend
        $Secure
        Boot
        Recovery
          4e504bc1-1a52-11e1-b032-c2e38a
          system32
        System Volume Information
      [unallocated space]
      [orphan]
```

Figure 35

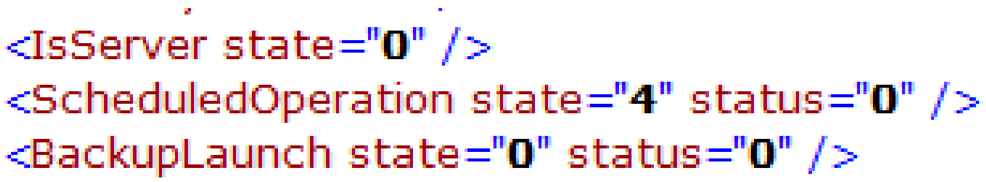
The following are the files under the same machine after a restore of the operating system. As you can see from the two images, the restored or refreshed machine contains a new folder called Logs.



```
Wind8_Consumer_SANS-cl1.vmdk
  Partition 1 [350MB]
    System Reserved [NTFS]
      [root]
        $BadClus
        $Extend
        $Secure
        $UpCase
        Boot
        Recovery
          b1d8602f-6346-11e1-b08c-ff298f72d4a8
          Logs
        System Volume Information
      [unallocated space]
      [orphan]
```

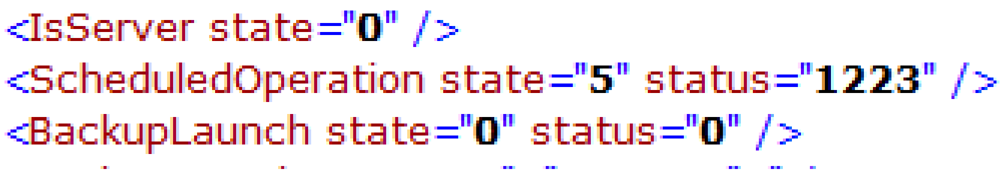
Figure 36

If we look into the Logs file we will see the file called Reload.xml within this file has some information related the refresh/restore process. We also notice that there is a file in the system32/Recovery called ReAgent.xml, this file can be found across all three images; Base, Restore and Refresh. When comparing the ReAgent file across the three platforms the files are identical. When comparing the ReAgent and the Reload files against each other they are identical except for the following line ScheduleOperation State. The first one is ReAgent, the second is the reload file.



```
<IsServer state="0" />
<ScheduledOperation state="4" status="0" />
<BackupLaunch state="0" status="0" />
```

Figure 37



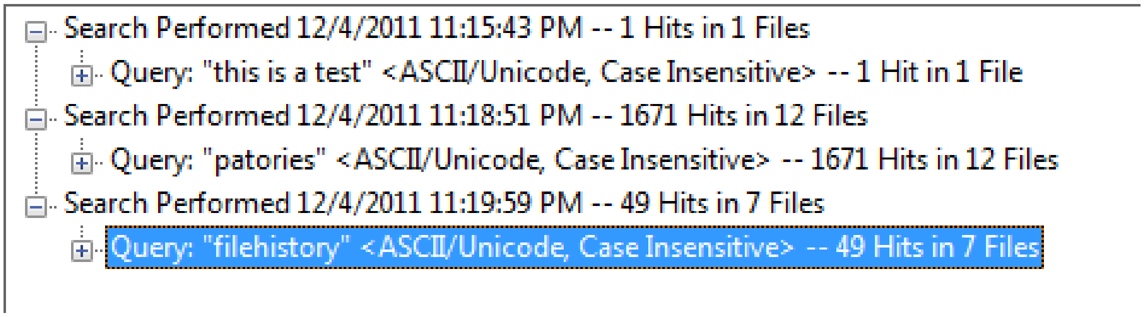
```
<IsServer state="0" />
<ScheduledOperation state="5" status="1223" />
<BackupLaunch state="0" status="0" />
```

Figure 38

The Reload.xml file is identical across both a Restore and a Refresh process. So this value must dictate that certain files where restored to the factory default. This may change with machines that are running the Unified Extensible Firmware Interface.

While comparing these two files we notice that the Date Modified Time on the ReAgent file is earlier then the modified date on the Reload file. The date on the Reload file is the date that the machine was refreshed and reloaded, this is another clue that the machine was refreshed or reloaded.

Even though Windows 8, has the ability to reset the operating system back to factory default, it is not impossible to do analysis and data carving on the unallocated spaces to find artifacts that once remained on the machine. As you can see from the image below, even after I did a reset on the machine, which should have returned everything to the factory default I was still able to carve out my test file, my username, and some files related to the file history option.



```
Search Performed 12/4/2011 11:15:43 PM -- 1 Hits in 1 Files
    Query: "this is a test" <ASCII/Unicode, Case Insensitive> -- 1 Hit in 1 File
Search Performed 12/4/2011 11:18:51 PM -- 1671 Hits in 12 Files
    Query: "patories" <ASCII/Unicode, Case Insensitive> -- 1671 Hits in 12 Files
Search Performed 12/4/2011 11:19:59 PM -- 49 Hits in 7 Files
    Query: "filehistory" <ASCII/Unicode, Case Insensitive> -- 49 Hits in 7 Files
```

Figure 39

# CHAPTER 5

# WINDOWS 8 FORENSIC CASE STUDIES

## Base System Artifacts

Table 7

| Artifact Name | Location |
|---|---|
| **DSC_0478** | **Picture Library** |
| **DSC_0481** | **Picture Library** |
| **DSC_0497** | **Picture Library** |

## System Restore Points Artifacts

### System Restore Initial Configuration

For my initial configuration for testing system restore points I created a snapshot with my virtual machine for a quick restore of my test environment. I made sure that some of my artifacts were present prior to the creation of my restore point. After creating the restore point I finished adding the remaining artifacts I wished to test, and I deleted a couple of previously installed artifacts. Based on previous versions of the system restore points behavioral assumption of this test is that all artifacts created prior to the creation of the restore point would remain or be restored. Those that were created after the restore point was created would be lost.

System Restore Findings

I have taken three screenshots of my Pictures directory throughout this testing process. The first image is prior to the creation of the restore point. Within this directory are three images DSC_0478, DSC_0481 and DSC_0497. The second image is after I have added DSC_0499 to the directory, removed DSC_0481 and after the restore point has been created.

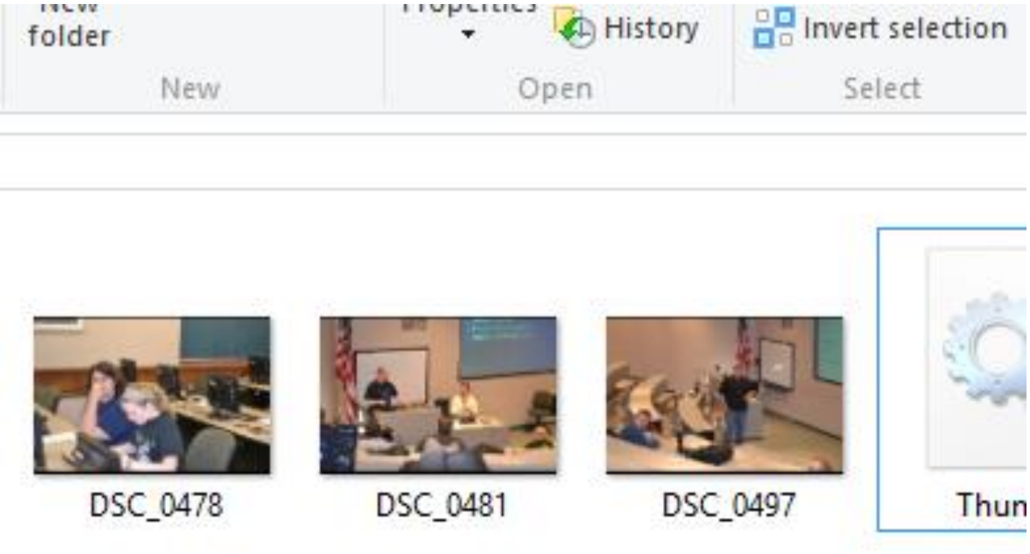


Figure 40

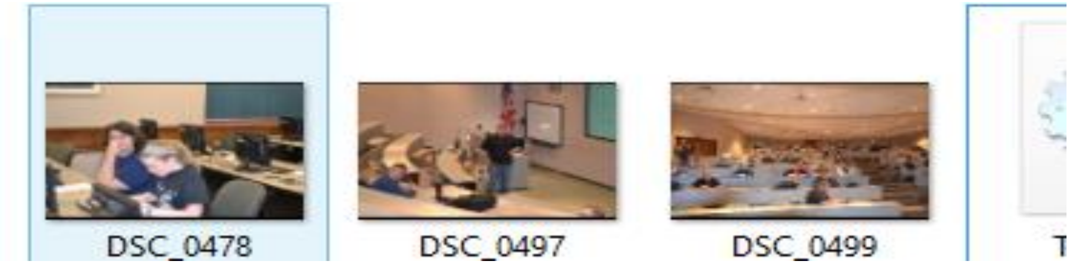
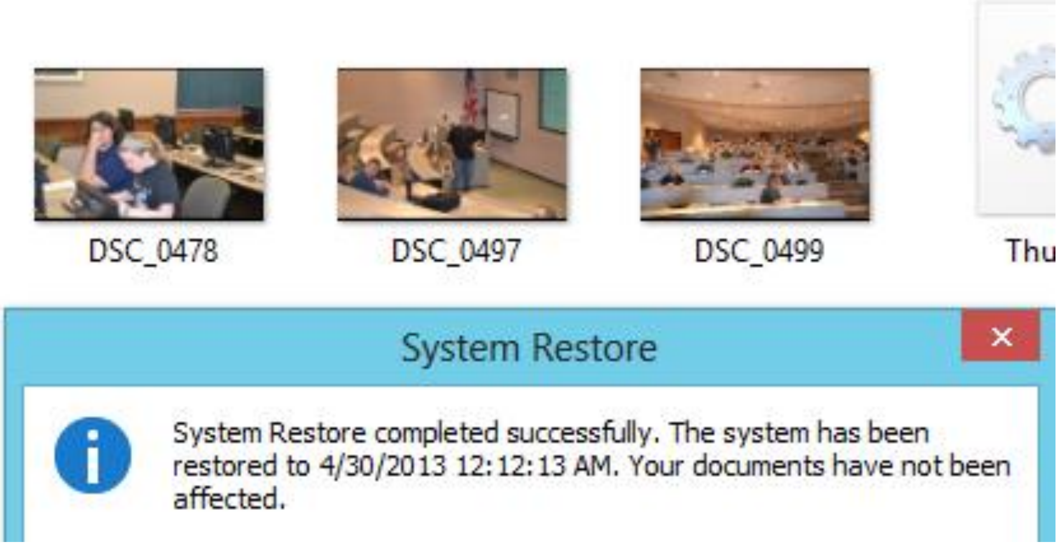


Figure 41

Looking at my Pictures Directory I was surprised to see that my DSC_0481 was not restored to the restore point. I became curious to see if my desktop had changed.



System Restore completed successfully. The system has been restored to 4/30/2013 12:12:13 AM. Your documents have not been affected.

Figure 42

This next set of data is stored on my desktop. In the initial screen shot there are multiple file types in place. In this testing I was concerned about the windows 8 – Fact.ppt, the Tools directory, and the 20121025104233 – WIN8_TESTBED Incident.7z files. In image X I show that I have added the zipped file called odbg201d, deleted the Tools Directory before utilizing the restore point. The final image in this set shows my desktop after I have utilized the restore point.

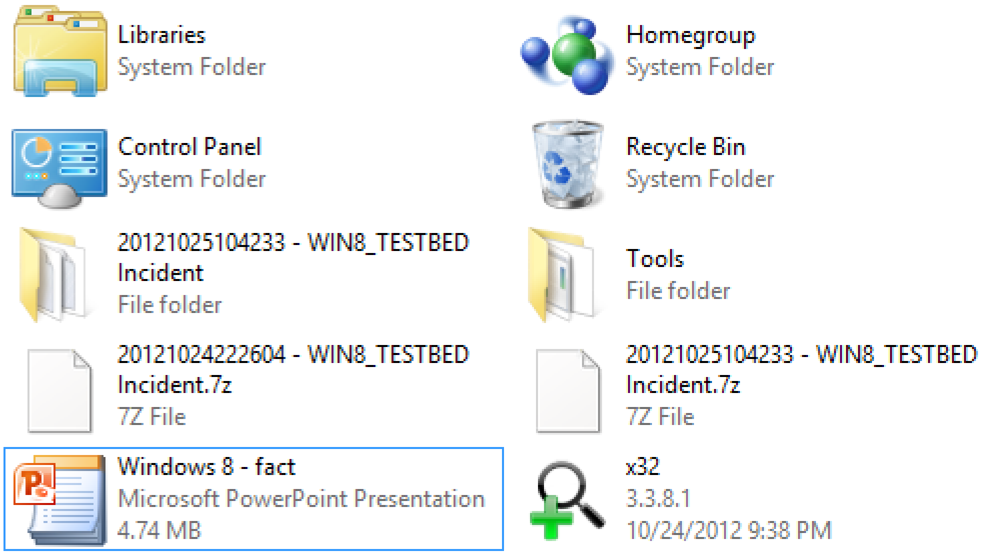


Figure 43

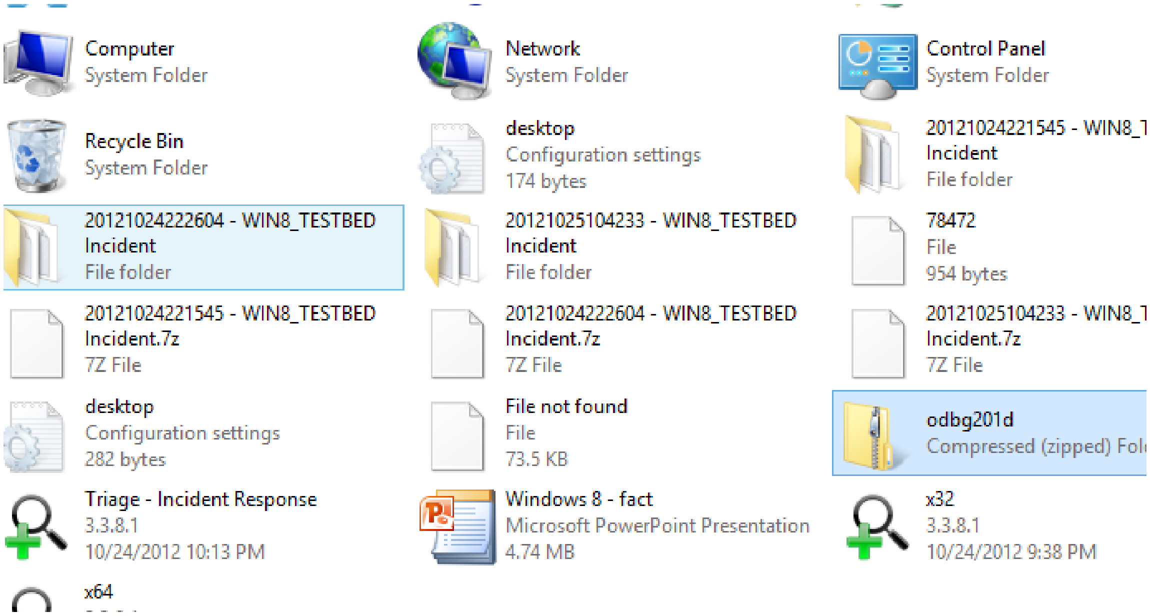
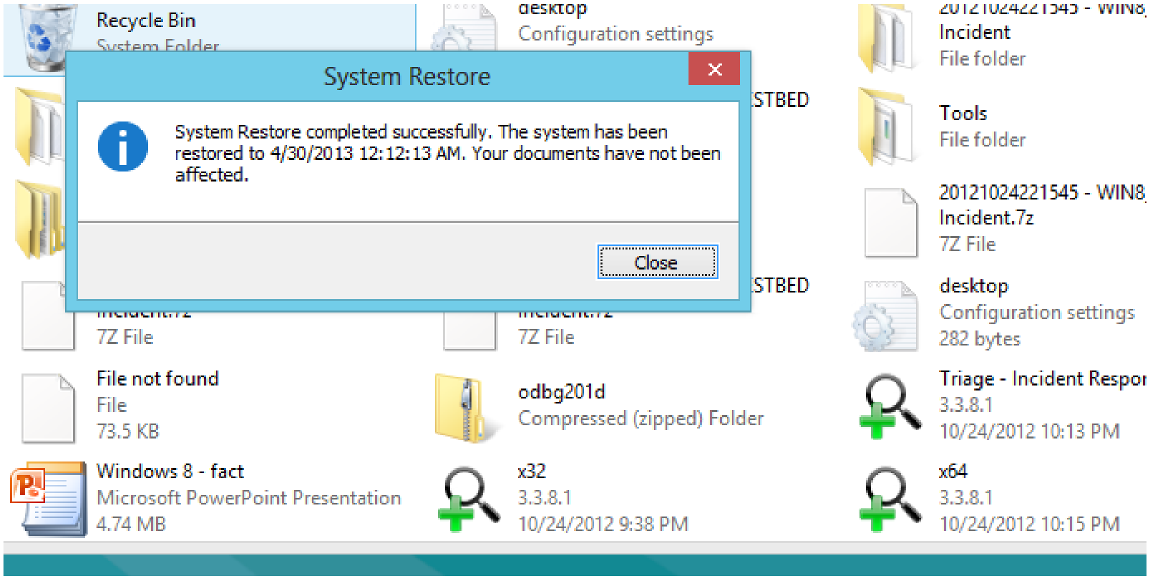


Figure 44

Looking at the desktop I was surprised to see that the odbg201d zipped folder was still on my desktop, but my Tools directory was restored as well.



System Restore completed successfully. The system has been restored to 4/30/2013 12:12:13 AM. Your documents have not been affected.

Figure 45

This was unexpected behavior on why I was able to recover directories and files within those directories but I was not able to recover pictures. More testing needs to be done on this to determine what other items are not recovered from a system restore.

## Baseline Artifacts

Table 8

| Artifact Name | When Installed | Present After Restore |
|---|---|---|
| **DSC_0478** | Before Restore Point | Yes |
| **DSC_0481** | Before Restore Point | No |
| **DSC_0497** | Before Restore Point | Yes |

| | | |
|---|---|---|
| Tools Directory | Before Restore Point | Yes |
| DSC_0499 | After Restore Point | Yes |
| Odbg201d.zip | After Restore Point | Yes |
| | | |

Restore Points Findings

Table 9

| Impacted User Created Artifacts | Impacted Applications |
|---|---|
| Impact depends on the artifact. When testing Pictures were not impacted by their state at the time of the Restore Point, but a directory created before a Restore point was recreated and a zip file created after a restore point was never removed. | If artifacts were present on the machine prior to the creation of the restore point they will be retained. |

## System Refresh Points Artifacts

### Default Refresh Initial Configuration

For my initial configuration for testing the default system refresh point I created a snapshot with my virtual machine for a quick restore of my test environment. I made sure that all of my artifacts were present on the machine before continuing with the system refresh. Based on previous research my assumption was that I would lose all installed applications unless they were installed from the windows store, and that my user created content would be safely transferred over.

Default Refresh Findings

| Impacted User Created Artifacts | Impacted Applications |
|---|---|
| User created documents, images, and files will be retained and copied over from a refresh. | Only the applications that were installed through the Windows Store will be retained. |

Table 10

Custom Refresh Initial Configuration

For my initial configuration for testing the custom system refresh point I created a snapshot with my virtual machine for a quick restore of my test environment. I made sure that most of my artifacts were present on the machine before continuing with the system refresh. After I had created the custom system refresh point I installed the last few applications. Based on previous research my assumption was that I would lose my installed applications unless they were installed from the windows store or prior to the creation of the custom restore point, and that my user created content would be safely transferred over.

Custom Refresh Findings

| Impacted User Created Artifacts | Impacted Applications |
|---|---|
| User created documents, images, | Applications that were installed through the |

| | |
|---|---|
| files will be retained and copied over from a refresh. | Windows Store will be retained. Applications that were installed at the time the custom refresh point was created will be retained. |

Table 11

**System Reset Artifacts**

Quick Reset Initial Configuration

For my initial configuration for testing the quick system reset I created a snapshot with my virtual machine for a quick restore of my test environment. I made sure that all of my artifacts were present on the machine before continuing with the system refresh. Based on previous research my assumption was that I would lose all of my system and user data from the machine, although I should be able to recover the majority of this data from the unallocated space.

Quick Reset Findings

| Impacted User Created Artifacts | Impacted Applications |
|---|---|
| User created artifacts will be available in the unallocated space. | Previously installed applications will leave artifacts in the unallocated space. |

Table 12

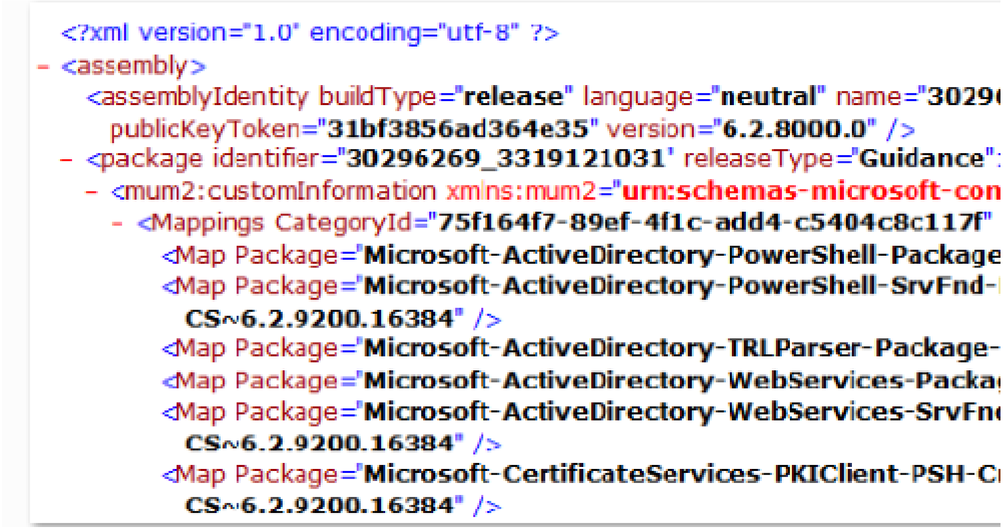
Thorough Reset Initial Configuration

For my initial configuration for testing the thorough system reset I created a snapshot with my virtual machine for a quick restore of my test environment. I made sure that all of my artifacts were present on the machine before continuing with the system refresh. Based on previous research my assumption was that I would lose all of my system and user data from the machine, although I should be able to recover some system created artifacts, user specific artifacts would be impossible to collect.

Thorough Reset Findings

When analyzing the artifact remenants from a Thorough reset there is minimal information that can be extracted. I was able to extract system configuration files but the ability to extract a user created artifact with current software options was impossible.

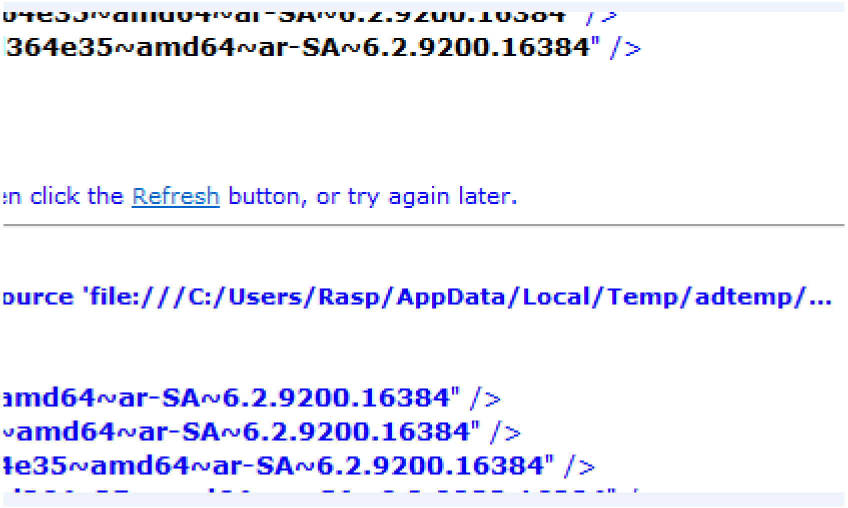
While examining the unallocated space in a hex viewer I came across two artifacts that may provide limited usable data for an analyst. The first data set is a partial hex string that will contain the user file path, with the way that Microsoft overwrites the drive on a Thorough wipe the ability to reliably search for this string is minimal. I was able to find it only by browsing. The second artifact was an XML configuration file. The start of the file looks like the following image.



```
  <?xml version="1.0" encoding="utf-8" ?>
- <assembly>
    <assemblyIdentity buildType="release" language="neutral" name="3029
      publicKeyToken="31bf3856ad364e35" version="6.2.8000.0" />
  - <package identifier="30296269_3319121031" releaseType="Guidance"
    - <mum2:customInformation xmlns:mum2="urn:schemas-microsoft-con
      - <Mappings CategoryId="75f164f7-89ef-4f1c-add4-c5404c8c117f"
          <Map Package="Microsoft-ActiveDirectory-PowerShell-Package
          <Map Package="Microsoft-ActiveDirectory-PowerShell-SrvFnd-
            CS~6.2.9200.16384" />
          <Map Package="Microsoft-ActiveDirectory-TRLParser-Package-
          <Map Package="Microsoft-ActiveDirectory-WebServices-Packa
          <Map Package="Microsoft-ActiveDirectory-WebServices-SrvFn
            CS~6.2.9200.16384" />
          <Map Package="Microsoft-CertificateServices-PKIClient-PSH-C
            CS~6.2.9200.16384" />
```

Figure 46

While there appears to be nothing of interest for the majority of the configuration file near the end of the file I discovered a filepath that listed the username.



```
4e35~amd64~ar-SA~6.2.9200.16384" />
364e35~amd64~ar-SA~6.2.9200.16384" />

n click the Refresh button, or try again later.

ource 'file:///C:/Users/Rasp/AppData/Local/Temp/adtemp/...

amd64~ar-SA~6.2.9200.16384" />
~amd64~ar-SA~6.2.9200.16384" />
4e35~amd64~ar-SA~6.2.9200.16384" />
```

Figure 47

While this may not be usable in every instance it may show an outlier account that was previously on the machine prior to a Thorough rest. Beyond those two artifacts I have been unable to successfully carve any user identifiable data from the disk.

Table 13

| Impacted User Created Artifacts | Impacted Applications |
|---|---|
| There is limited availability of user created artifacts. There is a high inability to extract user data based on file carving due to the way that the drive is randomly written. On the drive there may be references to previous user accounts on the system in XML files or when viewing the drive in a hex viewer.<br>When looking in a hex viewer the time digging for a userID maybe very prohibitive in time efficiency. | Previously installed applications might be extracted from unallocated space. Analysts are able to extract OS related artifacts from unallocated space. |

# CHAPTER 6

# SUMMARY AND FUTURE RESEARCH

## Summary

There is a lot of interesting challenges facing forensic investigation with the new Windows 8 operating system. These challenges include new registry keys, file history options, the increased usage of jump lists, the ability for applications to sync to remote cloud based storage, and the ease of resetting the operating system to a factory restore with little effort or time.

One of the primary challenges within Windows 8 will be how data is retained across the various recovery methods. What artifacts will be recoverable between user created and system created behaviors and how are they impacted with the various recovery options available.

A second challenge to Windows 8 forensics is how the new operating system will tie into the Windows Live cloud capabilities. This can provide a wealth of information on what machines a user access, emails, and traffic pattern across the different synced devices.

A third challenge to Windows 8 Forensics is how will the new Metro Applications and Windows Store interact with the operating system and what artifacts will they retain?

A fourth challenge will be is how the recent release of Windows 8.1 impact the previous research that has been conducted in this paper. How will the native syncing to the Microsoft Skydrive impact data analysis and tracking? How will the new Internet Explorer 11 impact browser artifacts?

While this paper doesn't cover all of the forensic concerns with Windows 8 it does help lay a foundation of knowledge on how forensic important artifacts will be impacted on the system after one of the numerous recovery methods are utilized on the machine.

While all of these challenges can be mitigated from research and analysis in the field, some will provide more of a challenge then others to mitigate. There have been research that has looked into the workings of Jump Lists, and registry keys from previous versions of Windows that will help mitigate these challenges. The challenges from the file history, cloud syncing for applications, and artifacts left over from a reset or a refresh of the operating system will require further research from those in the field.

## Future Research

Future research in this topic will include analysis of the algorithm that is used to randomize the wipe pattern in a thorough reset. Since we are able to recover system created artifacts and residuals of user created artifacts there should be a distinguishable pattern to extract larger sampling of user created data.

A second aspect for future research will be to understand how a Bitlocker encrypted drive is impacted through the various recovery methods. This will include analyzing the thorough reset artifacts will retain anything usable after a refresh. While the drive is encrypted with BitLocker will that actually impact the extraction of data if an analyst is able to decrypt the drive.

## REFERENCES

4n6k, D. P. (2011, 09 01). *jump list appids*. Retrieved 12 3, 2011, from Forensic Artifacts: http://forensicartifacts.com/2011/09/jump-list-appids/

Carvey, H. (2011, 08 8). *Links and Updates*. Retrieved 12 2, 2011, from windowsir: http://windowsir.blogspot.com/2011/08/links-and-updates.html

Carvey, H. (2011). Windows Registry Forensics. In H. Carvey, *Windows Registry Forensics* (pp. 15-21). Syngress.

Carvey, H. (n.d.). *Regripper*. Retrieved 12 2, 2012, from Regripper: http://regripper.wordpress.com/regripper/

Case, A. (n.d.). *Registry Decoder*. Retrieved 12 2, 2011, from http://dfsforensics.blogspot.com/2011/09/announcnig-registry-decoder.html

Corporation, M. (n.d.). *KB 256986*. Retrieved December 2, 2011, from Microsoft: http://support.microsoft.com/kb/256986

Microsoft. (n.d.). *KB 256986*. Retrieved 12 2, 2012, from Microsoft: http://support.microsoft.com/kb/256986

Microsoft. (n.d.). *Using Jump Lists to open programs and items*. Retrieved 12 2, 2011, from microsoft: http://windows.microsoft.com/en-US/windows7/Using-Jump-Lists-to-open-programs-and-items

Mitec. (n.d.). *Structured Storage*. Retrieved 12 2, 2011, from mitec: http://www.mitec.cz/ssv.html

Regshot. (n.d.). *Regshot*. Retrieved 12 2, 2012, from http://sourceforge.net/projects/regshot/

Serban, A. (n.d.). *How to use the Windows 8 File History*. Retrieved 12 2, 2011, from itproportal: http://www.itproportal.com/2011/11/28/how-use-windows-8-file-history-system/

Wegner, W. (2011, 09 16). *SAC-861T*. Retrieved 11 01, 2011, from Channel9.msdn.com: http://channel9.msdn.com/Events/BUILD/BUILD2011/SAC-861T

Wikipedia. (n.d.). *Windows Registry*. Retrieved 12 3, 2011, from http://en.wikipedia.org/wiki/Windows_Registry

Woan, M. (n.d.). Retrieved 12 2, 2011, from woanware.co.uk: http://www.woanware.co.uk/?page_id=266

# APPENDIX

## RECOVERY ARTIFACTS QUICK REFERENCE

### What Artifacts Are Impacted

Table 14

| Recovery Type | User Created | Application |
|---|---|---|
| Restore Point | If artifacts were present on the machine prior to the creation of the restore point. | If artifacts were present on the machine prior to the creation of the restore point. |
| Default Refresh | User created documents, images, and files will be retained and copied over from a refresh. | Only the applications that were installed through the Windows Store will be retained. |
| Custom Refresh | User created documents, images, files will be retained and copied over from a refresh. | Applications that were installed through the Windows Store will be retained. Applications that were installed at the time the custom refresh point was |

| | | |
|---|---|---|
| | | created will be retained. |
| Quick Reset | User created artifacts will be available in the unallocated space. | Previously installed applications will leave artifacts in the unallocated space. |
| Thorough Reset | There is limited availability of user created artifacts. There is a high inability to extract user data based on file carving due to the way that the drive is randomly written. | Previously installed applications might be extracted from unallocated space. Analysts are able to extract OS related artifacts from unallocated space. |