Working Copy Notes:

Schedule:

- Anti-Forensics: Wed sep 9th
- ~~Jump List on WIN7 complete by thursday...~~
- ~~index.dat …. add to Disk analysis with examples.~~
- recycler?
- disk: $MFT, $LogFile, $I30/INDX maybe?
- ~~disk: Journal analysis~~
- Add windows 8.1 artifacts
- Registry delete, reg cleaners.

Table of Contents

Know your archive: Rosha ARchive(RAR)

Introduction:

There are a lot of great articles already documenting the current RAR format and how one could carve for RARs etc, this article is not meant to replace but to build upon those resources and provide you as the forensicator further insight on how to maximize returns on exfiltrated data. I will cover all areas where the possibility of finding RAR artifacts can either be found or further built upon. This article's focus is on determining artifacts between RAR versions and exploiting those artifacts to gain greater returns. I will be touching on common well documented artifacts such as prefetch, registry, LNK files, and loose files. so you should be familiar with such subjects and have a working understanding of common windows artifacts prior to reading this article. Where I hope to expand and separate this paper from all others that have come before is diving deeper into recovered stream data(carved), specifically version identification, and deep dive rar recovery methods. I believe specifically my methods used to recover and identify exfiled data will take your skills to the next level and allow you to bring more to the table when identifying what has gone out the door at client sites. Hopefully, if this article goes over well it can be a recurring format for common and maybe not so common exfil formats, think tar, gzip, bzip, zip, 7z. In any case a lot of these methods apply.

First and foremost when it comes to exfil data you need to have a plan, direction, and most importantly good time management! If you tackle exfil without gathering intel first you could potentially waste a LOT of time with absolutely little to show or gain from it. I've seen it happen. Specifically, with RAR you need to identify if your threat is using command line or GUI tools. You might be reading this and thinking "no shit", but what is the first thing most forensicators do, open their all-in-one ez-mode solution of choice and start analysing common goto windows artifacts, good job.. who knows how much time you just wasted and found nothing, why? Because your threat was a common APT using command line. Ok, so it could have gone 50/50, but you see where time can be wasted and quick. So have a playbook ready for this, don't dive face first, first gather intel. Have a flow chart… We need to first identify artifacts that quickly answer the question of to GUI or not to GUI. I think its safe to speculate most threats will be using a version of WinRAR 4/5 or binary from WinRar 4/5.

Determining RAR version:

Without spending much time on RAR version history, which is documented fairly well in the wikipedia page among other resources, there are a few notable changes between versions which make for great artifacts. My goal is to give you the ability to quickly determine which RAR version an archive is by simply observing without the need for an external tool. In most cases just observing these subtle difference gives you a strong ballpark of which version of RAR was used and which artifacts you should hunt for. The most notable of these artifacts is the file header change between versions 1.402 and 1.54. If dealing with 1.402 archives and below, I have provided the RAR file format in appendix for legacy RAR 1 archives. With version 2.9 the multi-volume extension changed from *.r01-r0x to part001.rar-part00x.rar. If the rar extensions are in the r01 format, this is a good indicator that you could be looking at a pre 2.9 archive (Note: this is the default behavior, there is still a switch to use the legacy format). When dealing with 2.9 archives and below only the "-p" flag was used to password protect archives, also recovery volumes had not been implemented yet, so you will not

need to bother searching for rev files. The “-hp” switch along with recovery files were introduced in version 3 along with AES-128 encryption and header encryption (NOTE: -p switch could still be used, in most exfil cases you will see “-hp” being used). If you see a .rev file, you are dealing with a 3.0 archive and above. The simplest approach is to load the rar file in question with WinRar[1] and clicking on the “info” button which will provide all meta information pertaining to that archive. I have tested WinRar v5, it appears to be backward compatible with all rar legacy versions. Should you want to automate this process I have provided a sample python script using a modified version of the rarfile[2] library to help determine metadata on an archive.

Right about now you may be thinking these are such small insignificant changes between versions and really have no effect on your current methods. Most RAR headers are Rar!, true. You now know WinRar v5 is backward compatible with legacy formats, true. But what you will now have is the precision in which to operate your investigation, saving considerable hours, and providing intel at even higher accuracy than you before were capable of. The most significant piece that can be derived from these version artifacts is the preciseness in which you will now be able to hone in on password recovery and stronger identifiable indicators of what data exactly has been exfiltrated. These artifacts tell what and when you should be string searching for an “-hp” flag over an “-p” flag. These version will indicate whether or not the file headers are encrypted so that you could report on the contents even without the password being recovered. Below is a chart I have made to simplify this process for you.

Version	Header (RE~^) ‘52 45 7E 5E’	Header(Rar!) ‘52 61 72 21 1A 07 00’	encryption	Password Switch	OS	Recovery
1.3	X		CBC	p		RR
1.402	X		CBC	p		RR
1.54+		X	CBC	p		RR
2.0		X	CBC	p		RR
2.9		X	CBC	p		RR
3.0		X	AES(128)	p,hp		RR,RV
3.93		X	AES(128)	p,hp	98,NT4	RR,RV
4		X	AES(128)	p,hp		RR,RV
5		X	AES(256)	p,hp		RR,RV

(RR- recovery record, RV - Recovery version)

Recovery: (Recovery Volumes(.rev files), Recovery Record, PAR)

[1] http://www.rarlab.com/, Alexander Roshal
[2] https://pypi.python.org/pypi/rarfile, Marko Kreen

TODO: Notable switches to discuss

- -kb (keep broken files)
- -r (repair volume)
- -rc (reconstruct missing volumes) rar 3.90+
- -en Do not add "end of archive" block

Recovery Records:

To determine if a Recovery Record is present you will need to use the correct mapping based on version and examine the MAIN_HEAD->HEAD_FLAGS and determine if a RR is present. The simple approach is explained above using winrar. If you are receiving “CRC” or "archive damaged" error chances are you may be able to recover the damaged archive. Two dependencies must be met however,the file must not be missing any bytes and the archive must have a Recovery Record. If these two requirements have been met then you may use the command line “-r” switch to repair the archive, optionally the winrar gui has a first aid icon at the top right which you may use.

The greater the size of the RR improve the chances of recovering a corrupted RAR file. Prior to 2.7 recovery sectors were limited to 8 sectors, 2.7 extended this to 4096 sectors. “If the sectors number is not specified, the size of the recovery information is about 1.2% of the total archive size, this usually allows the recovery of up to 0.7% of continuously damaged data of the total archive size.”[3]

Recovery Volumes:

Recovery volumes are .rev files. Recovery volumes introduced in RAR 3.0 are used to reconstruct missing and damaged files in a volume set. Each .rev file is able to reconstruct one missing or damaged RAR volume. For example if you have been able to recover four rev files, you will be able to reconstruct four missing rar parts. If the number of missing rar parts exceeds number of .rev files, recovery will not be possible.

At the time of writing this I have found no way of dependably carving for .rev files. There does not appear to be a common magic number; However, there does appear to be some byte logic going on there that could form a fingerprint. From what I’ve noticed

```
if  ((0x00-0x05 & 0x07-0x0c & 0x14-0x19 & 0x1b,0x1c, 0x0x1f-0x21,0x24 ) !=0 ) AND ((0x06 &
0x0d-0x13 & 0x1d-0x1e & 0x22-0x23) == 0):
carve x bytes.
```

This is just a hypothesis and not tested. Your goto method for recovering rev files should be in the volume shadow copy or restore points. (in most cases i doubt rev files would be used in exfil, but with the current methods being deployed client side to disrupt exfil, this could be their method in the future to circumvent this measure, the use of recovery records may also work.

Hypothesis: Replace the current file identification process with headers and develop a reproducible method using light grep to identify file patterns to further enhance exfil recovery and accountability in face of growing exfil data manipulation threats…

“Lightgrep is a true multi-pattern search engine, able to search for many regular expressions and produce independent matches, regardless of their order. Instead of searching text files line-by-line, Lightgrep searches data as a binary stream, allowing you to find patterns in unstructured data.”[4]

---

[3] RAR 2.9 WHATSNET.TXT

[4] http://www.lightboxtechnologies.com/lightgrep-engine/

Parity Archives:

Parity Files or PAR Files are not part of the RAR format; however should be noted due to commonly found supporting RAR uploads particularly with the warez and newsgroup scenes. This could be useful when investigating IP or Child Exploitation Cases. For instance, you have an RAR archive known to have been downloaded from a given FTP or USENET group, you are missing one RAR, you may be able to recover a par file, or look for the par file on the download site to prove what the original archive contained. PAR is basicly the same concept as the RAR Recovery Volume, but supports any type of file or archive.

PAR1
50 41 52 00 00 00 00 00(PAR' followed by 5 null bytes.)[5]

PAR2
50 41 52 32 00 50 4B 54(PAR2PKT)[6][7]

PAR3
50 41 52 33 00 40(PAR3)
(At the time of writing this par3 has been proposed and is in development, header could be subject to change.)[8]

Artifacts Breakdown:

I want to start off by saying this is by no means an exhaustive list of artifacts, I am positive this section will be a continuous work in progress and forever growing as others share or i find additional noteworthy artifacts. I think it is also far to note that there will just be too many to put in one paper, but my goal is to document those that have some weight.

Volatile (memory, hiberfil, pagefile):

Memory can yield big gains from recent command line history and possible carved artifacts. The three discussed here will be raw memory, hiberfil.sys, and pagefile.sys and what artifacts can be recovered from each. Making raw memory the first part of you acquisition is vital. You want to preserve as much of the original data as possible, therefore it should be the first piece collected if possible. Best practice is to use a tool such as one of the tools list HERE and redirect the output either to an attached storage device or file share as to preserve the local disk as best possible. Once this has been achieved you may use a tool such as volatility or rekal to scan and report cmd history located in memory. Using Rekal and the cmdscan plugin the most recent command line history is displayed from a raw memory image. The last line is a good example of striking gold and recovering a RAR password through command line history.

```
jason@Danzig ~/Temp $ rekal -f WINDOWSXP-20140902-232209.raw cmdscan
**************************************************
CommandProcess: csrss.exe Pid: 584
CommandHistory: 0x14ffa00 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 28 LastAdded: 27 LastDisplayed: 27
```

[5] http://parchive.sourceforge.net/docs/specifications/parity-volume-spec-1.0/article-spec.html
[6] http://parchive.sourceforge.net/docs/specifications/parity-volume-spec/article-spec.html#i__134603784_1056
[7] http://en.wikipedia.org/wiki/Parchive
[8] ftp://82.36.19.228/sda/My%20SetUp/Various/Temp/MultiPar118/help/0409/par3_spec_prop.htm

```
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x5c8
Cmd  Address   Text
--- ---------- ----
…
 17 0x014ffbb0 cd ..
 18 0x004f6e98 dir
 19 0x014ffb88 cd RAR.ver.4.02
 20 0x004f6ea8 dir
 21 0x01517c60 rar -h
 22 0x014f3388 dir
 23 0x014f3398 rar -h
 24 0x014f3170 rar a -pT3stP@ssword TEST_CMD_ARCIVE.RAR 1995_CIA_World_Factbook.txt
```

Hiberfil.sys is nearly identical to how a raw memory is treated, the only difference is that you will first need to convert hiberfil.sys to a raw memory image. Tools such as Volatility and some tools listed above can be used to accomplish this. Once converted use the same processes used for raw memory dumps.

Pagefile.sys is a bit tricky and less documented at the time of this article. The good news though is that support has been planned in future releases of Volatility, but as of now the current methods still revolve around searching for strings and very limited file carving. I recommend using the method discussed in the file carving section which is to use bulk_extractor to search for defined strings and or regex taking advantage of the built in lightgrep and multithreaded processing capability.[9]

File Carving:

I have combined disk based string searches with carving, reason being what use to be two processes is now one when using bulk_extractor. This saves time 1. minimizing redundant recursing over data, 2. Takes advantage of multithreading, and 3. takes advantage of light-grep. When it comes to file carving there are primarily two points to remember, first know exactly what you are looking for! In our case which RAR version and associated artifacts, are we carving the new or old header, are we searching for the encrypted or unencrypted password switch strings '-hp' vs. '-p'? Two, know your tool! know exactly what your tool is capable of and is not capable of. As data sizes grow this is very important in time sensitive investigations. Further, it would behove you to understand and be able to form strong regular expressions. The difference between a good forensicator and a GREAT forensicator could just be who had the stronger regex at the end of the day.

At this point you should have a strong understanding of the difference between rar versions and know just what artifacts you are carving and string searching for. In my example below I used bulk_extractor, since I know what exactly I am looking for I will disable all functions that do not directly relate to carving rar and strings, this will greatly improve speed tailored to just what I need.

```
bulk_extractor32.exe -x all -e rar -e find -F wordlist.txt -S rar_carve_mode=2 -o output/windows7-c-drive.E01
windows7-c-drive\windows7-c-drive.E01
```

[9] http://digital-forensics.sans.org/media/memory-forensics-cheat-sheet.pdf
http://www.forensicswiki.org/wiki/Tools:Memory_Imaging
https://www.mandiant.com/blog/memory-acquisition-pagefiles-part-ii/
http://simson.net/ref/2012/2012-02-02%20USMA%20bulk_extractor.pdf
http://www.forensicswiki.org/wiki/Pagefile.sys
http://www.forensicswiki.org/wiki/Hiberfil.sys

bulk_extrator follows args in order which presented on the command line,-x all disables all functions to start with, -e enables each individual function i want to use. learn more at the bulk_extractor project page.

Below are our results:

```
jason@Danzig /media/output/windows7-c-drive.E01 $ ls -la
total 1520
drwx------ 1 jason jason     464 Sep  3 18:02 .
drwx------ 1 jason jason     184 Sep  3 18:01 ..
-rw------- 1 jason jason       0 Jul 26 00:49 alerts.txt
-rw------- 1 jason jason   23437 Jul 26 02:42 find_histogram.txt
-rw------- 1 jason jason 1208756 Jul 26 02:42 find.txt
drwx------ 1 jason jason     136 Jul 26 01:57 rar
-rw------- 1 jason jason   11749 Jul 26 02:33 rar.txt
-rw------- 1 jason jason  304920 Jul 26 02:42 report.xml
```

You will notice that a rar.txt, find.txt, and a directory labeled rar have been newly created. rar.txt will pride text info on what rar files have been recover and now reside in the rar dir. The find.txt will be filled with all the string matches that our regular expressions have hit on. This is where your regex-foo will shine and hopefully present you with a manageable amount of data. (PRO-TIP: if you find yourself with an unmanageable amount of data to sort through, tweak your regex and run it over the find.txt rather than run the carving process all over again)

Below are the recovered RAR files:

```
jason@Danzig /media/output/windows7-c-drive.E01/rar/000 $ ls -la
total 92
drwx------ 1 jason jason  496 Sep  3 18:03 .
drwx------ 1 jason jason  136 Jul 26 01:57 ..
-rw------- 1 jason jason 4136 Jul 26 02:16 14121947136.rar
-rw------- 1 jason jason 4224 Jul 26 02:16 14130044928.rar
-rw------- 1 jason jason 4118 Jul 26 02:26 19366260736.rar
-rw------- 1 jason jason 4129 Jul 26 02:26 19517542400.rar
-rw------- 1 jason jason 4637 Jul 26 02:26 19517661184.rar
-rw------- 1 jason jason   40 Jul 26 02:27 19540577840.rar
-rw------- 1 jason jason   40 Jul 26 02:27 19540775008.rar
-rw------- 1 jason jason   40 Jul 26 02:27 19540811464.rar
-rw------- 1 jason jason   40 Jul 26 02:27 19540821792.rar
-rw------- 1 jason jason   40 Jul 26 02:27 19541357600.rar
-rw------- 1 jason jason   40 Jul 26 02:27 19541408192.rar
-rw------- 1 jason jason   40 Jul 26 02:27 19541457256.rar
-rw------- 1 jason jason   40 Jul 26 02:27 19541597096.rar
-rw------- 1 jason jason   40 Jul 26 02:27 19546355336.rar
-rw------- 1 jason jason   40 Jul 26 02:27 19546427504.rar
-rw------- 1 jason jason   40 Jul 26 02:28 20329066296.rar
-rw------- 1 jason jason   40 Jul 26 02:28 20329122376.rar
-rw------- 1 jason jason   40 Jul 26 02:28 20329577512.rar
-rw------- 1 jason jason   40 Jul 26 02:28 20329637384.rar
-rw------- 1 jason jason   40 Jul 26 02:28 20329666616.rar
-rw------- 1 jason jason 4129 Jul 26 02:31 21404684288.rar
-rw------- 1 jason jason 4133 Jul 26 02:30 21423931392.rar
-rw------- 1 jason jason 8269 Jul 26 02:33 22436765696.rar
-rw------- 1 jason jason 4129 Jul 26 02:33 22439337984.rar
-rw------- 1 jason jason 4142 Jul 26 02:33 22439342080.rar
-rw------- 1 jason jason   40 Jul 26 01:57 9204223064.rar
```

after review of the find.txt, it looks like we hit the goldmine, after doing a quick ctrl+f on -hp to quickly filter through the string file it looks like we have what appears to be the command line used to create RAR archives with the password “!ZP4WQBc^#dUyUJ0PTV”

```
find.txt>
9158624097           -hp!ZP4WQBc^#dUyUJ0PTV exfil6 exfil6\x5C*\x0Aw \x00\x00\x00<\x00\x00\x00<\x00\x00\x00rar a
-hp!ZP4WQBc^#dUyUJ0PTV exfil6 exfil6\x5C*\x0Aw\x00s\x00\x00\x00\x00\x00\x08\x00\x0F\x00MFE0\x04

9158828705           -hp!ZP4WQBc^#dUyUJ0PTV system7 system7\x5C*\x0Ap\xA7\x1Ew
\x00\x00\x00<\x00\x00\x00<\x00\x00\x00rar a -hp!ZP4WQBc^#dUyUJ0PTV exfil5
exfil5\x5C*\x0Ap\xA7\x1Ew\x00\x00\x00\x00\x08\x00\x08\x02CcPc\x00\x00\x00\x00
```

```
wordlist.txt>
\s-[hH][pP].+
\s[pP]assword
```

But wait, we struck gold, but we did more than that. We now have a awesome pivot point to look at other data that could have been saved to disk around the same time. Above is just an example it looks like a account password in plain text, and after what appear to be ftp clients possibly used for exfil, and some very suspicious domains. Now be aware that this could be from anything possible an AV definition file, so you will want to rule that out. For me since it is so close to the obvious exfil activity I smell blood.

```
8991011061     \x0APassword   e=\x0D\x0ADomainName=\x0D\x0APassword=esgwwjHYSJajas
8991154421     \x0APassword   e=\x0D\x0ADomainName=\x0D\x0APassword=esgwwjHYSJajas
--pivot point--
9272751761     \x09Password   bia.com\x08tibia.pl\x09Password:\x05John:\x08sion\x5CRun
9272753912     \x09password
{DOWN}\x09%s\x5Ctmp.%d\x09password:\x84\x02\x00\x17\xFF\x1B\xC4\x19\xFF\x01\x03\x00\x05{T
```

Registry:

Before tackling registry it is strongly advised that you determine the threats primary archiving method, command line or GUI. Knowing your threat will save you time and frustration. The reason I say this is it really depends on the sophistication of the threat. Most APT cases will be commandline rar, where less sophisticated threats will most likely be using winrar. The fact is if an attacker used command line RAR there will be slim if any registry artifacts in which to recover. Best case scenario the threat is less sophisticated and used WinRar. If WinRar was used then the potential for recovering registry artifacts has drastically increased providing potential to recover saved passwords, archive names, and time activity pivot points. Below I have separated artifacts by their hive, their location, a quick example of the artifact, and I have also noted what type of data can be established by the artifact next to the artifacts name. I will not be covering details of artifacts here.

NTUSER Hives:

Artifact: UserAssist *(Where: GUI, What: Pivot Times, Location)*

XP ntuser.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\?*\Count

W7 ntuser.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\?*\Count

```
{75048700-EF1F-11D0-9888-006097DEACF9}\count | UEME_RUNPATH:C:\RAR\RAR.ver.3.1\rarx310.exe
{75048700-EF1F-11D0-9888-006097DEACF9}\count | UEME_RUNPATH:C:\RAR\RAR.ver.3b\rarx300.exe
{75048700-EF1F-11D0-9888-006097DEACF9}\count | UEME_RUNPATH:C:\RAR\RAR.ver.3b\RAR.ver.3.00 Beta.English\RAR.EXE
```

Artifact: Open/Save MRU

XP: ntuser.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU\?*

W7: ntuser.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePidlMRU\?*

```
09/02/2014 |  20:17:57.259 |     h | *          | C:\Documents and Settings\jason\Desktop\RAR1_36B.RAR
09/02/2014 |  20:17:57.259 |     g | *          | E:\RAR1_36B.RAR
09/02/2014 |  18:45:10.641 |  -> a | exe        | C:\Program Files\WinRAR4\WinRAR.exe
```

Artifact: LastVisitedMRU *(Where: GUI, What:* Pivot Point*, Location)*

XP: NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedMRU

W7: ntuser.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedPidlMRU

```
reg date    |  reg-UTC       | item# | filename
09/02/2014 |  19:53:18.651 |  -> a | Greenshot.exe
09/02/2014 |  19:53:18.651 |     f | OLLYDBG.EXE
09/02/2014 |  19:53:18.651 |     e | chrome.exe
09/02/2014 |  19:53:18.651 |     d | QuickPar.exe
09/02/2014 |  19:53:18.651 |     c | WinRAR.exe
09/02/2014 |  19:53:18.651 |     b | NOTEPAD.EXE
```

Artifact: CIDSizeMRU

W7: ntuser.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\CIDSizeMRU

```
Timestamp: 0x01cfdcfd81103ef6 (09/30/2014 22:25:56.719 UTC)

owner sid [S-1-5-21-3817071845-1377787776-1327029102-1000]
group sid [S-1-5-21-3817071845-1377787776-1327029102-513]

Discretionary Access Control List
access allowed  Restricted Code READ_CONTROL
access allowed  Local System    DELETE | READ_CONTROL | WRITE_DAC | WRITE_OWNER
access allowed  S-1-5-21-3817071845-1377787776-1327029102-1000 DELETE | READ_CONTROL | WRITE_DAC | WRITE_OWNER
access allowed  Admins          DELETE | READ_CONTROL | WRITE_DAC | WRITE_OWNER

0                                       REG_BINARY
0000 0000:  57 00 69 00 6e 00 52 00  41 00 52 00 2e 00 65 00  W.i.n.R.A.R...e.
0000 0010:  78 00 65 00 00 00 00 00  00 00 00 00 00 00 00 00  x.e.............
```

Artifact: RecentDocs *(Where: GUI, What: Archive Names, Pivot Times)*

XP: NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs

W7: ntuser.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs

```
09/05/2014 |  03:09:53.285 |      4 |        | RAR                       | RAR.lnk
09/05/2014 |  03:09:53.285 |      5 |        | h.txt                     | h.txt.lnk
09/05/2014 |  03:09:53.285 |     25 |        | Downloads                 | Downloads.lnk
09/05/2014 |  03:09:53.285 |     38 |        | DumpIt.zip                | DumpIt.zip.lnk
09/05/2014 |  03:09:53.285 |     37 |        | New WinRAR archive.rar    | New WinRAR archive.rar.lnk
09/05/2014 |  03:09:53.285 |     36 |        | QuickPar-0.9.1.0.rar      | QuickPar-0.9.1.0.rar.lnk
09/05/2014 |  03:09:53.285 |      3 |        | JASN.rar                  | JASN.rar.lnk
09/05/2014 |  03:09:53.285 |     35 |        | WinRAR4                   | WinRAR4.lnk
09/05/2014 |  03:09:53.285 |      1 |        | File_Id.diz               | File_Id.diz.lnk
09/05/2014 |  03:09:53.285 |     14 |        | rarfiles.lst              | rarfiles.lst.lnk
09/05/2014 |  03:09:53.285 |     34 |        | UnrarSrc.txt              | UnrarSrc.txt.lnk
```

Artifact: AppCompat

XP: SYSTEM\CurrentControlSet\Control\SessionManager\AppCompatibility

W7: SYSTEM\CurrentControlSet\Control\Session Manager\AppCompatCache

```
reg date    |  reg-UTC       | subkey     | value name        | value data
09/30/2014 |  22:19:49.492 | Persisted | C:\wrar511b1.exe | 0x00000001 (1)
```

WinRAR Specific *(Where: GUI, What: Archive Names, Location, Pivot Times)*

XP, W7: ntuser.dat\Software\WinRAR\DialogEditHistory\ArcName\

XP, W7: NTUSER.DAT\Software\WinRAR\ArcHistory

```
Artifact: WinRAR Recent Files
Registry key: NTUSER.DAT\Software\WinRAR\ArcHistory
-----------------------------------------------------------------------------------------------------

reg date    |  reg-UTC       | value name | value data
09/02/2014 |  22:59:28.008 | 0          | C:\Documents and Settings\jason\My Documents\Downloads\DumpIt.zip
09/02/2014 |  22:59:28.008 | 1          | C:\Documents and Settings\jason\Desktop\QuickPar-0.9.1.0.rar
09/02/2014 |  22:59:28.008 | 2          | C:\Documents and Settings\jason\My Documents\Downloads\pf32.v.1.04.win.zip
09/02/2014 |  22:59:28.008 | 3          | C:\Documents and Settings\jason\My Documents\Downloads\sbag32.v.0.37.win.zip
```

Shell Bags[10] *(Where: GUI, What: Archive Names, Location, Pivot Times)*

XP:

USRCLASS.DAT\Local Settings\Software\Microsoft\Windows\Shell\BagMRU

USRCLASS.DAT\Local Settings\Software\Microsoft\Windows\Shell\Bags

NTUSER.DAT\Software\Microsoft\Windows\Shell\BagMRU

---

[10] http://digital-forensics.sans.org/blog/2011/07/05/shellbags

NTUSER.DAT\Software\Microsoft\Windows\Shell\Bags
ntuser.dat\Software\Microsoft\Windows\ShellNoRoam\MUICache

```
09/15/2014 |  22:21:58.193 | C:\Program Files\WinRAR4\WinRAR.exe                          | WinRAR archiver
09/15/2014 |  22:21:58.193 | C:\Program Files\Windows NT\Accessories\WORDPAD.EXE          | WordPad
```

W7:
NTUSER.DAT\Software\Microsoft\Windows\ShellNoRoam\BagMRU
NTUSER.DAT\Software\Microsoft\Windows\ShellNoRoam\Bags
USRCLASS.DAT\Wow6432Node\Local Settings\Software\Microsoft\Windows\Shell\Bags
USRCLASS.DAT\Wow6432Node\Local Settings\Software

```
0x00028288 [a] : C:\Program Files (x86)\WinRAR\WinRAR.exe : UsrClass.dat\Local Settings\Software\Microsoft\Windows\Shell\MuiCache\
0x0003c6f2 [a] :                                  WinRAR : UsrClass.dat\Local Settings\Software\Microsoft\Windows\Shell\BagMRU\1\0\1\0\0\0\0\
0x000281cc [u] :                          WinRAR archiver : UsrClass.dat\Local Settings\Software\Microsoft\Windows\Shell\MuiCache\
0x0003c724 [u] :                                  WinRAR : UsrClass.dat\Local Settings\Software\Microsoft\Windows\Shell\BagMRU\1\0\1\0\0\0\0\
```

Artifact: ProgramsCache *(Where: GUI, What: Archive Names, Location, Pivot Point)*
XP: NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\StartPage\ProgramsCache
w7: ntuser.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\StartPage2\ProgramsCache

```
09/05/2014 |  04:25:10.492 | 09/02/2014 |  18:10:48    | 09/02/2014 |  21:14:58    | 09/02/2014 |  18:10:48    |            | Programs\WinRAR
09/05/2014 |  04:25:10.492 | 09/02/2014 |  18:10:48    | 09/02/2014 |  18:10:48    | 09/02/2014 |  18:10:48    | 0x000002b2 | Programs\WinRAR\Console RAR manual.lnk
09/05/2014 |  04:25:10.492 | 09/02/2014 |  18:10:48    | 09/02/2014 |  18:10:48    | 09/02/2014 |  18:10:48    | 0x000002c5 | Programs\WinRAR\WinRAR.lnk
09/05/2014 |  04:25:10.492 | 09/02/2014 |  18:10:48    | 09/02/2014 |  18:10:48    | 09/02/2014 |  18:10:48    | 0x000002c5 | Programs\WinRAR\WinRAR help.lnk
09/05/2014 |  04:25:10.492 | 05/03/2011 |  14:41:58    | 09/02/2014 |  23:22:10    | 09/02/2014 |  22:59:58    | 0x00032a88 | Programs\WinRAR\DumpIt.exe
09/05/2014 |  04:25:10.492 | 08/20/2014 |  17:08:42    | 09/04/2014 |  22:04:26    | 09/04/2014 |  22:04:26    | 0x00015137 | Programs\WinRAR\index.dat
09/05/2014 |  04:25:10.492 | 08/21/2014 |  16:14:44    | 09/02/2014 |  22:24:00    | 08/21/2014 |  16:14:44    | 0x000ccfb4 | Programs\WinRAR\par1.p01
09/05/2014 |  04:25:10.492 | 08/21/2014 |  16:14:44    | 09/02/2014 |  22:23:40    | 08/21/2014 |  16:14:44    | 0x000000b0 | Programs\WinRAR\par1.par
09/05/2014 |  04:25:10.492 | 09/02/2014 |  22:24:58    | 09/02/2014 |  22:24:58    | 09/02/2014 |  22:24:58    | 0x000cfba0 | Programs\WinRAR\par1.rar
09/05/2014 |  04:25:10.492 | 11/28/2001 |  06:04:00    | 08/21/2014 |  16:14:44    | 08/21/2014 |  16:06:36    | 0x0001e000 | Programs\WinRAR\par.exe
09/05/2014 |  04:25:10.492 | 09/05/2014 |  01:20:38    | 09/05/2014 |  01:20:38    | 09/05/2014 |  01:20:38    | 0x0005ca00 | Programs\WinRAR\Prefetch.rar
09/05/2014 |  04:25:10.492 | 08/21/2014 |  16:06:44    | 09/02/2014 |  19:53:14    | 08/21/2014 |  16:06:44    | 0x000002aa | Programs\WinRAR\QuickPar.lnk
09/05/2014 |  04:25:10.492 | 08/21/2014 |  16:02:48    | 09/02/2014 |  22:16:04    | 08/21/2014 |  16:06:16    | 0x0007a673 | Programs\WinRAR\QuickPar-0.9.1.0.exe
09/05/2014 |  04:25:10.492 | 09/02/2014 |  22:16:42    | 09/02/2014 |  22:16:42    | 09/02/2014 |  22:16:40    | 0x00072a2a | Programs\WinRAR\QuickPar-0.9.1.0.rar
09/05/2014 |  04:25:10.492 | 08/14/2014 |  20:00:58    | 08/21/2014 |  16:09:50    | 08/21/2014 |  16:09:10    | 0x000ccf04 | Programs\WinRAR\RAR1_36B.RAR
09/05/2014 |  04:25:10.492 | 08/21/2014 |  16:11:02    | 09/02/2014 |  22:24:00    | 08/21/2014 |  16:11:02    | 0x000001b8 | Programs\WinRAR\RAR1_36B.RAR.par2
09/05/2014 |  04:25:10.492 | 08/21/2014 |  16:11:02    | 08/21/2014 |  16:11:02    | 08/21/2014 |  16:11:02    | 0x0005ddfc | Programs\WinRAR\RAR1_36B.RAR.vol0+1.PAR2
09/05/2014 |  04:25:10.492 | 09/05/2014 |  01:21:02    | 09/05/2014 |  01:21:02    | 09/05/2014 |  01:21:02    | 0x000000c4 | Programs\WinRAR\Shortcut to Temp on 'vboxsrv' (E).lnk
09/05/2014 |  04:25:10.492 | 09/02/2014 |  23:22:18    | 09/02/2014 |  23:22:18    | 09/02/2014 |  23:22:12    | 0x3fff0000 | Programs\WinRAR\WINDOWSXP-20140902-232209.raw
09/05/2014 |  04:25:10.492 | 09/02/2014 |  18:10:48    | 09/02/2014 |  23:04:26    | 09/02/2014 |  18:10:48    | 0x000002b9 | Programs\WinRAR\WinRAR.lnk
09/05/2014 |  04:25:10.492 | 09/03/2014 |  19:23:06    | 09/03/2014 |  19:23:06    | 08/14/2014 |  19:17:36    | 0x00000715 | Programs\WinRAR\Google Chrome.lnk
```

#SYSTEM HIVE
Artifact: Shim Cache *(*Where: GUI, What: Pivot Point, Location)
XP: HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\AppCompatibility\AppCompatCache
W7: HKLM\System\CurrentControlSet\Control\Session Manager\AppCompatCache\AppCompatCache

```
07/03/2004 |  09:34:17.171 | 08/21/2014 |  16:09:42.296 |    90 |    29 | c:\program files\quickpar\quickpar.exe
07/12/2010 |  12:55:03.000 | 09/02/2014 |  20:51:59.105 |    14 |     8 | c:\program files\windows nt\accessories\wordpad.exe
03/02/2011 |  19:41:03.843 | 09/02/2014 |  18:10:39.334 |     6 |    20 | c:\program files\winrar4\uninstall.exe
03/02/2011 |  19:39:06.573 | 09/02/2014 |  20:57:42.701 |     0 |     3 | c:\program files\winrar4\winrar.exe
08/19/2014 |  02:33:08.181 | 08/21/2014 |  16:06:05.509 |    59 |    31 | c:\program files\winrar\rarext.dll
08/19/2014 |  02:33:07.370 | 09/02/2014 |  18:09:18.427 |    45 |    22 | c:\program files\winrar\uninstall.exe
08/19/2014 |  02:33:07.869 | 09/02/2014 |  17:58:37.247 |    85 |    26 | c:\program files\winrar\winrar.exe
```

#Software HIVE
Artifact: App Paths *(*Where: GUI, What: Pivot Point, Location)
XP: Software\Microsoft\Windows\CurrentVersion\App Paths\
W7: HKLM\Software\Microsoft\Windows\CurrentVersion\App Paths\

```
reg date    |  reg-UTC       | subkey         | value name              | value data
09/02/2014 |  18:10:45.998 | WinRAR.exe     | Path                    | C:\Program Files\WinRAR4
09/02/2014 |  18:10:45.998 | WinRAR.exe     |                         | C:\Program Files\WinRAR4\WinRAR.exe
08/21/2014 |  16:06:43.664 | QuickPar.exe   |                         | C:\Program Files\QuickPar\QuickPar.exe
```

Deleted Registry Slack
TODO: rewrite deleted.pl in python, create examples of delted winrar entries.

Registry Afterthoughts

- Though not directly related to RAR, but I would be looking for indicators of the user dropping down into command prompt whether it be cmd.exe or powershell:
  - ***runMRU***
    XP, W7: ntuser.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU

```
reg date   |  reg-UTC       | item# | filename
09/02/2014 |  20:59:03.565 |  -> a | cmd\1
09/02/2014 |  20:59:03.565 |     c | regedit\1
09/02/2014 |  20:59:03.565 |     b | winrar\1
```

- W7 has a scheduled task which creates backups of software, system, security, and sam hives located in C:\Windows\system32\config\RegBack every 10 days.
- VSS will have additional Registry backups.
- Notable signs of WinRar being installed:

```
XP, W7: registry hits for string search on "B41DB860-8EE4-11D2-9906-E49FADC173CA"
W7: SOFTWARE\Classes\*\shellex\ContextMenuHandlers\WinRAR32\
W7: SOFTWARE\Classes\Folder\ShellEx\ContextMenuHandlers\WinRAR32\
W7: SOFTWARE\Classes\Folder\ShellEx\DragDropHandlers\WinRAR32\
W7: SOFTWARE\Classes\Drive\shellex\DragDropHandlers\WinRAR32\
W7: ntuser.dat\Software\WinRAR SFX
```

Disk:

Jump List*(TAG: GUI, Archive Names, Locations, Pivot Times)*

WIN7: C:\Users\%USERNAME%\Recent\AppData\Roaming\Microsoft\Windows\Recent\

WIN7: C:\Users\%USERNAME%\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations

Windows 7+ viewing the AutomaticDestinations could lead to file names and staging directories. Much like registry, Jump List artifacts will be most lucrative when a threat is using a GUI based program such as WinRAR to extract or create archives. When dealing with more sophisticated threats such as APT and those using command line you may not find a lot here.[11]

```
jason@Danzig:~/Desktop/bin/tzworks/jmp64.v.0.27.lin$ ls -la /mnt/vmdk/Users/Jason/AppData/Roaming/Microsoft/Windows/Recent/AutomaticDestinations/
total 24
drwxrwxrwx 1 root root 4096 Oct  1 18:58 .
drwxrwxrwx 1 root root 4096 Oct  1 18:58 ..
-rwxrwxrwx 2 root root 6656 Oct  1 18:58 1b4dd67f29cb1962.automaticDestinations-ms
-rwxrwxrwx 2 root root 3072 Oct  1 18:58 290532160612e071.automaticDestinations-ms
-rwxrwxrwx 2 root root 3584 Oct  1 18:55 7e4dca80246863e3.automaticDestinations-ms
```

```
source path/filename:     /mnt/vmdk/Users/Jason/AppData/Roaming/Microsoft/Windows/Recent/AutomaticDestinations/290532160612e071.automaticDestinations-ms
MRU/MFU index:            1
stream #:                 1
MRU time:                 10/01/2014 22:58:13.082 [UTC]
file modified:            10/01/2014 22:58:13 [UTC]
file accessed:            10/01/2014 22:58:13 [UTC]
file stats changed:       10/01/2014 22:58:13 [UTC]
MFT Entry:                0x0000a2c5
MFT Sequence#:            0x0003
Target flags:             HasLinkTargetIDList, HasLinkInfo, IsUnicode
Target attributes:        FILE_ATTRIBUTE_ARCHIVE
Target modified:          09/30/2014 22:22:30.700 [UTC]
Target accessed:          09/30/2014 22:22:30.417 [UTC]
Target created:           09/30/2014 22:22:30.417 [UTC]
Target ObjID time:        09/30/2014 22:13:59.156 [UTC]
Parsed size:              0x00000222 [546 bytes]
Target file size:         0x000c4337 [803639 bytes]
Show cmd:                 [SW_SHOWNORMAL]
ID List:                  {CLSID_MyComputer}\C:\winzip_compression_benchmark_files.rar
Volume Type:              fixed
Volume serial num:        c0ab-12ef
Local base path:          C:\winzip_compression_benchmark_files.rar
NETBIOS name:             win-e2mg8kkl2b2
Volume ID:                3c5fb87e-7e27-4c6e-9a6c-11cbcf75e823
Object ID:                139f8bd5-48ef-11e4-b78e-000c29fc61a4
MAC address:              00:0c:29:fc:61:a4
```

LNK files*(TAG: GUI, Archive Names, Locations, Pivot Times)*

XP C:\Documents and Settings\<username>\Recent\

W7 C:\Users\<user>\AppData\Roaming\Microsoft\Windows\Recent\

Win7 C:\Users\<user>\AppData\Roaming\Microsoft\Office\Recent\

Note these are primary locations, but are found in other locations as well.

LNK files associated with rar files are goldmines of intel. Seen below you will have path locations now, file names, and meta. Some of the meta could provide useful data correlating MAC times with rar file times providing an idea of what files could have been exfiled.

---

[11] http://www.forensicswiki.org/wiki/Jump_Lists

source path/filename:	/home/jason/Documents/VMs/Windows XP/file.raw
file offset:	0xc03ed7a8
file modified:	09/08/2014 21:23:31 [UTC]
file accessed:	09/08/2014 21:23:51 [UTC]
file stats changed:	09/08/2014 21:23:31 [UTC]
Target flags:	HasLinkTargetIDList
Target attributes:	FILE_ATTRIBUTE_NORMAL
Target modified:	08/14/2014 20:53:55.996 [UTC]
Target accessed:	08/19/2014 20:00:12.154 [UTC]
Target created:	08/15/2014 03:11:14.281 [UTC]
File offset:	0xc03ed7a8 [3225343912]
Parsed size:	0x0000015e [350 bytes]
Target file size:	0x000ccf30 [839472 bytes]
Show cmd:	[SW_SHOWNORMAL]
ID List:	E:\rar\RAR1_54B.RAR
Network name:	\\vboxsrv\Temp
Device name:	E:
Common path:	rar\RAR1_54B.RAR

source path/filename:	/home/jason/Documents/VMs/Windows XP/file.raw
file offset:	0xc0f3efa0
file modified:	09/08/2014 21:23:31 [UTC]
file accessed:	09/08/2014 21:23:51 [UTC]
file stats changed:	09/08/2014 21:23:31 [UTC]
Target flags:	HasLinkTargetIDList
Target attributes:	FILE_ATTRIBUTE_ARCHIVE
Target modified:	09/02/2014 20:52:17.191 [UTC]
Target accessed:	09/02/2014 20:52:31.472 [UTC]
Target created:	09/02/2014 20:52:17.191 [UTC]
Target ObjID time:	08/21/2014 16:06:42.664 [UTC]
File offset:	0xc0f3efa0 [3237212064]
Parsed size:	0x000001cb [459 bytes]
Target file size:	0x00000014 [20 bytes]
Show cmd:	[SW_SHOWNORMAL]
ID List:	JASN.rar
Volume Type:	fixed
Volume serial num:	200e-1183
Local base path:	C:\Documents and Settings\jason\Desktop\JASN.rar
Relative path:	..\Desktop\JASN.rar
Working directory:	C:\Documents and Settings\jason\Desktop
NETBIOS name:	windowsxp
Volume ID:	7666eaf6-9695-44a2-9c3b-a3888eac5213
Object ID:	2453b2cc-294d-11e4-b58b-08002726b78d

Prefetch *(TAG: GUI, Archive Names, Locations, Pivot Times)*

W7/XP C:\Windows\Prefetch

You should be checking prefetch for any obvious signs such as the pf files pertaining to RAR processes, if any sort of GUI based transfer client was used you might find artifacts of that too. If WinRAR was used it is possible that you will find cached rar history when winrar loads and it tries to touch cache RAR file history. A pivot time could be obtained from pf timestamps, run location, path location could also be possible to obtain here, and possible rar names. Note, if you find multipart file names, these could lead you in the direction of what format to carve for and further artifacts to hunt for.

Listing of suspend prefetch files:

```
-rwxrwxrwx 2 root root  66962 Aug 20 13:21 WGATRAY.EXE-0ED38BED.pf
-rwxrwxrwx 2 root root  60112 Sep  4 21:20 WINRAR.EXE-2499FD54.pf
-rwxrwxrwx 2 root root  28412 Sep  2 13:58 WINRAR.EXE-39C6DAD9.pf
-rwxrwxrwx 2 root root  33410 Aug 20 14:50 WMIADAP.EXE-2DF425B2.pf
-rwxrwxrwx 2 root root  46834 Sep  4 18:55 WMIPRVSE.EXE-28F301A9.pf
-rwxrwxrwx 2 root root  22716 Sep  2 16:52 WORDPAD.EXE-24533991.pf
-rwxrwxrwx 2 root root  49574 Sep  2 14:10 WRAR400.EXE-247D62FB.pf
```

Example of RAR archive names being obtained from a prefetch file:

WFSERVICESREG.EXE-063492A2.pf	\DEVICE\HARDDISKVOLUME1\DOCUMENTS AND SETTINGS\JASON\MY DOCUMENTS\DOWNLOADS\DUMPIT.ZIP
WGANOTIFYPACKAGEINNER.EXE-24665	\DEVICE\HARDDISKVOLUME1\DOCUMENTS AND SETTINGS\JASON\DESKTOP\PAR1.RAR
WGATRAY.EXE-0ED38BED.pf	\DEVICE\HARDDISKVOLUME1\DOCUMENTS AND SETTINGS\JASON\DESKTOP\PAR1.PAR
WINRAR.EXE-2499FD54.pf	\DEVICE\HARDDISKVOLUME1\DOCUMENTS AND SETTINGS\JASON\DESKTOP\NEW WINRAR ARCHIVE (2).RAR
WINRAR.EXE-39C6DAD9.pf	\DEVICE\HARDDISKVOLUME1\PROGRAM FILES\WINRAR4\FORMATS\TAR.FMT
WMIADAP.EXE-2DF425B2.pf	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSTEM32\WIN32K.SYS
WMIPRVSE.EXE-28F301A9.pf	\DEVICE\HARDDISKVOLUME1\DOCUMENTS AND SETTINGS\JASON\DESKTOP\NEW WINRAR ARCHIVE.RAR
WORDPAD.EXE-24533991.pf	\DEVICE\HARDDISKVOLUME1\WINDOWS\SYSTEM32\DRIVERS\KMIXER.SYS
	\DEVICE\HARDDISKVOLUME1\DOCUMENTS AND SETTINGS\JASON\DESKTOP\QUICKPAR-0.9.1.0.RAR
	\DEVICE\HARDDISKVOLUME1\DOCUMENTS AND SETTINGS\JASON\DESKTOP\QUICKPAR-0.9.1.0.EXE

$MFT(From: GUI, CMDLINE What: Archive Names, Pivot Times)

$UsnJrnl(From: GUI, CMDLINE What: Archive Names, Pivot Times)

W7, XP if enabled.

```
10/01/2014 |  22:57:26.351 | 0x00000000103e | 0x0005 | 0x0000000001ce | 0x0002 | 0x000001570808 | archive
| WinRAR - Shortcut.lnk | data_appended; file_created; file_closed

10/01/2014 |  22:57:41.376 | 0x000000001090 | 0x0004 | 0x000000003f0c | 0x0003 | 0x000001570a10 | archive
| WinRAR - Shortcut.lnk | data_overwritten; data_appended; file_created;basic_info_changed;file_closed
```

```
10/01/2014 |  22:57:41.378 | 0x000000001090 | 0x0004 | 0x000000003f0c | 0x0003 | 0x000001570ae0 | archive
| WinRAR - Shortcut.lnk | objid_changed; file_closed

10/01/2014 |  22:58:13.084 | 0x00000000a2c5 | 0x0003 | 0x000000000005 | 0x0005 | 0x000001570bd0 | archive
| winzip_compression_benchmark_files.rar        | objid_changed; file_closed
```

index.dat(From: GUI, CMDLINE What: Archive Names, Pivot Times)

XP:

09/02/2014	20:57:16 UTC	URL	:2014090220140903: jason@file:///C:/Documents%20and%20Settings/jason/Desktop/New%20WinRAR%20archive.rar
09/02/2014	20:55:10 UTC	URL	:2014090220140903: jason@file:///C:/Documents%20and%20Settings/jason/Desktop/QuickPar-0.9.1.0.rar
09/05/2014	03:09:38 UTC	URL	:2014090420140905: jason@file:///E:/rar/h.txt
09/02/2014	17:15:50 UTC	URL	:2014090220140903: jason@file:///E:/rar/reg_artifacts.txt
09/02/2014	20:39:14 UTC	URL	:2014090220140903: jason@file:///C:/Program%20Files/WinRAR4/UnrarSrc.txt
09/02/2014	20:54:26 UTC	URL	:2014090220140903: jason@file:///C:/Documents%20and%20Settings/jason/Desktop/JASN.rar
09/02/2014	20:57:16 UTC	URL	:2014090220140903: jason@file:///C:/Documents%20and%20Settings/jason/Desktop/New%20WinRAR%20archive.rar
09/02/2014	20:55:10 UTC	URL	:2014090220140903: jason@file:///C:/Documents%20and%20Settings/jason/Desktop/QuickPar-0.9.1.0.rar
08/19/2014	17:58:36 UTC	URL	Visited: jason@file:///E:/rar/clipboard.txt

Volume Shadow Copy(vss)

File carving is not always the best place to recover deleted archives. Simply carving will miss any deleted archives still present in the vss due to the mechanics in which vss data is stored. In this case you will need to review vssadmin output, the log for rar files and search through the vss for RAR files. I have created a script to automate this process for you. It will search each file header and report those matching the RAR header formats.

```
Danzig mnt # python test.py
ewfmount 20140608

vshadowmount 20140731

/mnt/vss/vss4/Users/Public/Temp/system4.rar
/mnt/vss/vss4/Users/Public/Temp/system5.rar
/mnt/vss/vss4/Users/Public/Temp/system6.rar
/mnt/vss/vss4/Users/Public/Temp/system7.rar
/mnt/vss/vss7/Users/Public/Temp/system4.rar
/mnt/vss/vss7/Users/Public/Temp/system5.rar
/mnt/vss/vss7/Users/Public/Temp/system6.rar
/mnt/vss/vss7/Users/Public/Temp/system7.rar
/mnt/vss/vss5/Users/Public/Temp/system4.rar
/mnt/vss/vss5/Users/Public/Temp/system5.rar
/mnt/vss/vss5/Users/Public/Temp/system6.rar
/mnt/vss/vss5/Users/Public/Temp/system7.rar
/mnt/vss/vss6/Users/Public/Temp/system4.rar
/mnt/vss/vss6/Users/Public/Temp/system5.rar
/mnt/vss/vss6/Users/Public/Temp/system6.rar
/mnt/vss/vss6/Users/Public/Temp/system7.rar
```

Disk Afterthoughts:

1. Between version 3 and version 4 the commandline version was no longer being released separately, but included with the winrar package. From experience I have noticed that the bundled version creates “C\:Users\<USER>\AppData\Roaming\WinRar”, however unlike the GUI it does not create the “version.dat” file in the above directory. This could be an quick indicator of a few things:
    a. If he above directory exists, rar has been used by the profile.
    b. If a version.dat file is not present, then most likely commandline has been used.
    c. If “version.dat” exists we have an artifact indicating winrar was used and what version.

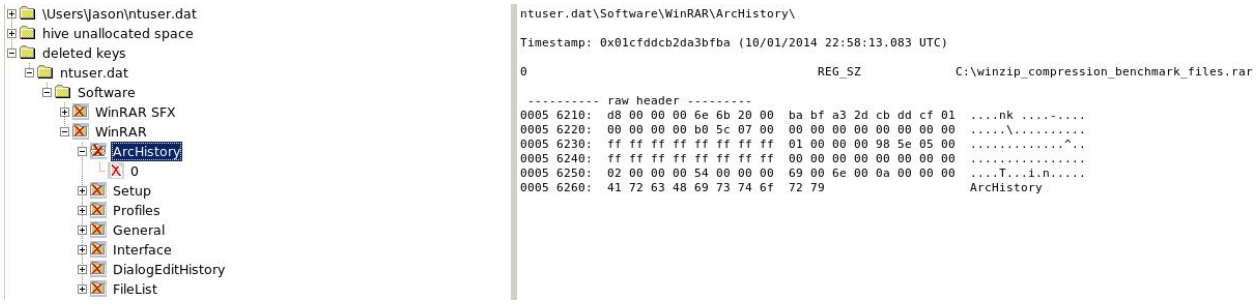
RAR Anti-Forensics:

WinRar

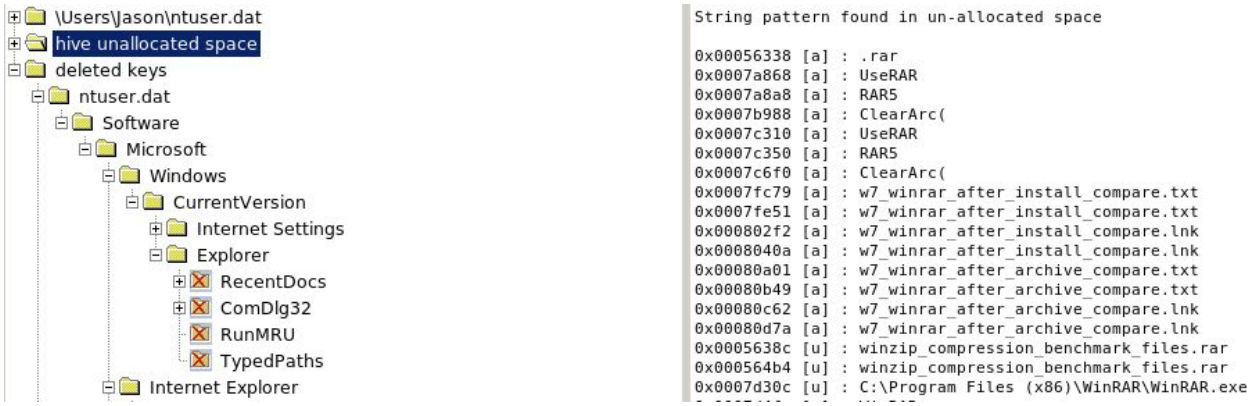
- rarnew.dat appears to be the file header stamp being used when creating new archives through the context menu. This could be used to alter the RAR! file header into an unique header to bypass file carving techniques.
- as with rarnew.dat, more advanced threats could manually edit the rar headers with hex editors or even automate this to help bypass carving techniques.
- What's the answer? I want to coin the term bitprint, the digital algorithm finger print, not using the header to determine the archive format but a producible algorithm that could bitprint file types based on identified structures and low level binary or hex patterns, much like ssdeep.

Registry

Below is an example of a windows 7 box just after using regedit to delete the WinRar registry entries. Notice I can still see the keys and structures under deleted keys allowing me to recover the ArcHistory key.



Below is an example taking anti forensics one step further. Below is a snapshot after running both CCleaner and Privazer. What I found, was that though I was unable to recover RAR related data under deleted keys, running string searches under unallocated registry space still gave me a lot of artifacts. I expected it to be zeroed or sevened out, which was not the case here.



Command Line History

- powershell - does not save history over sessions, ie stored in memory only.
  live triage command - "get-history"
- CMD.EXE
  live triage command - "doskey /history"

APPENDIX A:

RAR 1 File Format:

(Previously Undocumented)

Archive Header:

HEAD_ID	4 bytes	0x52 0x45 0x7e 0x5e
HEAD_LEN	2 bytes	header archive, including comments
HEAD_FLAGS	1 byte	0x01 - a sign of Volume (archive volume) 0x02 - there is a comment to the archive 0x04 - the archive was protected from modification 0x08 - a sign of Solid (solid archive)
COMM_LEN	1 byte	Present only if (HEAD_FLAGS & 0x02)! = 0
MAIN_COMMENT		Present only if (HEAD_FLAGS & 0x02)! = 0

Header File, in the archive:

PACK_LEN	4 byte	file size in compressed form
UNP_LEN	4	source file size
CHECK_SUM	2	checksum
HEAD_LEN	2	Full length 2 bytes file header, including comments to the file and the line with the name of the file
FTIME	4	Date and time of the file in a standard format MS DOS
ATTR	1	file attributes
FLAGS	1	0x01 - file is continued from the previous volume 0x02 - file is continued in the next volume 0x04 - the file is protected with a password 0x08 - there is a file comment
VER	1	Minimum version of RAR to unpack: 0 - RAR 0.99 (currently not supported) 1 - RAR 1.00 2 - RAR 1.30
NAME_LEN	1	length of the file name
METHOD	1	Compression level: 0 - storing , 1 - fastest , 2 - fast , 3 - normal ,4 - good, 5 - maximal
FILE_COMM_LEN	Present only if (FLAGS & 0x08)! = 0	length file comment.
FILE_COMMENT	Present only if (FLAGS & 0x08)! = 0	Comments to the file.
FILE_NAME	Filename - string length NAME_LEN	filename

APPENDIX B:

Understanding the Recovery Volume(REV) schema:

Lets take a really simple case.

You have four volumes and a recovery volume, each with 1 bit of information in it:

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

Volumes: = Volume 1 = Volume 2 = Volume 3 = Volume 4 = Recovery =
=----------=----------=----------=----------=----------=

Bits: = 1 = 0 = 1 = 1 = 1 =

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

The recovery volume would contain the result of each of these bits XORd together:

```
1 XOR 0 XOR 1 XOR 1 = 1
```

So, our recovery volume contains the single bit `1`.

Now, lets say volume 1 fails.

If we XOR the remaining volumes 2, 3 and 4 with the recovery bit in place of the failed volume we get:

```
1 XOR 0 XOR 1 XOR 1 = 1 ^
```

So, this tells us that volume 1 contained `1`, since it is the result of the equation.

Let us pretend volume 2 died instead, so we replace it's value in the equation with the recovery bit:

```
1 XOR 1 XOR 1 XOR 1 = 0 ^
```

So we know that volume 2 contained `0`, since it is the result of the equation.

If volume 3 or 4 failed, they would both produce `1` in this equation.

So, if any of the volumes failed, the recovery volume can be used to reconstruct the data based on the remaining volumes. This is probably the simplest form of error correction you can have. If two volumes failed, you can't recover anything.[12]

---

12 http://superuser.com/questions/507456/how-is-it-possible-that-winrar-can-repair-any-volume-with-one-rev-file

APPENDIX C:

Common Recovery Errors:

TODO: Errors

"Unexpected end of archive" *- My take on this is could be either the full file was not recovered, or the byte order is off. If in case of the first, a rev file will be needed to recover. In case of the later, if a recovery record is present you may try repairing.*

"file header is corrupt" *- I would first check the first 7 bytes to ensure they are correct. If so treat as you would a CRC error.*
"Unknown Method" - This is a good indication that the rar algorithm being used is not supported by your rar program, most likely this points to the rar format being used is newer than your rar program being used. Try using the latest version of winrar.

"Packed data CRC failed in volume name. The volume is corrupt"! *-This message may be displayed together with the message "CRC failed in file name. The file is corrupt" and may help to detect exactly which volume is corrupt. If a big file is split between a few volumes and only one volume has been damaged after archive creation. Depending on corrupt archive details, RAR may display or skip this message. The message "CRC failed in file name. The file is corrupt" is always displayed if the file data are damaged. Check to see if you have installed the recent version. If not, please download the actual WinRAR/RAR version here. We always improve all functions with every update. (http://www.win-rar.com/faq.html?&L=0)*

How can I extract a volume to view the content? *-You can unpack the archives using the "Keep broken files" option from the "Extraction path and options dialog". But most of the files don't work if they are broken. You can try it. Perhaps a mpg video works, if you unpack only the first parts. For most of the files you need the file header. The file header is placed in the first part of the file. In the header you find the information, if it's a movie or a text or a sound file. So the best way is to download the first volume and extract it with the "keep broken files" option. If you have ie: "sample.r08", extraction only works, if the multi volume archives are non-solid. You can see information about the archive using the info function. But not all movies will be played with the first part. Some players check the length of the file and if it is different from the header entry they report an error. The same occurs with pictures.(http://www.win-rar.com/faq.html?&L=0)*

I get an error like "CRC failed in file name. The file is corrupt"! -*File data are corrupt. Archive may be damaged after creation. In this case it is sometimes possible to repair it if it has the recovery record. It also could have been broken while creating, due to hardware failures (usually caused by an overclocked CPU or unreliable memory). In the latter case the recovery record will not help. The only way to detect if the repair will help is to try to apply the repair command to the archive and test the results. Check, if you have installed the recent version - if not, please download the actual WinRAR/RAR version from here. We always improve all functions with every update.(http://www.win-rar.com/faq.html?&L=0)*

APPENDIX D:

Useful Scripts:

```
# Desc: recursively runs bulk_extractor over image files, useful when carving and string
# searching working copy drives filled with images.Run script sit back and kick up your feet.
#
#bulk_extractor -x all -e aes -o output diskimage.raw
#
#-S zip_min_uncompr_size=6
#-S zip_max_uncompr_size=268435456
#-S rar_find_components=YES
#-S raw_find_volumes=YES Search for RAR volumes (rar)
#-S unrar_carve_mode=1 0=carve none; 1=carve encoded; 2=carve all (rar)


#bulk_extractor -S rar_carve_mode=1 -o output diskimage.raw
import os

basedir = r"\\192.168.45.128\cases\windows7-c-drive"
outputdir = r"C:\Users\jblanks\Desktop\rar\output"
imagetypes = [".e01", ".dd",".DD",".001",".E01"]
wordlist = "wordlist.txt"

for root, dirs, files in os.walk(basedir):
    for f in files:
        if (f.endswith(".e01") or f.endswith(".E01") or f.endswith(".001") or
f.endswith(".dd") or f.endswith(".DD")):
            os.system("bulk_extractor32.exe -x all -e zip -e rar -e outlook -e find -F
"+wordlist+" -S rar_carve_mode=2 -o "+outputdir+"/"+f+" "+os.path.join(root,f))


#Shell script
#mounts all shadow copies
for i in vss*; do mount -o ro,loop,show_sys_files,streams_interface=windows $i
/mnt/volume_shadows/$i; done

TODO: write script to auto-mount vss then scan each file for rar header and return results.
aka script kiddie vss bulk_extractor
```