
RSA MASS TRIAGE
(RMT)

TRIAGE VS MASS TRIAGE

OR HOW I LEARNED TO STOP WORRYING AND LOVE DATA

RSA

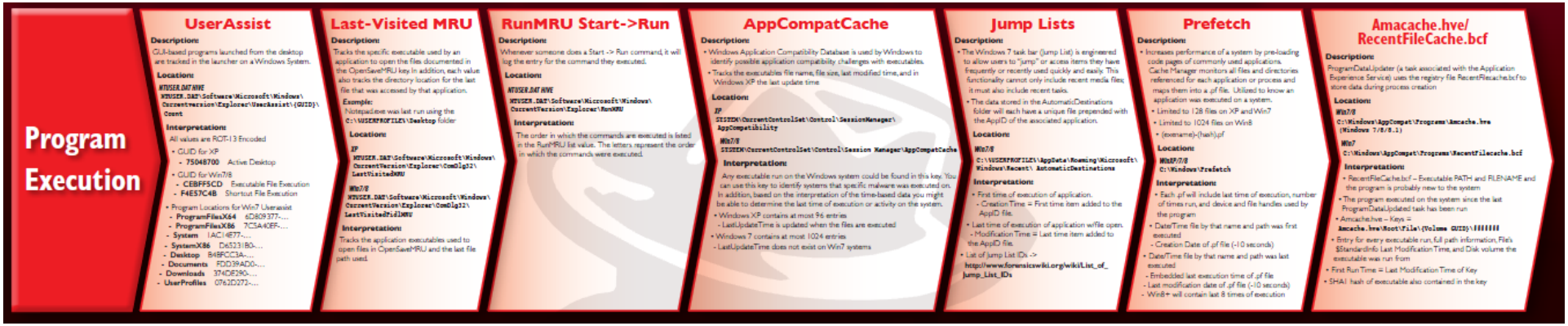
TRIAGE

Do more with less

Rapid situational and historical awareness

Catalyst for follow-on analysis

Initial investigative process



TRADITIONAL IR

- **Indicator leads to System(s)**
- **Review Systems One-at-a-Time**

Single-step analysis and scoping

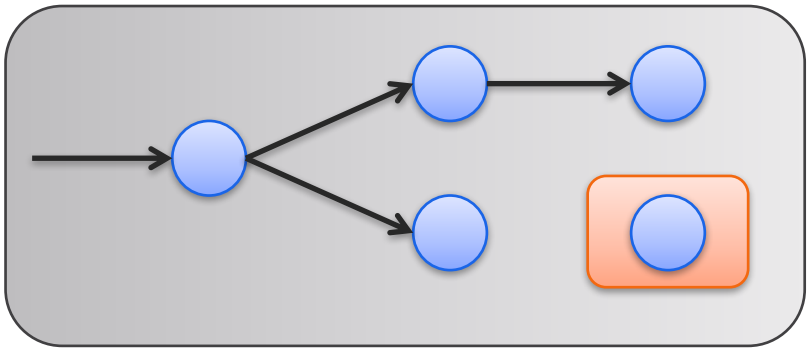
- **Collect Thousand of Artifacts to find Single Indicators**

Manually pull data from various sources, files, folders
Use collection of specialized commercial and free tools to analyze

- **Can Take Weeks and Months to Investigate**

Dead box – Turn off, image, analyze.
Slow process for large compromises

Memory Forensics – Pull Memory image, analyze.
Large Data set. Time / Storage intensive

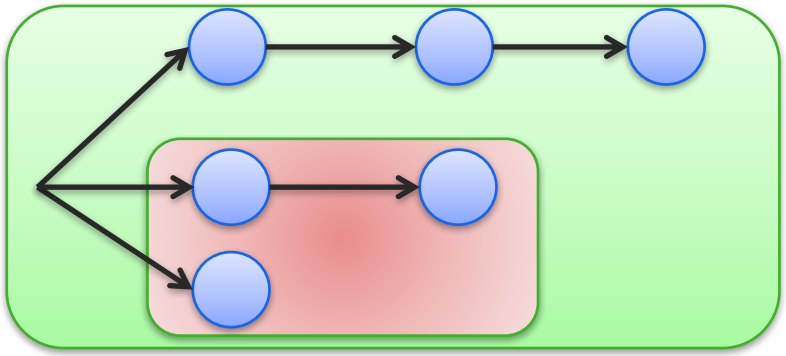


MASS TRIAGE

Breadth-First View of Host Metadata

Recursive Process

- Initiate Follow-On or Concurrent Analysis
- Collect Mass Set of forensic data
- Analyze for outliers and indicators
- Perform remote forensics on flagged systems
- Analyze Mass Set for new Indicators
- Expand scope as needed



RSA

RMT

OR HOW I LEARNED TO STOP WORRYING AND LOVE THE PROCESS

RSA

MASS TRIAGE IN A NUTSHELL

Download triage artifacts from as many systems as possible using NetWitness Endpoint (NWE).

- Priority should be given to servers if resources are limited.

Associate the artifact with the source system.

Processing / Parsing data ensues.

Analyze the results.

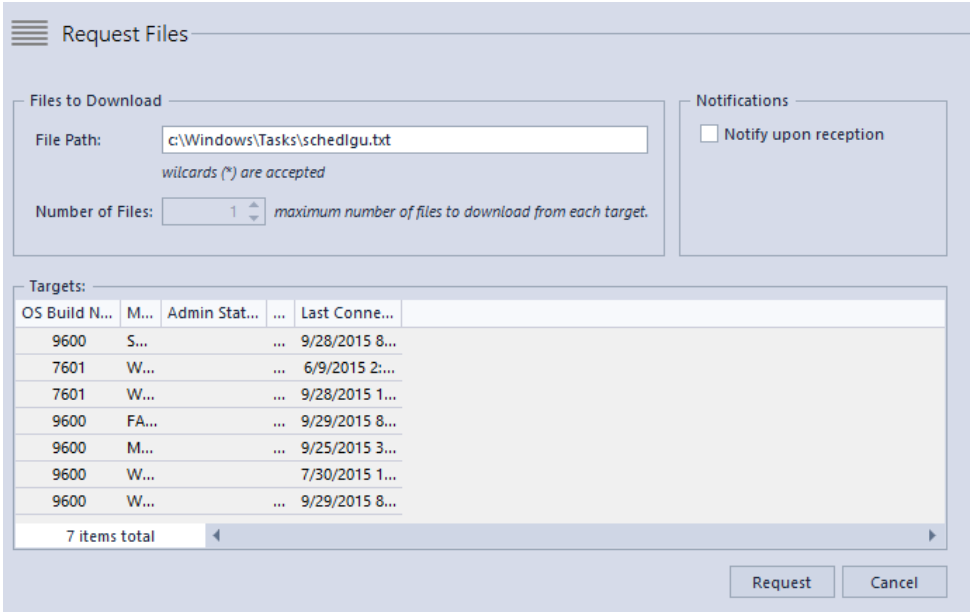
Investigate the findings

Repeat as new data emerges.

MANUAL FILE DOWNLOAD USING NWE UI

NWE can request files from one or many systems

- One of the key features to Mass Triage
- Request files that are forensically significant
- Can use * for wildcard on files only (Cannot wildcard a directory)
- Cannot use Win Environment Variables
  - OK c:\temp*
  - OK c:\badguy*.exe
  - NOT OK C:\users*\ntuser.dat
  - NOT OK %USERPROFILE%\ntuser.dat



RSA

WINDOWS EVIDENCE OF EXECUTION REFERENCE

ShimCache / AppCompat Cache

- C:\Windows\system32\config\SYSTEM

AmCache

- C:\Windows\AppCompat\Programs\Amcache.hve (Win8+)

Recent File Cache

- C:\Windows\AppCompat\Programs\RecentFilecache.bcf (Win7)

Prefetch (Not enabled by default on Servers / Systems with SSDs)

- C:\Windows\Prefetch*.pf

AT Job files

- C:\Windows\System32\Tasks\At*.job (Win7+)
- C:\Windows\Tasks\At*.job (2000, XP, 2003)

User Assist & MuiCache (NTUSER.DAT / USRCLASS.DAT)

- C:\Users\%USERNAME%\NTUSER.DAT (Win7+)
- C:\Users\%USERNAME%\AppData\Local\Microsoft\Windows\USRCLASS.DAT (Win7+)
- C:\Documents and Settings\%USERNAME%\NTUSER.DAT (XP)

ADDITIONAL WINDOWS FORENSIC FILES

Scheduled Tasks

- C:\Windows\Tasks\Schedlgu.txt (2003, Win7+)
- C:\Windows\Schedlgu.txt (XP)Task Scheduler Logs (Vista+)
- C:\Windows\System32\winevt\Logs\Microsoft-Windows-TaskScheduler Operational.evtx

Registry Hives

- C:\Windows\System32\config\SOFTWARE
- C:\Users\<Compromised_User>\NTUSER.DAT

WMI Objects.data

- C:\Windows\System32\wbem\Repository\OBJECTS.DATA
- C:\Windows\System32\wbem\Repository\FS\OBJECTS.DATA

Sticky Keys Backdoors

- c:\windows\system32\sethc.exe
- c:\Windows\SysWOW64\sethc.exe
- c:\windows\system32\utilman.exe
- c:\Windows\SysWOW64\utilman.exe
- c:\windows\system32\osk.exe
- c:\Windows\SysWOW64\osk.exe

Event Logs

- C:\Windows\System32\winevt\security.evtx
- C:\Windows\System32\winevt\system.evtx
- C:\Windows\System32\winevt*powershell*.evtx
- C:\Windows\System32\winevt*TerminalServices*.evtx
- C:\Windows\System32\winevt\Logs*.evtx

ADDITIONAL WINDOWS FORENSIC FILES BITMAP CACHE

Bitmap Cache

- XP - %USERPROFILE%\ Local Settings\Application Data\Microsoft\ Terminal Server Client\Cache\Bcache*.bmc
- Win7+ -%USERPROFILE%\AppData\Local\Microsoft\Terminal Server\Cache\cache*.bin

- http://cert.ssi.gouv.fr/actualite/CERTFR-2016-ACT-017/
- https://github.com/ANSSI-FR/bmc-tools

RSA

ADDITIONAL FILE EXAMPLES AKA SHOTS IN THE DARK

Download all powershell, txt or bat files in a dir :

- c:\windows\system32*.ps1
- c:\windows\system32*.bat
- c:\windows\temp*.txt

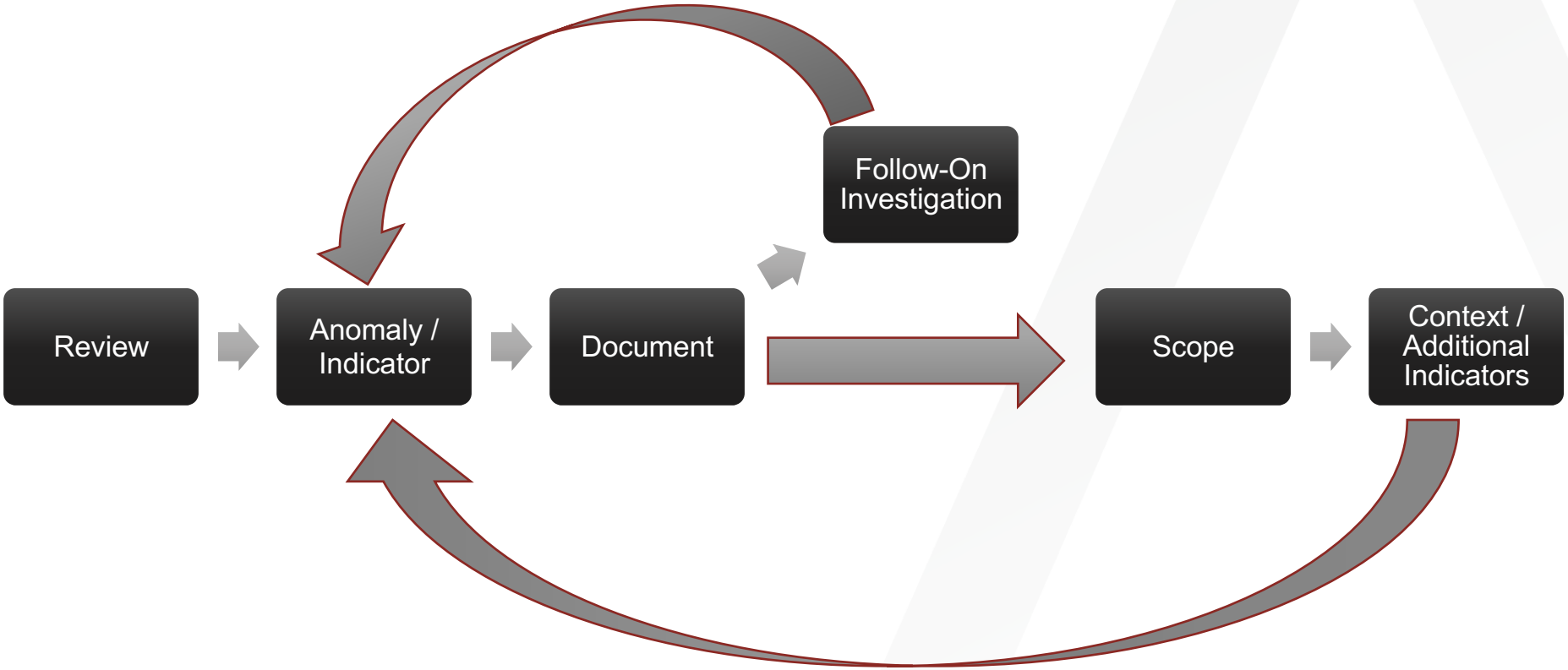
Grab everything in a dir

- c:\temp*

Grab a specific file from all systems

- C:\badguy\bad.exe

MASS TRIAGE ANALYSIS PROCESS



FOUND SOMETHING INTERESTING? WHERE TO GO FROM HERE

From Execution History

- Download MFT of system in question
  - Search for file
  - Timetology
  - Any other interesting files in directory?
    - If so, may want to request all files in directory from systems or all systems
- Request download of file (from system in question)
  - Analyze file
  - If bad, request file from all systems

From Jobs

- Search execution history (Verify job executed)
- MFT / Timetology / Download EXE

Related Artifacts

- Event Logs
- User Context (recently created profiles / ntuser)
- Cross correlate with Tracking Data if possible

WHAT'S THE POINT?

RMT is about historical execution data that is captured from Windows Systems

- Execution history from forensic artifacts
- Shows what was executed on systems **BEFORE** NetWitness Endpoint Agent was installed
- Not as detailed as NWE Tracking data
- **Collected from Disk; Not Memory!**
  - Information is typically stored in two places (Memory & Disk).
  - Memory is only written to disk on clean shutdown/reboot.

RMT

OR HOW I LEARNED TO STOP WORRYING AND LOVE AUTOMATION WITH PYTHON SCRIPTS

RSA

RMT SCRIPT REQUIREMENTS

Install with pip from internet or download

pyodbc>=3.0.10

unicodecsv>=0.14.1>=2.9.6

MarkupSafe>=0.23

jinja2>=2.9.6

tqdm>=4.11.2.

rmt_3_aggregator has specific requirements for each type of input (More on that later)

0_DOWNLOADER

rmt_0_downloader_windows.py

```
usage: rmt_0_downloader_windows.py [-h] -f FILENAME [-m MAXFILES] [-b BATCH]
                                   [-u <user>] [-p <password>]
                                   [-s <hostname or IP>] [-db <database>]
                                   [--debug]

Download files using NWEndpoint from a supplied list of files

optional arguments:
  -h, --help            show this help message and exit
  -f FILENAME, --filename FILENAME
                        Input list of files to download
  -m MAXFILES, --maxfiles MAXFILES
                        Maximum number of files to download when wildcard (*)
                        is used. Default 100
  -b BATCH, --batch BATCH
                        Number of systems to request download from per batch.
                        Default 100
  -u <user>, --user <user>
                        Username for SQL Database. Default: Windows
                        Credentials
  -p <password>, --pass <password>
                        Password for SQL Database. (If user specified with no
                        pass then you will be prompted for the pass)
  -s <hostname or IP>, --server <hostname or IP>
                        Hostname or IP for SQL Server. Default: localhost
  -db <database>, --database <database>
                        ECAT database
  --debug               Enable Debug Messages
```

0_DOWNLOADER

rmt_0_downloader_windows.py

>rmt_0_downloader_windows.py –f <list_of_files_to_download.txt>

Example

```
C:\Windows\system32\config\SYSTEM
C:\Windows\AppCompat\Programs\RecentFilecache.bcf
C:\Windows\AppCompat\Programs\Amcache.hve
C:\Windows\System32\Tasks\At*.job
C:\Windows\Tasks\At*.job
C:\Windows\Prefetch\*.pf
```

0_DOWNLOADER

HELPER FILES

- machine-file-list-SAMPLE.txt
- RMT_AMCACHE_ONLY.txt
- RMT_Artifacts_NO_SYSTEM.txt
- RMT_Artifacts.txt
- RMT_EVTX_Logs_ALL.txt
- RMT_EVTX_Logs_Security.txt
- RMT_EVTX_Logs_SecuritySystem.txt
- RMT_Objects.data.txt
- RMT_StickyKeys.txt
- RMT_SYSTEM_ONLY.txt
- Triage_Artifacts_Notes.txt

HELPER SQL

- RSAIR_CancelDownloadsfromSpecificUser.sql
- RSAIR_FileDownloaded_ntuser.dat-usrclass.dat.sql
- RSAIR_FileDownloaded_ntuser.dat.sql
- RSAIR_FileDownloaded_SYSTEM.sql
- RSAIR_FileDownloaded_usrclass.dat.sql
- RSAIR_MachineCommandStats.sql
- RSAIR_SystemsThat_Did_DownloadNtuser.dat.sql
- RSAIR_SystemsThat_DidNot_DownloadNtuser.dat.sql

0_DOWNLOADER

Other Downloaders / Listers

rmt_0_downloader_machine-file.py

Takes in a file with a set of machinename,filename tuples. Requests downloads of that file only from the system provided.

rmt_0_downloader_ntuser.dat.py

Searching directory paths for all known locations of a User's directory. Then downloads the ntuser.dat for each of those on every system.

rmt_0_downloader_usrclass.dat.py

Searching directory paths for all known locations of a User's directory. Then downloads the usrclass.dat for each of those on every system.

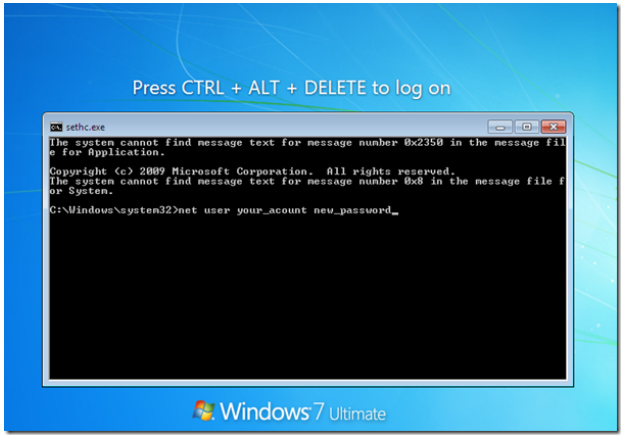
rmt_0_list_ntuser.dat.py

Lists possible ntuser.dat files.

rmt_0_list_usrclass.dat.py
Lists possible usrclass.dat files.

0_DOWNLOADER - SPECIAL TOPIC STICKY KEYS BACKDOORS

- Use rmt_0_downloader_windows.py to download from all systems
- RMT_StickyKeys.txt
  - c:\windows\system32\sethc.exe
  - c:\Windows\SysWOW64\sethc.exe
  - c:\windows\system32\utilman.exe
  - c:\Windows\SysWOW64\utilman.exe
  - c:\windows\system32\osk.exe
  - c:\Windows\SysWOW64\osk.exe



- Compare hashes to cmd.exe & each other. Examine anomalies.

0_DOWNLOADER - WARNING

WARNING

- **You can cause problems with these download scripts**
- Start off small / simple, build your way up as needed.
- Start with System Hive, RecentFileCache and Amcache
- Sticky Keys next
- Consider NWE / Client Environment impact before launching the ntuser / usrclass downloaders
- Don’t forget you can manually download items on a mass basis from the NWE UI.

0_DOWNLOADER – COPYING FILES

COPY FILES FROM NWE FILES DIRECTORY

This is a manual step

- After manual request or script run NWE will issue the download command to agents on check in.
- Offline systems will get the download command only after checking in.
- Wait some time
  - Good time to go hunting in modules or NWP
- Go to the NWE Files directory and copy the files to their own working directory
  - System_ -> rmt\System
  - RecentFileCache_ -> rmt\RecentFileCache
  - Amcache_ -> rmt\Amcache

SEARCHING WINDOWS FOR FILES

Starts With

System.Filename:~<system_

System.Filename:~<amcache_

System.Filename:~<recentfilecache_

System.Filename:~<at_

Contains

System.Filename:~=

Ends with

System.Filename:~>.pf_

System.Filename:~>.job_

RSA

1_DEDUPLICATOR

Optional step to reduce the number of duplicate files (multiple download requests used)

From https://github.com/thorsummoner/duplicate-files

```
usage: DuplicateFiles.py [-h] [-gui] [-root <path>] [-remove]

Finds duplicate files.

optional arguments:
  -h, --help    show this help message and exit
  -gui          Display graphical user interface.
  -root <path>  Dir to search.
  -remove       Delete duplicate files.
```

This script will scan a directory tree looking for duplicate files, it uses a two stage approach of comparing file sizes and then hashes of file contents to find duplicates.

Running Duplicate files.

An example of running this script to just list all the duplicate files would be:

```
python DuplicateFiles.py -root /Users/Daniel/Documents
```

An example of running this script to list and delete all the duplicate files would be:

```
python DuplicateFiles.py -root /Users/Daniel/Documents -remove
```

2_RENAMER

rmt_2_renamer.py

>rmt_2_renamer.py –d <path_to_files>

```
usage: rmt_2_renamer.py [-h] -d <directory> [-u <user>] [-p <password>]
                        [-s <hostname or IP>] [-db <database>]

For each file in the supplied directory, script will get the associated
MachineName from the ECAT DB and insert the MachineName into the filename.

optional arguments:
  -h, --help            show this help message and exit
  -d <directory>, --dir <directory>
                        Directory where files are stored
  -u <user>, --user <user>
                        Username for SQL Database. Default: Windows
                        Credentials
  -p <password>, --pass <password>
                        Password for SQL Database. Default: Windows
                        Credentials
  -s <hostname or IP>, --server <hostname or IP>
                        Hostname or IP for SQL Server. Default: localhost
  -db <database>, --database <database>
                        ECAT database
```

3_AGGREGATORS

rmt_3_executioner.py

Aggregates Execution history from

- **Shimcache**
- **Amcache**
- **RecentFileCache**
- **Prefetch**
- **Jobs**
- **UserAssist**
- **Muicache** *(Beta)

rmt_3_ntuser.dat_RunKeys.py

Parses Run Keys from ntuser.dat

3_AGGREGATORS – RMT_3_EXECUTIONER.PY

rmt_3_executioner.py

```
usage: rmt_3_Executioner.py [-h] [-s SYSTEM] [-a AMCACHE] [-r RFC] [-j JOBS]
                            [-p PREFETCH] [-u USERASSIST] [-m MUICACHE] -o
                            OUTPUT [--debug] [--append]

optional arguments:
  -h, --help            show this help message and exit
  -s SYSTEM, --system SYSTEM
                        SYSTEM Hive Directory
  -a AMCACHE, --amcache AMCACHE
                        Amcache.hve Directory
  -r RFC, --rfc RFC     RecentFileCache.BCF Directory
  -j JOBS, --jobs JOBS  .Job Directory
  -p PREFETCH, --prefetch PREFETCH
                        Prefetch (.PF) Directory
  -u USERASSIST, --userassist USERASSIST
                        UserAssist entries from NTUSER.DAT Directory
  -m MUICACHE, --muicache MUICACHE
                        Muicache entries from NTUSER.DAT/USRCLASS.DAT
                        Directory
  -o OUTPUT, --output OUTPUT
                        Output CSV file
  --debug               Enable Debug
  --append              Append to Output File (instead of overwriting)
```

RSA

3_AGGREGATORS – RMT_3_EXECUTIONER.PY

INPUT

- SYSTEM Hives (Shimcache)
- AmCache
- RecentFileCache.
- Prefetch
- Jobs
- NTUSER (User Assist)

Limited (has some bugs)

- NTUSER (MuiCache)
- USRCLASS (Muicache)

OUTPUT

- Can process each type individually or aggregate all into single CSV
- Each run can append or overwrite output csv

3_AGGREGATORS – RMT_3_EXECUTIONER.PY

Shimcache - Windows Application Compatibility Cache – WinXP+

- https://www.fireeye.com/content/dam/fireeye-www/services/freeware/shimcache-whitepaper.pdf
- https://www.fireeye.com/blog/threat-research/2015/06/caching_out_the_val.html
- Records file execution Windows XP+. (Execution Time - WinXP / Last Modified - Time Vista +)
- Windows Vista+ may record entries for files in a directory that a user interactively browses. E.G., if a directory contains the files “foo.txt” and “bar.exe”, a Windows 7 system may record both.

Recent File Cache - Win7

- Simplified Cache (Replaced by Amcache). No timestamps.
- **http://journeyintoir.blogspot.in/2013/12/revealing-recentfilecachebcf-file.html**

Amcache - Windows Application Experience and Compatibility – Win7+

- Shimcache on steroids. Contains execution history, Last Modified, Sha1 Hash of file. Lots of other details available that are not parsed by out script at the moment
- http://www.swiftforensics.com/2013/12/amcachehve-in-windows-8-goldmine-for.html
- https://binaryforay.blogspot.com/2017/10/amcache-still-rules-everything-around.html?m=1

3_AGGREGATORS – RMT_3_EXECUTIONER.PY

Prefetch - WinXP+ (Typically Disabled on Servers & Systems with SSDs)

- http://www.forensicswiki.org/wiki/Prefetch
- Designed to speed up the application startup process. Augmented / Replaced by SuperFetch
- Prefetch files (.pf) contain:
  - Name of the executable
  - Unicode list of DLLs used by that executable
  - Count of how many times the executable has been run
  - Timestamp indicating the last time the program was run

Jobs

- https://digital-forensics.sans.org/blog/2009/09/16/windows-scheduler-at-job-forensics
- http://www.forensicswiki.org/wiki/Windows_Job_File_Format
- Scheduled Tasks GUI, or 'schtasks', or 'at' command line tools. RMT typically downloads all at*.job files. Though all *.job files can be downloaded.
- NWE parses scheduled tasks / jobs natively, but may not parse all historical .job files on disk.

3_AGGREGATORS – RMT_3_EXECUTIONER.PY

NTUSER.DAT / USRCLASS.DAT

User Assist

userassist.py in the same path as well as its dependencies to be installed -- https://raw.githubusercontent.com/sysforensics/python-regparse/master/plugins/userassist.py

Muicache

muicache.py in the same path -- https://raw.githubusercontent.com/timetology/registry/master/muicache.py

3_AGGREGATORS – RMT_3_EXECUTIONER.PY

rmt_3_aggregator.py Requirements

Shimcache

ShimCacheParser.py in the same path -- https://github.com/mandiant/ShimCacheParser

Registry
python-registry Registry Directory in the same path -- https://github.com/williballenthin/python-registry

Prefetch

prefetch.py in the same path -- https://raw.githubusercontent.com/PoorBillionaire/Windows-Prefetch-Parser/master/windowsprefetch/prefetch.py

Jobs

jobparser.py in the same path -- https://raw.githubusercontent.com/gleeda/misc-scripts/master/misc_python/jobparser.py

Amcache
amcache.py in the same path -- https://raw.githubusercontent.com/williballenthin/python-registry/master/samples/amcache.py

User Assist

userassist.py in the same path as well as its dependencies to be installed -- https://raw.githubusercontent.com/sysforensics/python-regparse/master/plugins/userassist.py

Muicache

muicache.py in the same path -- https://raw.githubusercontent.com/timetology/registry/master/muicache.py

3_AGGREGATORS – RMT_3_NTUSER.DAT_RUNKEYS.PY

rmt_3_ntuser.dat_RunKeys.py

Parses Run Keys from ntuser.dat

- Microsoft\\Windows\\CurrentVersion\\Run
- Microsoft\\Windows\\CurrentVersion\\RunOnce
- Microsoft\\Windows\\CurrentVersion\\RunOnceEx
- Microsoft\\Windows\\CurrentVersion\\Policies\\Explorer\\Run
- Microsoft\\Windows\\CurrentVersion\\RunServicesOnce
- Wow6432Node\\Microsoft\\Windows\\CurrentVersion\\Run
- Wow6432Node\\Microsoft\\Windows\\CurrentVersion\\RunOnce
- Software\\Microsoft\\Windows\\CurrentVersion\\Run
- Software\\Microsoft\\Windows\\CurrentVersion\\RunOnce
- Software\\Microsoft\\Windows\\CurrentVersion\\RunServices
- Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\Explorer\\Run

3_AGGREGATORS – RMT_3_NTUSER.DAT_RUNKEYS.PY

rmt_3_ntuser.dat_RunKeys.py Output

Hostname, Last Write, Key Name, Name, Value

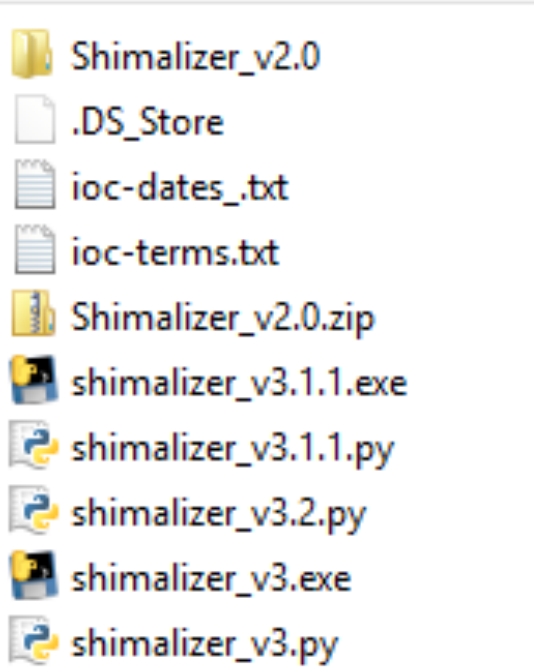
Hostname	Last Write	Key Name	Name	Value
DC01	2/16/2016 20:37	Software\Microsoft\Windows\CurrentVersion\Run	BgMonitor_{79662E04-7C6C-4d	C:\Program Files (x86)\Common Files\Nero\Lib\NMBgMonitor.exe
CORP02	4/2/2017 17:18	Software\Microsoft\Windows\CurrentVersion\Run	Lync	C:\Program Files (x86)\Microsoft Office\Office15\lync.exe /fromrunkey
CORP02	4/2/2017 17:18	Software\Microsoft\Windows\CurrentVersion\Run	OfficeSyncProcess	C:\Program Files (x86)\Microsoft Office\Office14\MSOSYNC.EXE
CORP02	4/2/2017 17:18	Software\Microsoft\Windows\CurrentVersion\Run	rxxmkr	C:\WINDOWS\system32\mshta.exe javascript:pvby0P="B7";dV7=new%20ActiveXObject("WScript.Shell");iGOy21="bzpVQO7";qP7Fd2=dV7.RegRead("HKCU\\software\\wyjmmyq\\qpkika");hVu1DHAU="DziL9z";eval(qP7Fd2);W8o0ozazL="7rD2f";
CORP02	4/2/2017 17:18	Software\Microsoft\Windows\CurrentVersion\Run	dvrox	C:\Users\jadmin\AppData\Local\71ce52\4043fe.lnk
CORP02	4/2/2017 17:18	Software\Microsoft\Windows\CurrentVersion\Run	Adobe Reader Synchronizer	C:\Program Files (x86)\Adobe\Reader 11.0\Reader\AdobeCollabSync.exe
CORP01	6/15/2017 15:08	Software\Microsoft\Windows\CurrentVersion\Run	Sidebar	%ProgramFiles%\Windows Sidebar\Sidebar.exe /autoRun
CORP01	6/15/2017 15:08	Software\Microsoft\Windows\CurrentVersion\Run	Lync	C:\Program Files\Microsoft Office\Office15\lync.exe /fromrunkey
CORP01	6/15/2017 15:08	Software\Microsoft\Windows\CurrentVersion\Run	startup	C:\Users\ladmins\AppData\Roaming\Java\Java.exe

4_GREPALIZER (SHIMALIZER)

Frequency Analysis and Grep-fu

Two versions

- Shimalizer v2.0 (.bat script)
- Shimalizer.py v3.x (python script)



4_GREPALIZER (SHIMALIZER)

Shimalizer v2.0 (.bat scripts)

- shimalizer.bat <path_to_RMT_CSV>

Processing web directories....
Processing reserved names....
Processing Windows folder...
Processing system32 folder...
Processing TEMP folder...
Processing 0-99 byte size files...
Processing 100-999 byte size files...
Processing TMP folder...
Processing 2 char filenames...
Processing files with suspicious extensions...
Processing files with interesting extensions...
Processing filenames with .tmp extension...
Processing 1 char filenames...
Processing files one directory deep...
Processing self-extracting folders...
Processing batch filenames...
Processing keywords in ioc-terms.txt file...
Processing keywords in ioc-dates-terms.txt file...

TIPS

- --debug is your friend
- If parsing fails on a particular file, set that file aside and try again.
  - Save and send to me for troubleshooting if possible
- Processing SHIM/System hives are much faster than processing Amcache.
  - Quick Wins: Shimcache, RecentFileCache, Jobs
  - Process Amcache by itself
  - Prefetch results in a large number of files, limited results due to Servers / SSDs
  - NTUSER / USRCLASS results in large number of files / data. Not the first choice.
- Can process types individually and then concatenate them together.

THANK YOU!