Windows Surface RT Tablet Forensics

Asif Iqbal, Hanan AlObaidli, Andrew Marrington and Andy Jones

Outline

- Introduction
  - Problem statement
  - Research Question
- Literature Review
- Methodology
- Acquisition
- Analysis
- Conclusion

Introduction

- According to (Marturana, Me, Berte and Tacconi) (2011) and NIST (National Institute of Standards and Technology) it's more likely that law enforcement will encounter a suspect with a mobile device in his/her possession than a PC or laptop.
- Proximity and comparatively small capacity make these devices attractive "targets" for examination

Introduction: Problem Statement

- Tablets and smart phones are heterogenous when compared to PCs
- Many manufacturers, with significant differences in hardware and software
- As opposed to "general purpose" PCs, OS often locked down and usage is constrained
- How can we acquire and examine images from the plethora or different devices?

Introduction: Research Question

- The aim of this research is to forensically investigate a Windows RT tablet (Surface RT)
- The Research Questions that this research attempts to answer are as follow:
  - Question 1: How do we perform physical acquisition of the Windows RT tablet?
  - Question 2: What are the difficulties faced by the forensic investigator when acquiring an image of the Windows RT tablet?
  - Question 3: What are the difficulties faced by the forensic investigator when analyzing an image of the Windows RT tablet?
  - Question 4: What are the artifacts that can be found in the Windows RT tablet?

Research Methodology

- In order to answer these questions the test and examination procedure will be derived from the Computer Forensics Tool Testing program guidelines established by the National Institute of Standards and Technology (NIST)

Literature Review

- With every new device such as Windows Surface RT there is some research needed into acquisition and examination techniques
- Different connectors
- Different secondary storage configurations
- Different operating systems with different security mechanisms which need to be bypassed for forensics to be possible

Literature Review

- In 2012 Amanda Thomson attempted the study of the forensic artifacts left on Windows 8 Consumer Preview 32-bit Edition in order to create a guide book for forensic examiners
  - Using FTK Imager v3.0.1 she imaged the VM where Windows 8 is installed
  - She Utilized EnCase Forensic v6.17 for examination and analysis

Literature Review

- An example of the artifacts found by Thomson were
  - Information about Apps that are displayed on the Metro interface
  - web cache and cookies specific to Metro Apps
- Also in 2012 Ethan Fleisher investigated the effect of Reset and Refresh function which is a feature in Windows 8 with regard to forensic evidence

Literature Review

- In 2012 Schaefer, Höfken and Schuba discussed the acquisition and analysis of a Windows Phone 7 device
  - Their work explains the main characteristics of the platform
  - The problems that forensic investigators face
  - Methods to circumvent those problems
  - A set of tools to get data from the phone.

Literature Review

- Data that can be acquired from the phone include:
  - The file system
  - The registry
  - Active tasks
  - Based on the file system, further information like SMSs, Emails and Facebook data can be extracted

Literature Review

- In 2013 Kaart, Klaver and van Baar stated that Windows 8-based mobile devices might still have some of the files located on Windows Phone 7 operating system.
  - Hence we studied the artifacts found with regard to Windows Phone 7

Literature Review

- Kaart et. al. have reverse-engineered significant parts of the EDB volume format and extensively analyzed the pim.vol file
  - That contains information related to contacts, appointments, call history, speed-dial settings and tasks.
- They have also implemented a parser for the EDB volume format structure
  - They compared their results to the traditional approach using an emulator and the API provided by the Windows CE operating system.
  - The parser was able to recover additional databases, additional properties per record and unallocated records

Literature Review

- In January 2013 according to Computer Fraud & Security Journal, a researcher known as CL Rokr (aka ‘clrokr’), claims to have found a way to bypass the code integrity checking in Windows RT which allows users to run unsigned code on Surface tablets and other devices, effectively jailbreaking the platform
- Microsoft has now patched this vulnerability (but what about the next vulnerability, and the next one, etc)

Methodology: Test environment and requirements

Following is a list of devices and analysis tools used in this investigation:

- Microsoft Surface RT device (32 GB)
- Custom data acquisition tool (compiled for Windows RT on ARM platform)
- Ophcrack (to decrypt passwords)
- Cafae (to parse registry)
- Vim Text Editor
- Yaru
- Mitec Structured Storage Viewer (to view OLE)
- IrfanView

Methodology: Test environment and requirements

- A lab computer was setup to run all the analysis tools listed above.
- All system applications running on the Surface RT device were updated after the user account setup.
- Configuration was not modified during the setup of the device.

Methodology: Test Procedures

- A fictional scenario was developed for the purpose of conducting the investigation.
- A Microsoft account was created using fictional information
  - This was done since installing applications, using Skydrive (now OneDrive) features and sync features require a Microsoft Account

Methodology: Test Procedures

Input	Value
First Name	John
Last Name	Doe
Email Address	m80003052@zu.ac.ae
Password	Testing1

Table 1: contains the information used for creation of the account

Methodology: Test Procedures

- A set of common activities was conducted, such as
  - browsing Internet,
  - using Camera,
  - and uploading images to Skydrive (now OneDrive)

- A base image was acquired after creation of user account and analyzed for system related artifacts.
  - It was then set as a Base image for further investigation.

Methodology: Test Procedures

- After each activity a new image was acquired from the device and compared to the base image
- The changes were then analyzed and reported

Acquisition

- A kernel level vulnerability in Windows Driver Signature Enforcement has existed in Microsoft Windows for sometime now.
  - Since Windows RT is a port of Windows on ARM platform, this vulnerability was also ported.

Acquisition

- The minimum signing level determines how good an executable's signature is on a scale like this:
  - Unsigned(0)
  - Authenticode(4)
  - Microsoft(8)
  - Windows(12)

Acquisition

- The default value on x86 machines 0
  - Where you can run anything you like on your computer
- On ARM machines, the default is set to 8
  - This is not a user setting, but a hard-coded global value in the kernel itself
  - It cannot be changed permanently on devices with UEFI's Secure Boot enabled.
  - It can, however, be changed in memory

Acquisition

- By adapting the technique mentioned , The value of signing level is set to Unsigned(0). This allows execution of unsigned applications.

- It was decided that Windows Volume Shadow Copy service would be used to acquire an image of locked files.
  - For this purpose a C++ program was written to use VSS APIs to copy all of disk's content.

Usage Scenario

Application	Activity
Internet Explorer (Immersive)	• Browsed to zu.ac.ae • Navigated to link “About ZU” (http://www.zu.ac.ae/main/en/explore_zu/index.aspx) • Opened a new tab and navigated to slashdot.org • Favorite the page slashdot.org
Internet Explorer (Desktop)	• Browsed to google.com • Browsed to microsoft.com • Add microsoft.com to favorites
Camera	• Took 2 different photos and a video clip. • Uploaded the photos and video clips to Skydrive

Usage Scenario

Application	Activity
Mail	• Setup email access to m80003052@zu.ac.ae • Accessed an email message • Composed and sent an email to asif@babariqbal.com • Subject: My Surface RT, Message: Hello from surface RT
Skydrive	• Created and uploaded a document titled “Document2” • Downloaded and accessed “Document1.doc” previously uploaded to Skydrive from a different PC
Photos	• Downloaded pic1, pic2 and pic3 previously uploaded to Skydrive • Viewed pic1, pic2, pic3

Usage Scenario

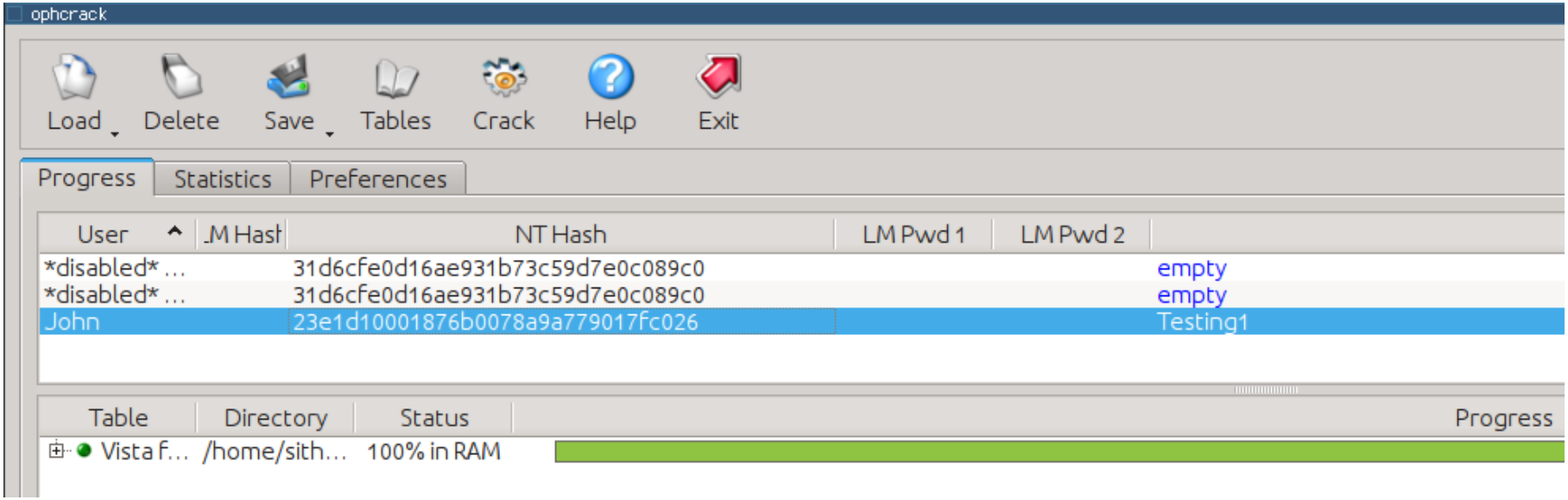
Application	Activity
Music	• Listened to a radio station, Artist “coldplay” • Note: During the process a Microsoft Xbox account was automatically created.
Calendar	• Added a new calendar event
Weather	• Added a new weather city “Seoul, South Korea”
Store	• Searched for and installed Skype app from the Windows Store
Bing	• Searched for keywords “security”, “forensics” and “digital forensics”
Skype	• Added contact “asifiqbal.ai” • Initiated a text chat session with “asifiqbal.ai” • initiated a video chat with “asifiqbal.ai”

Analysis: Non-Metro application artifacts recovered

- The user's online and offline account passwords were successfully recovered
  - The information from user's online Microsoft account is used in local Windows user account
  - The local account also shares the password of online account of the user
  - Since Windows stores password hashes of the local account in SAM file located in %WINDOWS%/System32/config directory, the content of this directory were investigated.

Analysis: Non-Metro application artifacts recovered

- In order to decode SAM file found on the device Ophcrack tool was used
  - The password of user's online and offline account was successfully recovered



Analysis: Non-Metro application artifacts recovered

- All the photos and videos taken on device itself were stored in
  - %userprofile%\Pictures\Camera Roll\ directory
  - Photos were stored in JPEG format while the videos were stored in MP4 file format.
  - Variable %userprofile% = C:\Users\%User Name%\

Analysis of Metro application artifacts

- Metro applications do not have direct access to file system, all their content are stored under
  - %userprofile%\AppData\Local\Packages\%Application Name%_%id%.
  - Table 3 details the structure of Metro application data directory.

Analysis of Metro application artifacts:

Table 3

Directory	Content
-AC	
--INetCache	Cache file are stored in randomly named sub-directories. Cache files retain their original filename and extension.
--INetCookies	Cookies are stored in this Directory with cookies for each website stored in a single textfile with random name
--	
-LocalState	This directory can be directly accessed by the application and can be used to permanently store any type to data.
-Settings	Application settings are stored in this directory as Registry Hive (regf) files. These files can be analyzed using yaru

Analysis of Metro application artifacts: Camera Application

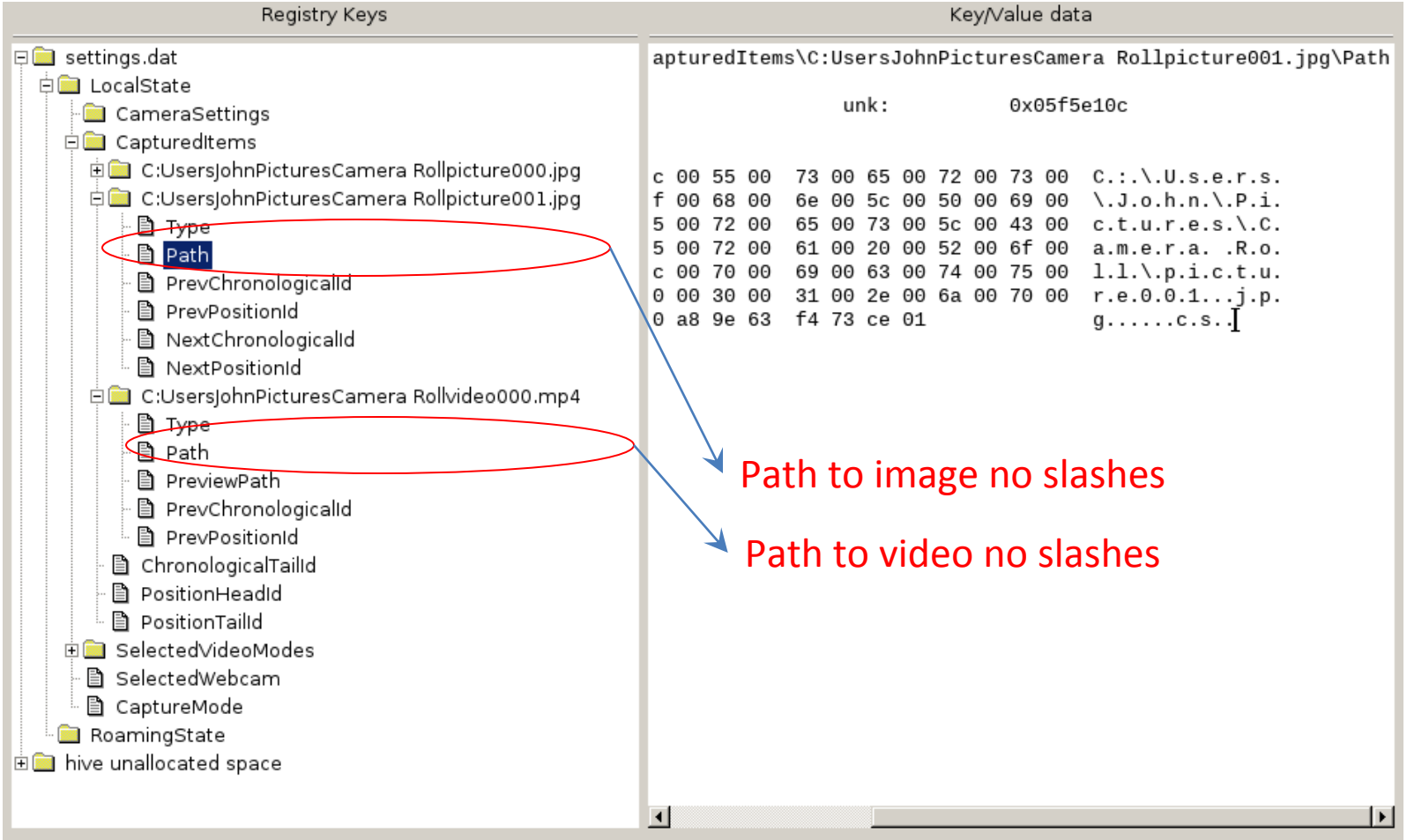
- Artifacts left by Camera metro application were found in:
  - directory “Microsoft.Camera_8wekyb3d8bbwe” under Packages directory
- Registry hive “settings.dat” was found under Settings directory
  - Table 4 details the data found inside the settings registry hive.

Analysis of Metro app artifacts: Table 4

Tree	Content
-Local State	
--Captured Items	A list of all items captured with the Camera application.
---%Path to image no slashes%	A single captured item
----Path	Full path to the image file, images are stored in jpg format under %userprofile%\Pictures\Camera Roll\.
---%Path to video no slashes%	
----Path	Full path to video file, videos are stored in mp4 file format under %userprofile%\Pictures\Camera Roll\.
----PreviewPath	Path to preview thumbnail to one frame of the captured video. These images are stored under %Application Package Directory%\Local State\CameraSettings\Preview Bitmaps\{random filename} in JPEG file format.

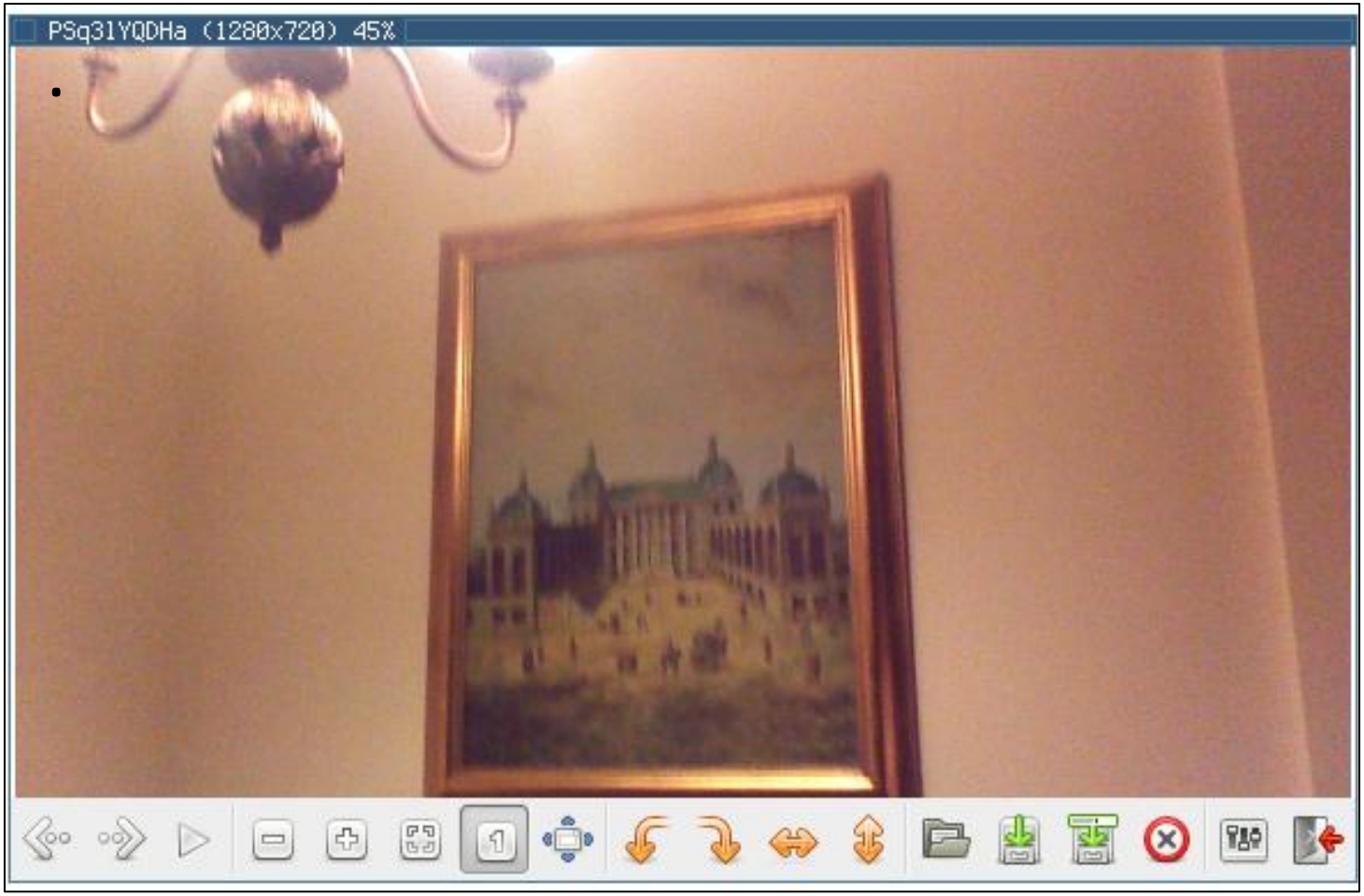
Analysis of Metro application artifacts:

Camera Application



setting.db Registry hive file being analyzed in yaru

Analysis of Metro application artifacts:
Camera Application



Analysis of Metro application artifacts:
Photos Application

- Photos application display not only the local photos stored in the Pictures directory in the user's home directory but also online photos stored in the user's Skydrive.
- All the online Skydrive photos are cached to expected INetCache directory under the package directory
  - (microsoft.windowsphotos_8wekyb3d8bbwe).

Analysis of Metro application artifacts:
Skydrive application

- Data related to files uploaded to skydrive using the application was found in the directory
  - LocalState\bt\uploads\%User ID%\
  - under package directory microsoft.microsoftskydrive_8wekyb3d8bbwe
  - It is stored in xml files with GUID as file-name

Analysis of Metro application artifacts:
Skydrive application

- Transaction records for Skydrive (OneDrive):
  - transferGUID: Unique ID of the transaction, this id is also used in filename.
  - UserCid: Unique ID of user's Microsoft account, this is previouly mentioned as variable %User ID%.
  - FileName: Name of file on local system, this will also be used as name of the file on Skydrive servers.
  - destinationGroup: Name of the directory files is to be sent to on Skydrive server.

Analysis of Metro application artifacts:

Skydrive application


```
{"transferGuid":"32fafdc5-d785-40a7-b439-8b119a2e9490",
 "userCid":"8545a7d827ea818e","initiationSource":1,
 "fileName":"video000.mp4",
 "eTag":null,
 "parentResourceId":"8545A7D827EA818E!130",
 "locationItemKey":"id=8545a7d827ea818e%21130&cid=8545a7d827ea818e&group=0&qt=",
 "nameConflictResolutionType":2,
 "destinationGroupName":"Pictures"}
```

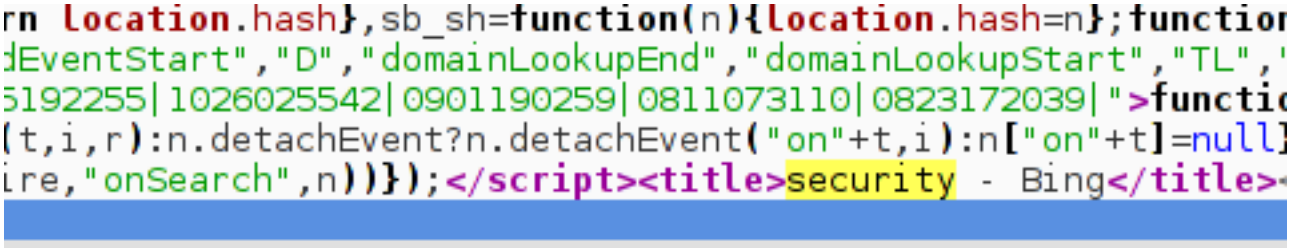

File recovered indicating upload of video000.mp4 captured to Pictures directory on Skydrive

Analysis of Metro application artifacts:
Bing Search

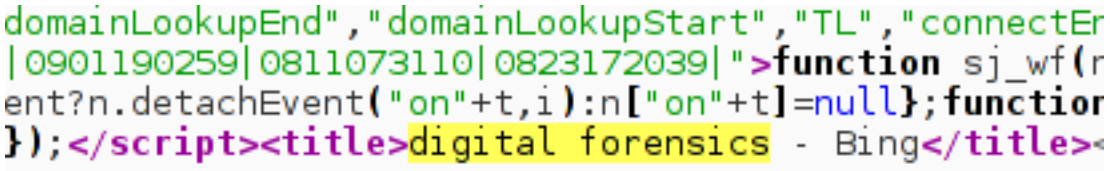
- Bing search artifacts were recovered from randomly named subdirectories of INetCache directory in the Bing package directory (Microsoft.Bing_8wekyb3d8bbwe)
- The files recovered were cache files of page search.htm (generated when a new search term is entered).
- The cache directories also held several image files that were displayed in result of the search.

Analysis of Metro application artifacts:

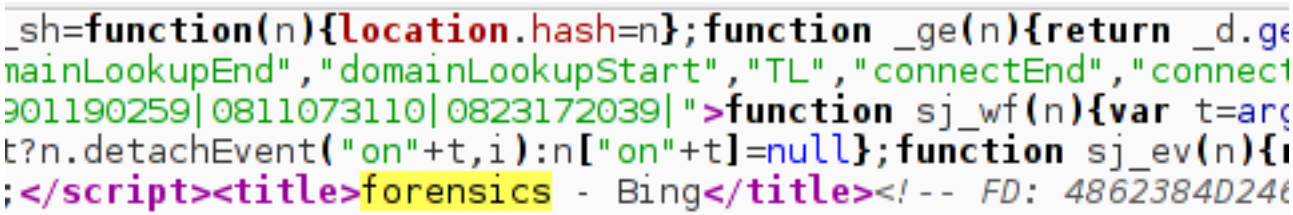
Bing Search



Excerpt from cached file search.htm indicating search for term “security”



Excerpt from cached file search.htm indicating search for term “digital forensics”



Excerpt from cached file search.htm indicating search for term “forensics”

Analysis of Metro application artifacts:
Internet Explorer artifacts

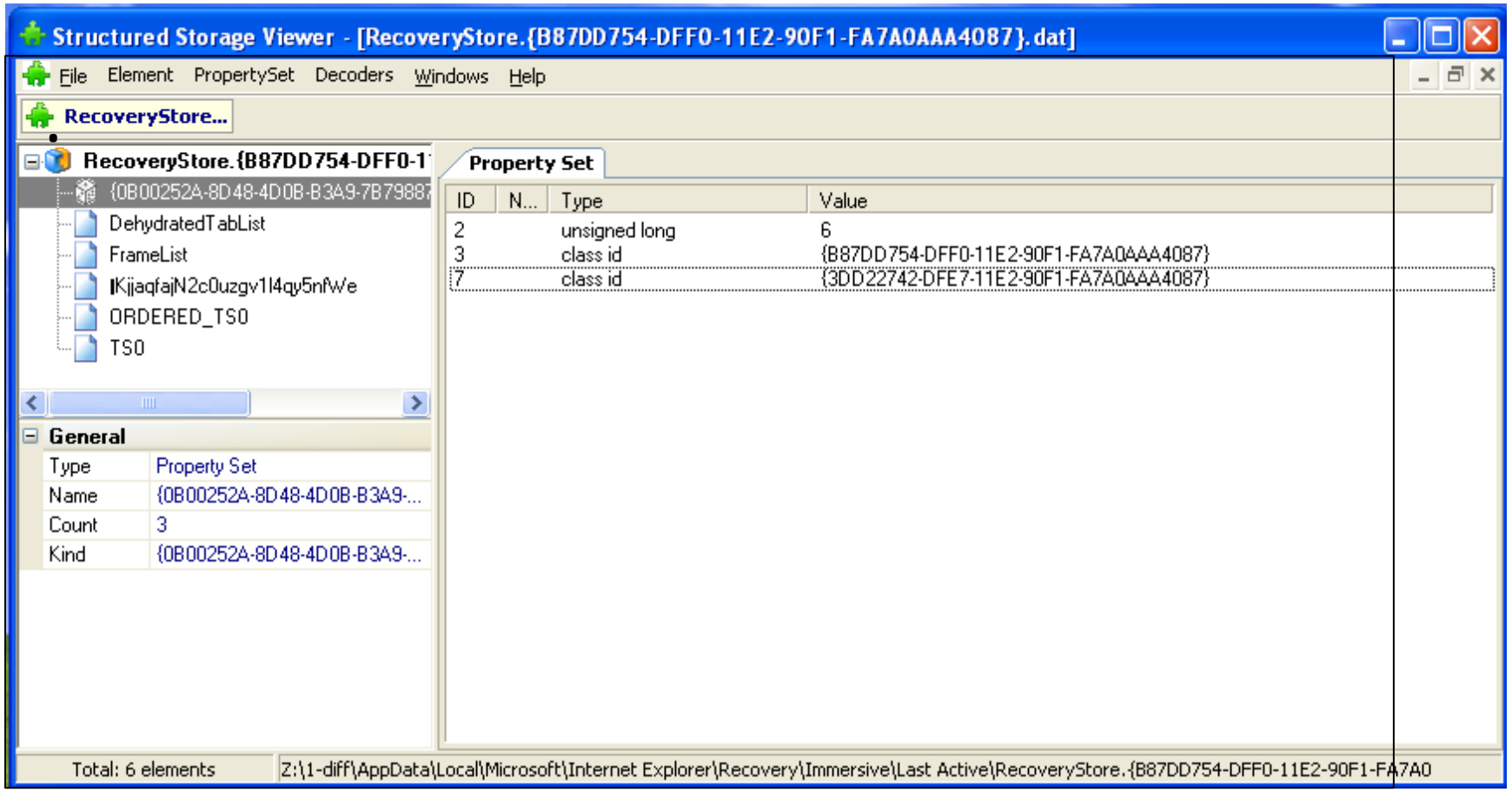
- Internet Explorer browser session artifacts were recovered from %userprofile%\AppData\Local\Microsoft\Internet Explorer\Recovery\.
  - Table 5 details the structure of “Recovery” directory.
- Using the data found in these files a timeline of user's activity in last browsing session was established.

Internet Explorer artifacts

Directory	Comments
-Active	
-Last Active	Data related to tabs open in last session of Internet Explorer Desktop version
-Immersive	
--Active	
--Last Active	This directory holds several files stored with filename %GUID%.dat. All of these files represent an open tab in last browsing session. Apart from these files there is a RecoveryState.{%GUID%}.dat file which serves to hold the GUID variable used in each of the files mentioned above. The files use Object Linking and Embedding (OLE) Compound File (CF) format and can be identified by the header D0 CF 11 E0 A1 B1 1A E1. These files can be analyzed by using the free tool “Mitec Structured Storage Viewer”[ref] or by utilizing open-source library libolecf (code.google.com/p/libolecf/)

Analysis of Metro application artifacts:

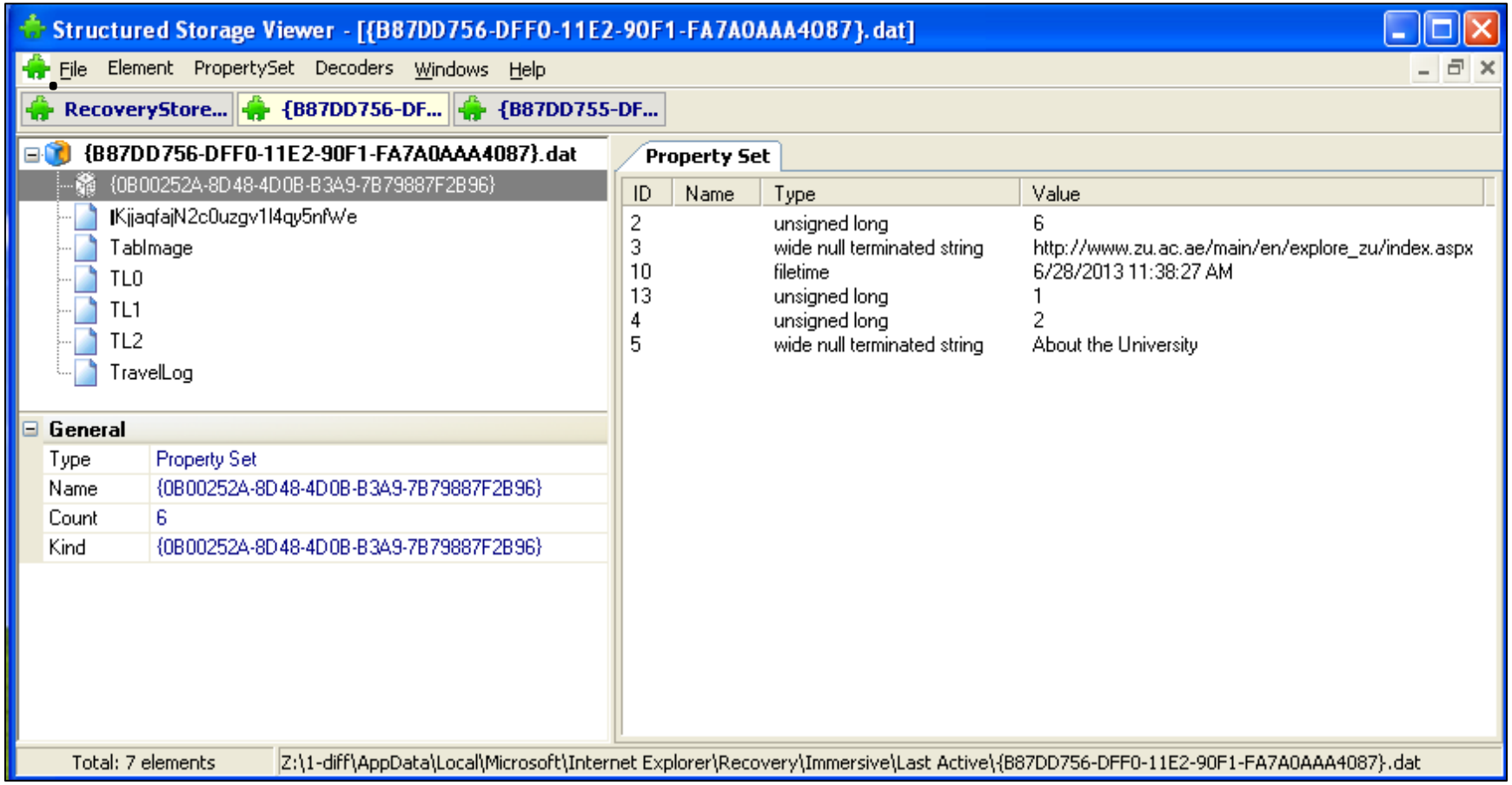
Internet Explorer artifacts



RecoveryState.{%GUID%}.dat file being analyzed in Mitec SSV. Two GUIDs can be seen that indicates that two tabs were open

Analysis of Metro application artifacts:

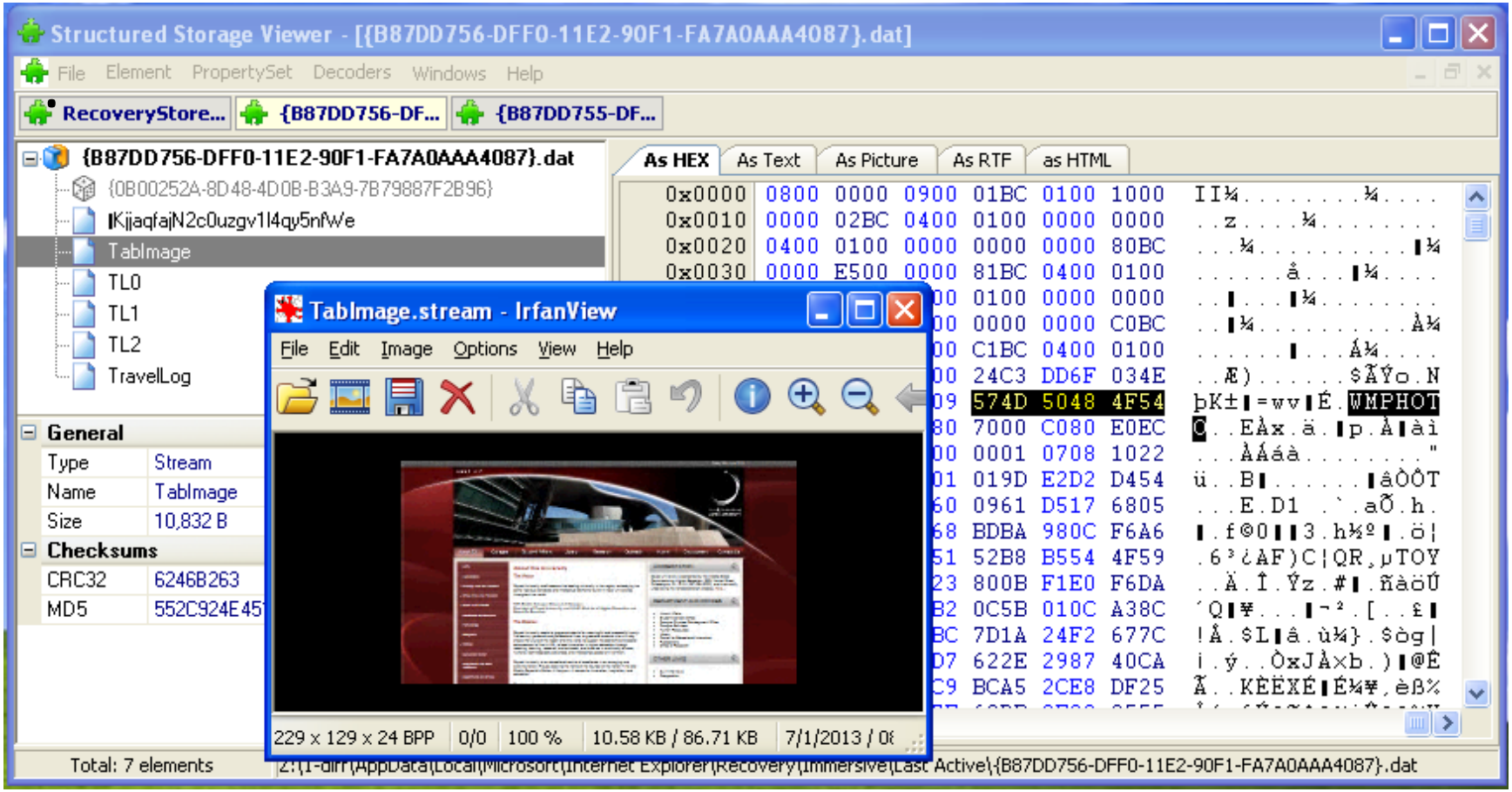
Internet Explorer artifacts



”{B87DD755-DFF0-11E2-90F1-FA7A0AAA4087}.dat file being analyzed in Mitec SSV.

Analysis of Metro application artifacts:

Internet Explorer artifacts



TabImage stream in file {B87DD755-DFF0-11E2-90F1-FA7A0AAA4087}.dat. This file holds information about a single tab in browsing session

Analysis of Metro application artifacts:

Internet Explorer artifacts

="TL2 stream in file {B87DD755-DFF0-11E2-90F1-FA7A0AAA4087}.dat. URL is highlighted."

Analysis of Metro application artifacts:
Internet Explorer artifacts

All the information recovered during this analysis was found to be consistent with the case study conducted in first stage of the experiment.

Conclusion

- Utilizing a vulnerability in Windows RT code we were able to perform the acquisition of the device
- Recovered artifacts can be divided as
  - Non- Metro application artifacts
  - Metro application artifacts

Conclusion

- As in terms of Non- Metro application artifacts we were able to retrieve the password of user's online and offline account
  - This security vulnerability could be useful during a forensic investigation
- In terms of Metro Application artifacts we were able to retrieve information about for example:
  - Camera Application
  - Photos Application
  - Skydrive
  - Bing Search
  - Internet Explorer

Future work

- Plan to develop a tool that would automate the process of acquisition and analysis of a Surface RT device.
- Analysis of 3rd party & social apps on Surface RT device
- To compare the artifacts from Surface RT with other Windows RT devices (in case there are vendor-specific variances).