

FRAC Documentation

Table of Contents

Retrieve Interesting Files Tool (RIFT).....	2
Gathering system files to a USB drive using the command line.....	2
Gathering system files to a network share using the command line.....	3
Forensic Response ACquisition (FRAC).....	5
FRAC: Gathering Files from *NIX.....	10
Running Other Commands with FRAC.....	13
FRAC Impact.....	15
Winexe Install on OSX.....	17
Sleuthkit-APFS Notes.....	18
“Compiling” FRAC and RIFT.....	20
License.....	21

Figures

Figure 1: Gathering system files step 4.....	3
Figure 2: Gathering system files step 4.....	4
Figure 3: PAExec Output example.....	8
Figure 4: winexe Output Example.....	9
Figure 5: Contents Of The go.bat File to Capture Memory.....	14
Figure 6: Altered line of the cmd.txt file to run the go.bat batch file.....	14
Figure 7: Virtual Machine Information.....	15
Figure 8: Performance Monitor Stats.....	16

Retrieve Interesting Files Tool (RIFT)

Retrieve Interesting Files Tool (RIFT) was written to obtain a set of files/directories in an automated forensically sound manner. RIFT retrieves files/directories based upon a regex list of filenames/directories. The tool starts off by parsing the output from the Sleuthkit's FLS command of the \$MFT. Each line of output is compared to the regex list to check for a match. If there is a match, Sleuthkit's ICAT is used to forensically retrieve the file and save it to the location specified.

Gathering system files to a USB drive using the command line

Tools Needed

rft.exe – Perl script that uses fls.exe and icat.exe to retrieve a predefined list of files on target file system. Perl is not required as the script has been compiled into a binary.

icat.exe – Part of Sleuthkit for Windows (Download at <http://sourceforge.net/projects/sleuthkit/files/sleuthkit/4.0.1/sleuthkit-win32-4.0.1.zip/download>)

fls.exe – Part of Sleuthkit for Windows (Download at <http://sourceforge.net/projects/sleuthkit/files/sleuthkit/4.0.1/sleuthkit-win32-4.0.1.zip/download>)

libewf.dll – Part of Sleuthkit for Windows (Download at <http://sourceforge.net/projects/sleuthkit/files/sleuthkit/4.0.1/sleuthkit-win32-4.0.1.zip/download>)

zlib.dll – Part of Sleuthkit for Windows (Download at <http://sourceforge.net/projects/sleuthkit/files/sleuthkit/4.0.1/sleuthkit-win32-4.0.1.zip/download>)

getfiles.txt – Contains a regex list of files and directories that will be retrieved

USB Drive – The drive needs to be big enough to hold the files. If the machine has 8 gigs of ram then, the USB drive needs to be 25 gigs or bigger. The pagefile.sys and hiberfil.sys will be requested, due to the size of both these files, it is best to plan on three times the amount of memory being required to complete this task.

IMPORTANT: Everything you do on the machine will change memory. DO NOT perform any other tasks (other than using getfiles.exe) on the machine you are going to gather the files from. Please do not run other applications or execute any other commands as it may erase evidence in memory that is needed.

Getting Started

- 1) Make sure your USB drive is empty and big enough to store the files
- 2) Copy the tools onto the USB drive

Gathering System Files:

- 1) Insert the USB drive into the machine you want to obtain the live memory image from
- 2) Bring up a command prompt. **NOTE:** The command prompt will need Administrator privileges.
- 3) Change directory to where you copied the tools on the USB drive
- 4) Execute getfiles.exe --savedrive=<USB drive path>

```

Administrator: Command Prompt
Y:\zzz>rift.exe --savedrive y:\zzz
Retrieve Interesting Files Tool (RIFT) -- Version 0.02

Saving files to: y:\zzz\WIN-589EGM23G1A_762015_10-1-20
Reading Y:\zzz\getfileslist.txt for the list of files to gather.
Recovering File from WIN-589EGM23G1A (172.16.13.54): /Windows/Tasks/GoogleUpdat
eTaskMachineCore.job Cluster/Inode:74985-128-4
Recovering File from WIN-589EGM23G1A (172.16.13.54): /Windows/Tasks/GoogleUpdat
eTaskMachineUA.job Cluster/Inode:82514-128-4
Recovering File from WIN-589EGM23G1A (172.16.13.54): /Windows/Tasks/SA.DAT
Cluster/Inode:40692-128-58
Recovering File from WIN-589EGM23G1A (172.16.13.54): /Windows/Tasks/SCHEDLGU.TXT
Cluster/Inode:40693-128-3
Recovering File from WIN-589EGM23G1A (172.16.13.54): /Windows/System32/config/S
YSTEM Cluster/Inode:58932-128-3

Y:\zzz>

```

Figure 1: Gathering system files step 4

Gathering system files to a network share using the command line

Tools Needed

rift.exe – Perl script that uses fls.exe and icat.exe to retrieve a predefined list of files on target file system. Perl is not required as the script has been compiled into a binary.

icat.exe – Part of Sleuthkit for Windows (Download at <http://sourceforge.net/projects/sleuthkit/files/sleuthkit/4.0.1/sleuthkit-win32-4.0.1.zip/download>)

fls.exe – Part of Sleuthkit for Windows (Download at <http://sourceforge.net/projects/sleuthkit/files/sleuthkit/4.0.1/sleuthkit-win32-4.0.1.zip/download>)

libewf.dll – Part of Sleuthkit for Windows (Download at <http://sourceforge.net/projects/sleuthkit/files/sleuthkit/4.0.1/sleuthkit-win32-4.0.1.zip/download>)

zlib.dll – Part of Sleuthkit for Windows (Download at <http://sourceforge.net/projects/sleuthkit/files/sleuthkit/4.0.1/sleuthkit-win32-4.0.1.zip/download>)

getfiles.txt – Contains a regex list of files and directories that will be retrieved

Network Share Drive – The drive needs to be big enough to hold the files. If the machine has 8 gigs of ram then, the USB drive needs to 25 gigs or bigger. The pagefile.sys and hiberfil.sys will be requested, due to the size of both these files, it is best to plan on three times the amount of memory being required to complete this task.

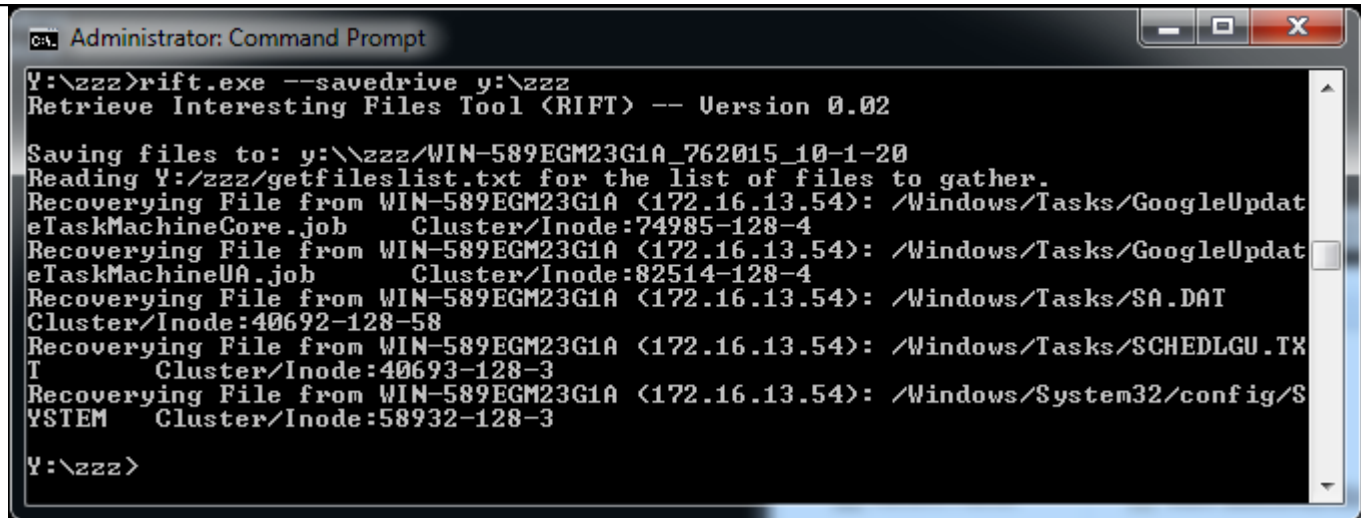
IMPORTANT: Everything you do on the machine will change memory. DO NOT perform any other tasks (other than using getfiles.exe) on the machine you are going to gather the files from. Please do not run other applications or execute any other commands as it may erase evidence in memory that is needed.

Getting Started

- 1) Make sure your network share drive is empty and big enough to store the files
- 2) Copy the tools onto the network share drive
- 3) Try mounting the network shared drive from a different computer than the one you are going to take the image of. You need to ensure that the machine has the same permissions as the machine you are going to image. This test is to ensure that the machine will be able to see the network share and reduce changes to memory (see **IMPORTANT NOTE** above).

Gathering System Files

- 1) Map the network share on the machine you will be taking the files from.
- 2) Bring up a command prompt. **NOTE:** The command prompt will need Administrator privileges.
- 3) Change directory to the network share.
- 4) Execute getfiles.exe --savedir=savedrive<mapped network drive letter>



```
Administrator: Command Prompt
Y:\zzz>rift.exe --savedrive y:\zzz
Retrieve Interesting Files Tool (RIFT) -- Version 0.02

Saving files to: y:\zzz\WIN-589EGM23G1A_762015_10-1-20
Reading Y:\zzz\getfileslist.txt for the list of files to gather.
Recovering File from WIN-589EGM23G1A (172.16.13.54): /Windows/Tasks/GoogleUpdat
eTaskMachineCore.job Cluster/Inode:74985-128-4
Recovering File from WIN-589EGM23G1A (172.16.13.54): /Windows/Tasks/GoogleUpdat
eTaskMachineUA.job Cluster/Inode:82514-128-4
Recovering File from WIN-589EGM23G1A (172.16.13.54): /Windows/Tasks/SA.DAT
Cluster/Inode:40692-128-58
Recovering File from WIN-589EGM23G1A (172.16.13.54): /Windows/Tasks/SCHEDLGU.TX
T Cluster/Inode:40693-128-3
Recovering File from WIN-589EGM23G1A (172.16.13.54): /Windows/System32/config/S
YSTEM Cluster/Inode:58932-128-3

Y:\zzz>
```

Figure 2: Gathering system files step 4

Forensic Response ACquisition (FRAC)

Forensic Response ACquisition (FRAC) is a network tool that uses RIFT to retrieve forensically interesting files. Its primary goal is to pull back files for review during incident response. The tool will take an IP range and connect to each machine to run a command. If it cannot connect to an IP address it will log the IP as unresponsive so that it can be re-ran at a later time. FRAC uses either PAExec or Winexe to connect to the remote Windows boxes. Once connected, it will run the command given to it on the machine and then disconnect. Primarily, FRAC is used to retrieve files like Atjobs or system hives, however, it is possible to retrieve the system memory using Winpmem. The section entitled “Running Other Commands with FRAC” has details on how to run other commands with FRAC.

As with any tool, the author recommends you test FRAC on a small subset of machines to ensure that the tool performs to your specifications.

Windows: Tools Needed

rft.exe – Perl script that uses fls.exe and icat.exe to retrieve a predefined list of files on target file system.

icat.exe – Part of Sleuthkit for Windows (Download at <http://sourceforge.net/projects/sleuthkit/files/sleuthkit/4.0.1/sleuthkit-win32-4.0.1.zip/download>)

fls.exe – Part of Sleuthkit for Windows (Download at <http://sourceforge.net/projects/sleuthkit/files/sleuthkit/4.0.1/sleuthkit-win32-4.0.1.zip/download>)

libewf.dll – Part of Sleuthkit for Windows (Download at <http://sourceforge.net/projects/sleuthkit/files/sleuthkit/4.0.1/sleuthkit-win32-4.0.1.zip/download>)

zlib.dll – Part of Sleuthkit for Windows (Download at <http://sourceforge.net/projects/sleuthkit/files/sleuthkit/4.0.1/sleuthkit-win32-4.0.1.zip/download>)

cmd.txt – This file contains the command that is used to do the work on each machine.

getfiles.txt – Contains a regex list of files and directories that will be retrieved

config.ini – Used to configure how the program runs.

iplist.txt – This file contains the network ranges used to scan for interesting files.

frac.exe – This is the program that will read in the IP ranges and conduct the scans.

Network Share (SMB) – The network share should be fairly large. It is hard to recommend a size. It all depends on what is being downloaded to the share. Also, the share needs to be mountable by the machines that will be scanned. The author recommends that an admin mount the share by hand to ensure that it will mount correctly.

Paexec – Used to run processes on the remote machines. (Download at <http://www.poweradmin.com/paexec/>; Source code available at: <https://github.com/poweradminllc/PAExec>)

Winexec – Used to run processes on the remote machines. Try to find a pre-compiled version as winexe can be a bear to compile. (Source code: <http://sourceforge.net/projects/winexe>)

PYExec.exe – Used to run processes on remote machines. The new versions of Windows 10 work with this and is what is recommended to be used over Paexec.

*NIX: Tools Needed

rft – Perl script that uses fls.exe and icat.exe to retrieve a predefined list of files on target file system.

icat – Part of Sleuthkit for Windows (Download at <http://sourceforge.net/projects/sleuthkit/files/sleuthkit/4.0.1/sleuthkit-win32-4.0.1.zip/download>)

fls – Part of Sleuthkit for Windows (Download at <http://sourceforge.net/projects/sleuthkit/files/sleuthkit/4.0.1/sleuthkit-win32-4.0.1.zip/download>)

cmd.txt – This file contains the command that is used to do the work on each machine.

getfiles.txt – Contains a regex list of files and directories that will be retrieved

config.ini – Used to configure how the program runs.

iplist.txt – This file contains the network ranges used to scan for interesting files.

frac – This is the program that will read in the IP ranges and conduct the scans.

Network Share (NFS) – The network share should be fairly large. It is hard to recommend a size. It all depends on what is being downloaded to the share. Also, the share needs to be mountable by the machines that will be scanned. The author recommends that an admin mount the share by hand to ensure that it will mount correctly.

IMPORTANT: Please insure any processes that reboot the machines happens after the network scan is done. Otherwise critical machines may be skipped during the process of collecting the files.

IMPACT: As with any network scan there is some impact. When a machine is scanned, there will be some disk IO and network impact to the performance of the machine. The disk IO is not high, but some users may notice the hard drive lights staying on longer than normal. Network impact depends on how many and how large the files are that are requested to be pulled back. For example, if pagefile.sys is requested to be pulled back by FRAC, it will effect network performance in that pagefile.sys is a large file. While one to 5 machines may not be too bad in regards to network performance, 20+ machines may adversely affect network performance in the amount of available bandwidth. The author recommends making smart decisions on what is really required to be pulled back for review.

Methodology

The solution uses two possible methods to remotely connect the machines. The methods with pros and cons are listed below:

- PYExec – Recommended method as the user will not have any visual windows popping up during the scan.
 - Works best for Windows 10 and new Windows OSes.
 - Pros: Same basic functionality and pros that PAExec has. In addition, PYExec has an encryption option.
 - Cons: Some AV vendors may detect PAExec as malicious as it has been used by actors.
- Paexec – Recommended method as the user will not have any visual windows popping up during the scan.
 - Pros: Paexec is freeware and does not require any additional software to be purchased. Also, it can conduct the scan silently. The user will not see any windows pop up during the scan. In addition PAExec will scramble the parameters to protect them from casual wire sniffers, but they are NOT encrypted.
 - Cons: Some AV vendors may detect PAExec as malicious as it has been used by actors.
- Winexe – Recommended method as the user will not have any visual windows popping up during the scan.
 - Pros: Winexe is open source and GNU licensed. It runs on Linux or *NIX. Also, it can conduct the scan silently. The user will not see any windows pop up during the scan.
 - Cons: Some type of Linux/*NIX/OSX would be needed to run the scan from. In some environments this may be a problem.
- PSEXec – PSEXec was tested. While it does work, it was decided that PSEXec would not work due to the way PSEXec works in regards to the system account. In order for FRAC to gather up the files, it requires system level privileges. Using PSEXec with the system flag, creates a window the users can disable the triage. By closing down the window or denying access, the user will effective cancel the scan on the box and there is no feedback to FRAC that the scan was cancelled.

Getting Started: Network share

- 1) Create and share out the network share. Please make sure all of the machine that are part of the scan can mount the share. The author recommends that after the share is setup, an administrator mounts the share by hand to ensure it is working correctly.

Note: If your organization wishes to use Samba for the network share that is OK. All of the tools have been tested with Samba.

- 2) The following files must be on the network share:

- rift.exe
- icat.exe
- fls.exe
- libewf.dll
- zlib.dll
- getfiles.txt
- config.ini

Getting Started: Text File Configuration (cmd.txt)

- 1) Chose a method (PYExec/PAExec/Winexe) to use that meets the requirements of the environment.
- 2) Put the appropriate command into cmd.txt:
 - PYExec – Used if using a Windows box to control the file gathering
`pyexec --dest [IP] --user [ADMINID] --pass [ADMINPASS] --cmd cmd --cmdarg "net use [SHAREDRV] /delete /yes & net use [SHAREDRV] [SHARE] /user:[SHAREUSERID] [SHAREPASSWD] && cd /d [SAVEDRIVE] && rift.exe --verbose --savedrive [SAVEDRIVE] && net use [SHAREDRV] /delete /yes"`
 - Paexec – Used if using a Windows box to control the file gathering
`paexec.exe \\[IP] -n 4 /SYSTEM -u [ADMINID] -p [ADMINPASS] -s cmd /C "net use [SHAREDRV] [SHARE] /user:[SHAREUSERID] [SHAREPASSWD] && cd /d [SAVEDRIVE] && rift.exe --savedrive [SAVEDRIVE] && net use [SHAREDRV] /delete /yes"`

- Winexe – Used if using a Linux/*Nix/OSX box to control the file gathering:

#Winexe version 1.0fe

```
[FULL PATH to binary]/winexe --user [ADMINID] --password=[ADMINPASS] --uninstall --system //[IP] "cmd /C net use [SHAREDRV] \\[SHARE] /user:[SHAREUSERID] [SHAREPASSWD] && cd /d [SAVEDRIVE] && rift.exe --savedrive [SAVEDRIVE] && net use [SHAREDRV] /delete /yes"
```

#Winexe version 1.1

```
[FULL PATH to binary]/winexe --user=[ADMINID]%[ADMINPASS] --uninstall --system //[IP] "cmd /C net use [SHAREDRV] \\[SHARE] /user:[SHAREUSERID] [SHAREPASSWD] && cd /d [SAVEDRIVE] && rift.exe --savedrive [SAVEDRIVE] && net use [SHAREDRV] /delete /yes"
```

FRAC is able to run multiple commands on the same machine. For example, in addition to the system files, a memory dump of the machine was required. By adding a second command into the cmd.txt file, this can be done. The example below will gather up the requested files and then execute go.bat which creates a copy of the system's memory.

```
paexec.exe \\[IP] -n 4 -u [ADMINID] -p [ADMINPASS] -s cmd /C "net use [SHAREDRV] /delete /yes & net use [SHAREDRV] [SHARE] /user:[SHAREUSERID] [SHAREPASSWD] && cd /d [SAVEDRIVE] && rift.exe --verbose --savedrive [SAVEDRIVE] && net use [SHAREDRV] /delete /yes"
paexec.exe \\[IP] -n 4 -u [ADMINID] -p [ADMINPASS] -s cmd /C "net use [SHAREDRV] /delete /yes & net use [SHAREDRV] [SHARE] /user:[SHAREUSERID] [SHAREPASSWD] && cd /d [SAVEDRIVE] && go.bat && net use [SHAREDRV] /delete /yes"
```

- 3) The method that was picked in step 2 will need some adjustments done to command listed in cmd.txt file. Adjust the appropriate line as follows:

Machine/domain – put in the domain used for the network. If a single machine is being scanned and it is not part of the domain, put the machine name in.

Administrator – Put the administrator account that the program should use to connect to the remote box.

Password – Enter in the admin account password for the account entered above.

Drive – Pick a drive letter that is not current used in the environment.

Share – Enter in the name of the share.

Userid – Enter in the account used to mount the share.

Sharepassword – Enter in the account password used to mount the share.

Double check to ensure all of the fields are correct.

NOTE: Do not change the following in the command line. The program will replace the place holders with the correct information.

- [IP]
- [ADMINID]
- [ADMINPASS]
- [SHAREDRV]
- [SHARE]
- [SHAREUSERID]
- [SHAREPASSWD]
- [SAVEDRIVE]

For example Paexec was selected at the method because a Windows box was selected:

```
paexec.exe \\[IP] -n 4 /SYSTEM -u admin -p Password123 -s cmd /C "net use z: \\172.16.1.32 /user:admin Password123 && cd /d z: && rift.exe --savedrive z: && net use z: /delete /yes"
```

- 4) Save the cmd.txt file.

Getting Started: Text File Configuration (iplist.txt)

- 1) The iplist.txt file contains the list of network ranges or single IP addresses of which machines to scan. Below are some examples of valid lines for the file:

```
#Example of valid IP/ranges
192.168.1.10
192.168.5.5-192.168.5.25 admin Go1234DW=
172.16.13.0/24
10.0.0.0/16
```

Note: The program is setup to ignore any lines starting with #, blank spaces, or empty lines. DO NOT a # for comments after the IP address or network range.

Note: The program will ignore the broadcast address for any CIDR notation entered. It calculates the broadcast address automatically. IT WILL NOT be able to figure out the broadcast address for IP ranges like 192.168.5.5-192.168.5.25. For example, if 192.168.5.25 is a broadcast address for the network, do not put it in the range.

- 2) After entering the IP address/range put in the admin account to be used and password. Please use TABs between each field.
For example:
10.0.0.0/16 admin password123
- 3) After all of the IP address and ranges are entered save the file.

Running the Scan

- 1) Ensure all of the altered text files and files are saved to the share.

Note: All of the retrieved data will be saved to the share.

- 2) Run the scan:
frac.exe --iplist iplist.txt --cmd cmd.txt

Figure 3 shows an example sessions for PAExec.

```
M:\zzz>frac --iplist iplist.txt --cmd cmd.txt
FRAC (Forensic Response ACquisition) -- Version 0.04

Using config.ini located at: M:\zzz\config.ini
Reading iplist.txt for the list of IPs/Networks.
Saving unreachable IPs to: unreachableips_7202015_11-27-51.txt
Reading cmd.txt for the CMD(s) to run.
Will execute the following on each IP:
paexec.exe \\[IP] -n 4 -u [ADMINID] -p [ADMINPASS] -s cmd /C "net use [SHAREDROU] /delete /yes & net use [SHAREDROU] [SHARE
ER] /user:[SHAREUSERID] [SHAREPASSWD] && cd /d [SAVEDRIVE] && rift.exe --verbose --savedrive [SAVEDRIVE] && net use [SHAR
EDROU] /delete /yes"

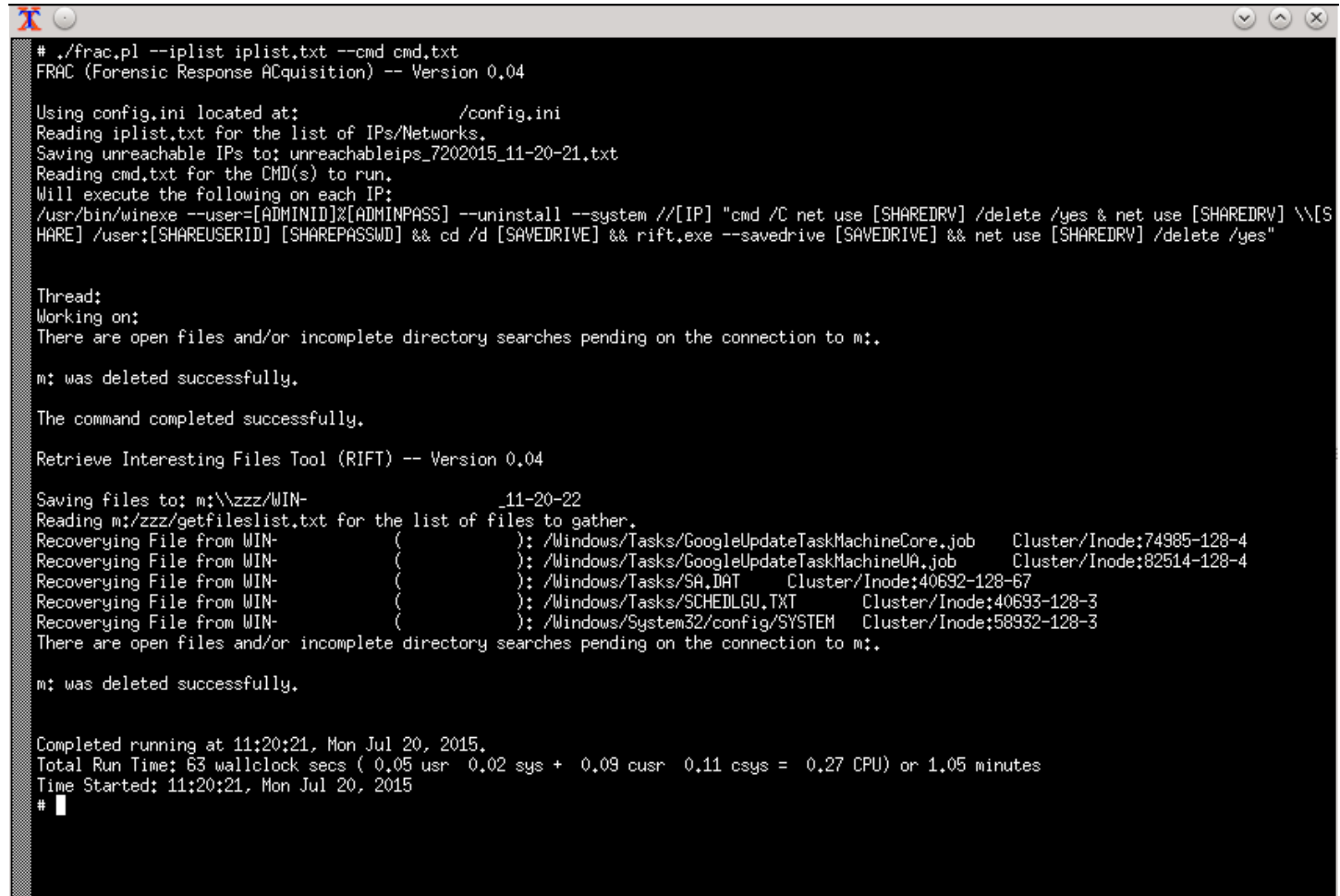
Thread:
Working on:
There are open files and/or incomplete directory searches pending on the connection to m:.
m: was deleted successfully.
The command completed successfully.
Retrieve Interesting Files Tool (RIFT) -- Version 0.04
Saving files to: m:\zzz\WIN-07202015_11-27-53
Reading m:\zzz\getfileslist.txt for the list of files to gather.
Searching for:
system32\config\SYSTEM$
\Windows\Tasks\

Recovering File from WIN-589EGM23G1A < 985-128-4 >: /Windows/Tasks/GoogleUpdateTaskMachineCore.job Cluster/Inode:74
Recovering File from WIN-589EGM23G1A < 128-4 >: /Windows/Tasks/GoogleUpdateTaskMachineUA.job Cluster/Inode:82514-
Recovering File from WIN-589EGM23G1A < >: /Windows/Tasks/SA.DAT Cluster/Inode:40692-128-67
Recovering File from WIN-589EGM23G1A < >: /Windows/Tasks/SCHEDLGU.TXT Cluster/Inode:40693-128-3
Recovering File from WIN-589EGM23G1A < >: /Windows/System32/config/SYSTEM Cluster/Inode:58932-128-3

Completed running at 11:27:51, Mon Jul 20, 2015.
Total Run Time: 79 wallclock secs < 0.05 usr 0.01 sys + 0.00 cusr 0.00 csys = 0.06 CPU> or 1.31666666666667 minutes
Time Started: 11:27:51, Mon Jul 20, 2015
M:\zzz>
```

Figure 3: PAExec Output example

Figure 4 shows an example session using winexe.



```
# ./frac.pl --iplist iplist.txt --cmd cmd.txt
FRAC (Forensic Response Acquisition) -- Version 0.04

Using config.ini located at: /config.ini
Reading iplist.txt for the list of IPs/Networks.
Saving unreachable IPs to: unreachableips_7202015_11-20-21.txt
Reading cmd.txt for the CMD(s) to run.
Will execute the following on each IP:
/usr/bin/winexe --user=[ADMINID]%[ADMINPASS] --uninstall --system //[IP] "cmd /C net use [SHAREDV] /delete /yes & net use [SHAREDV] \\[S
HARE] /user:[SHAREUSERID] [SHAREPASSWD] && cd /d [SAVEDRIVE] && rift.exe --savedrive [SAVEDRIVE] && net use [SHAREDV] /delete /yes"

Thread:
Working on:
There are open files and/or incomplete directory searches pending on the connection to m:.

m: was deleted successfully.

The command completed successfully.

Retrieve Interesting Files Tool (RIFT) -- Version 0.04

Saving files to: m:\\zzz\\WIN- _11-20-22
Reading m:\\zzz\\getfileslist.txt for the list of files to gather.
Recovering File from WIN- ( ): /Windows/Tasks/GoogleUpdateTaskMachineCore.job Cluster/Inode:74985-128-4
Recovering File from WIN- ( ): /Windows/Tasks/GoogleUpdateTaskMachineUA.job Cluster/Inode:82514-128-4
Recovering File from WIN- ( ): /Windows/Tasks/SA.DAT Cluster/Inode:40692-128-67
Recovering File from WIN- ( ): /Windows/Tasks/SCHEDLGU.TXT Cluster/Inode:40693-128-3
Recovering File from WIN- ( ): /Windows/System32/config/SYSTEM Cluster/Inode:58932-128-3
There are open files and/or incomplete directory searches pending on the connection to m:.

m: was deleted successfully.

Completed running at 11:20:21, Mon Jul 20, 2015.
Total Run Time: 63 wallclock secs ( 0.05 usr 0.02 sys + 0.09 cusr 0.11 csys = 0.27 CPU) or 1.05 minutes
Time Started: 11:20:21, Mon Jul 20, 2015
# █
```

Figure 4: winexe Output Example

After the scan

After running the scan, any IP addresses that were not reachable are listed in a file that starts with “unreachableips_” and ends with the date & time of the scan. To hit those IP addresses that were missed, just use the unreachable text file for the --iplist option.

FRAC: Gathering Files from *NIX

*NIX: Tools Needed

rift – Perl script that uses fls.exe and icat.exe to retrieve a predefined list of files on target file system.

icat – Part of Sleuthkit for *NIX version you need files from

fls – Part of Sleuthkit for *NIX version you need files from

cmd.txt – This file contains the command that is used to do the work on each machine.

getfiles.txt – Contains a regex list of files and directories that will be retrieved

config.ini – Used to configure how the program runs.

iplist.txt – This file contains the network ranges used to scan for interesting files.

frac – This is the program that will read in the IP ranges and conduct the scans.

Network Share (NFS) – The network share should be fairly large. It is hard to recommend a size. It all depends on what is being downloaded to the share. Also, the share needs to be mountable by the machines that will be scanned. The author recommends that an admin mount the share by hand to ensure that it will mount correctly.

IMPORTANT: Please insure any processes that reboot the machines happens after the network scan is done. Otherwise critical machines may be skipped during the process of collecting the files.

IMPACT: As with any network scan there is some impact. When a machine is scanned, there will be some disk IO and network impact to the performance of the machine. The disk IO is not high, but some users may notice the hard drive lights staying on longer than normal. Network impact depends on how many and how large the files are that are requested to be pulled back. For example, if pagefile.sys (from Windows) is requested to be pulled back by FRAC, it will effect network performance in that pagefile.sys is a large file. While one to 5 machines may not be too bad in regards to network performance, 20+ machines may adversely affect network performance in the amount of available bandwidth. The author recommends making smart decisions on what is really required to be pulled back for review.

OS Notes:

Linux – No problems observed. Just make sure all boxes scanned have the appropriate libraries for the binaries.

OSX – The Sleuthkit toolset doesn't have capabilities for APFS. However, a fork of Sleuthkit, available at <https://github.com/blackbagtech/sleuthkit-APFS>, by BlackBag Technologies can access APFS. Please see OX notes for more information.

Methodology

The solution uses two possible methods to remotely connect the machines. The methods with pros and cons are listed below:

- **SSH** – SSH is used to remote into each machine. It is recommended that the machine running FRAC have the package sshpass installed on it. The sshpass package make it easier for scripting the password to the SSH service when logging in to the remote machine.

Getting Started: Network share

- 1) Create and share out the network share. Please make sure all of the machine that are part of the scan can mount the share. The author recommends that after the share is setup, an administrator mounts the share by hand to ensure it is working correctly.

Note: If your organization wishes to use Samba for the network share that is OK. All of the tools have been tested with Samba.

- 2) The following files must be on the network share:
 - rift (compiled version of rift.pl)
 - icat
 - fls
 - getfiles.txt
 - config.ini

Note: If your environment that will be scanned has different OSes, you will need different binaries for RIFT, icat, and fls for each OS. The next section below on cmd.txt contains details on how this would work.

Getting Started: Text File Configuration (cmd.txt)

- 1) Chose a method (sshpas/ssh) to use that meets the requirements of the environment.
- 2) Put the appropriate command into cmd.txt:
 - sshpas

```
sshpas -p [ADMINPASS] ssh [IP] 'mkdir [SHAREDRV];mount [MNTOPTS] [SRCBOX]:[SRCMNT] [SHAREDRV] ; cd [SHAREDRV]/
frac_v0.05a && ./rift_`uname -s` --verbose --os=`uname -s` --savedrive [SAVEDRIVE] ; cd / && umount [SHAREDRV] &&
rmdir [SHAREDRV]' 2>&1
```

Note: The `uname -s` in the command above is an easy way to get each version of *NIX to run the appropriate binary. So for each different OS type, run “uname -s” on machine to get the value. Then rename the binary like icat to icat_{uname -s value}. For example on a Linux box the value produced by “uname -s” is “Linux”. Therefore the icat binary should be renamed to “icat_Linux”.

- 3) The method that was picked in step 2 will need some adjustments done to command listed in cmd.txt file. Adjust the appropriate line as follows:

adminpass – The root account password. FATE should be ran as root and will log into each box as root.
 sharedrv – The directory where the NFS/SMB share will be mounted. This is the directory where the data will be saved to.
 mntopts – Any mount options that are need to mount the share.
 srcbox – The machine that has the NFS/SMB share.
 srcmnt – The share directory from the srcbox.
 savedrive – Where to save the output from RIFT.

Double check to ensure all of the fields are correct.

NOTE: Do not change the following in the command line. The program will replace the place holders with the correct information.

- [IP]
- [ADMINID]
- [ADMINPASS]
- [MNTOPTS]
- [SHAREDRV]
- [SHARE]
- [SHAREUSERID]
- [SHAREPASSWD]
- [SAVEDRIVE]
- [SRCBOX]
- [SRCMNT]
- [SHAREDRV]

- 4) Save the cmd.txt file.

Getting Started: Text File Configuration (iplist.txt)

- 1) The iplist.txt file contains the list of network ranges or single IP addresses of which machines to scan. Below are some examples of valid lines for the file:
 #Example of valid IP/ranges
 192.168.1.10
 192.168.5.5-192.168.5.25 admin Go1234DW=
 172.16.13.0/24
 10.0.0.0/16

Note: The program is setup to ignore any lines starting with #, blank spaces, or empty lines. DO NOT a # for comments after the IP address or network range.

Note: The program will ignore the broadcast address for any CIDR notation entered. It calculates the broadcast address automatically. IT WILL NOT be able to figure out the broadcast address for IP ranges like 192.168.5.5-192.168.5.25. For example, if 192.168.5.25 is a broadcast address for the network, do not put it in the range.

- 2) After entering the IP address/range put in the admin account to be used and password. Please use TABs between each field.
For example:
10.0.0.0/16 admin password123
- 3) After all of the IP address and ranges are entered save the file.

Running the Scan

- 1) Ensure all of the altered text files and files are saved to the share.

Note: All of the retrieved data will be saved to the share.

- 2) Run the scan:
`frac.exe --iplist iplist.txt --cmd cmd.txt`

The scan runs the just like the Windows version of it.

After the scan

After running the scan, any IP addresses that were not reachable are listed in a file that starts with “unreachableips_” and ends with the date & time of the scan. To hit those IP addresses that were missed, just use the unreachableips text file for the --iplist option.

Running Other Commands with FRAC

Tools Needed

{TOOLS} – You will need to gather any tools that need to be ran into a location where RIFT would have been installed. For the rest of this section the winpmem from Volatility will be used as an example. Also, it may help to create a batch file to run multiple commands.

cmd.txt – This file contains the command that is used to do the work on each machine.

getfiles.txt – Contains a regex list of files and directories that will be retrieved

config.ini – Used to configure how the program runs.

iplist.txt – This file contains the network ranges used to scan for interesting files.

frac.exe – This is the program that will read in the IP ranges and conduct the scans.

Network Share – The network share should be fairly large. It is hard to recommend a size. It all depends on what is being downloaded to the share. Also, the share needs to be mountable by the machines that will be scanned. The author recommends that an admin mount the share by hand to ensure that it will mount correctly.

Paexec – Used to run processes on the remote machines. (Download at [HTTP://www.poweradmin.com/paexec/](http://www.poweradmin.com/paexec/); Source code available at: [HTTPS://github.com/poweradminllc/PAExec](https://github.com/poweradminllc/PAExec))

Winexe – Used to run processes on the remote machines. Try to find a pre-compiled version as winexe can be a bear to compile. (Source code: [HTTP://sourceforge.net/projects/winexe](http://sourceforge.net/projects/winexe))

PYExec.exe – Used to run processes on remote machines. The new versions of Windows 10 work with this and is what is recommended to be used over Paexec.

IMPORTANT: Please insure any processes that reboot the machines happens after the network scan is done. Otherwise critical machines may be skipped during the process of collecting the files.

IMPACT: As with any network scan there is some impact. When a machine is scanned, there will be some disk IO and network impact to the performance of the machine. The disk IO is not high, but some users may notice the hard drive lights staying on longer than normal. Network impact depends on how many and how large the files are that are requested to be pulled back. For example, if pagefile.sys is requested to be pulled back by FRAC, it will effect network performance in that pagefile.sys is a large file. While one to 5 machines may not be too bad in regards to network performance, 20+ machines may adversely affect network performance in the amount of available bandwidth. The author recommends making smart decisions on what is really required to be pulled back for review.

Methodology

The solution uses two possible methods to remotely connect the machines. The methods with pros and cons are listed below:

- **PYExec** – Recommended method as the user will not have any visual windows popping up during the scan.
 - Works best for Windows 10 and new Windows OSes.
 - Pros: Same basic functionality and pros that PAExec has. In addition, PYExec has an encryption option.
 - Cons: Some AV vendors may detect PAExec as malicious as it has been used by actors.
- **Paexec** – Recommended method as the user will not have any visual windows popping up during the scan.
 - Pros: Paexec is freeware and does not require any additional software to be purchased. Also, it can conduct the scan silently. The user will not see any windows pop up during the scan. In addition PAExec will scramble the parameters to protect them from casual wire sniffers, but they are NOT encrypted.
 - Cons: Some AV vendors may detect PAExec as malicious as it has been used by actors.
- **Winexe** – Recommended method as the user will not have any visual windows popping up during the scan.
 - Pros: Winexe is open source and GNU licensed. It runs on Linux or *NIX. Also, it can conduct the scan silently. The user will not see any windows pop up during the scan.
 - Cons: Some type of Linux/*NIX/OSX would be needed to run the scan from. In some environments this may be a problem.
- **PSExec** – PSExec was tested. While it does work, it was decided that PSExec would not work due to the way PSExec works in regards to the system account. In order for FRAC to gather up the files, it requires system level privileges. Using PSExec with the system flag, creates a window the users can disable the triage. By closing down the window or denying access, the user will effective cancel the scan on the box and there is no feedback to FRAC that the scan was canceled.

Creating the Batch File

It is recommended that whatever that needs to be ran is run through a batch file. It makes it easier to run multiple commands at once, and there is a “record” of what was ran. To further the example of using winpmem to capture memory on remote machines,

the batch file shown in Figure 5 was created and called go.bat. It is strongly recommended that the batch file is ran, with all of the tools it requires in place, to ensure it runs correct and the desired output is there.

```
mkdir %computername%_%date:~4,4%%date:~10,2%%date:~7,2%
winpmem_1.4.exe %computername%_%date:~4,4%%date:~10,2%%date:~7,2%\%computername%_%date:~4,4%%date:~10,2%
%date:~7,2%_%time::=%mem
```

Figure 5: Contents Of The go.bat File to Capture Memory

Alter the cmd.txt file

After the batch file has been created and tested, the cmd.txt file has to be altered to run the new command. Figure 6 shows the altered paexec.exe line. Highlighted in yellow is the part that was changed.

```
paexec.exe \\[IP] -n 4 -u [ADMINID] -p [ADMINPASS] -s cmd /C "net use [SHAREDREV] [SHARE] /user:[SHAREUSERID]
[SHAREPASSWD] && cd /d [SAVEDRIVE] && go.bat && net use [SHAREDREV] /delete /yes"
```

Figure 6: Altered line of the cmd.txt file to run the go.bat batch file

Getting Started: Network share

- 1) Create and share out the network share. Please make sure all of the machine that are part of the scan can mount the share. The author recommends that after the share is setup, an administrator mounts the share by hand to ensure it is working correctly.

Note: If your organization wishes to use Samba for the network share that is OK. All of the tools have been tested with Samba.

- 2) Copy all of the files listed in the *Tools Needed* section to the network share.

Getting Started: Text File Configuration (iplist.txt)

- 1) The iplist.txt file contains the list of network ranges or single IP addresses of which machines to scan. Below are some examples of valid lines for the file:

#Example of valid IP/ranges

192.168.1.10

192.168.5.5-192.168.5.25

172.16.13.0/24

10.0.0.0/16

Note: The program is setup to ignore any lines starting with #, blank spaces, or empty lines. DO NOT a # for comments after the IP address or network range.

Note: The program will ignore the broadcast address for any CIDR notation entered. It calculates the broadcast address automatically. IT WILL NOT be able to figure out the broadcast address for IP ranges like 192.168.5.5-192.168.5.25. For example, if 192.168.5.25 is a broadcast address for the network, do not put it in the range.

- 2) After entering the IP address/range put in the admin account to be used and password. Please use TABs between each field.
For example:
10.0.0.0/16 admin password123
- 3) After all of the IP address and ranges are entered save the file.

Running the Scan

- 1) Ensure all of the altered text files and files are saved to the share.

Note: All of the retrieved data will be saved to the share.

- 2) Run the scan:
frac.exe --iplist iplist.txt --cmd cmd.txt

FRAC Impact

The author does not guarantee that the impact on your environment will be the same as outlined in this section. The author recommends that a small subset of machines are tested before using FRAC in the entire environment to determine impact.

For testing impact, the author used Windows 7 64-bit running in VMware Workstation 11 on Fedora 21. Figure 8 shows the computer information captured by the Windows Performance Monitor tool.

Computer Information	
Computer:	WIN-589EGM23G1A
Windows Build:	7601
Processors:	1
Processor Speed:	3500 MHz
Memory:	10807 MB
Platform:	64 Bit

Collection Information	
Start Time:	Friday, July 10, 2015 10:08:30 AM
End Time:	Friday, July 10, 2015 10:14:48 AM
Duration:	378 Seconds
Buffers:	430
Processed Events:	360687
Lost Events:	0
Skipped Events:	18
Use Timing Window:	Yes

Figure 7: Virtual Machine Information

Figure 8 shows the impact on a quiet virtual system. The system was running Windows 7 64-bit and was a virtual machine. The physical machine was running Fedora 21 with VMware Workstation 11.X. Per the statistics it shows that fls.exe and rift.exe has very little impact to the system overall with 81.5% of CPU idle.

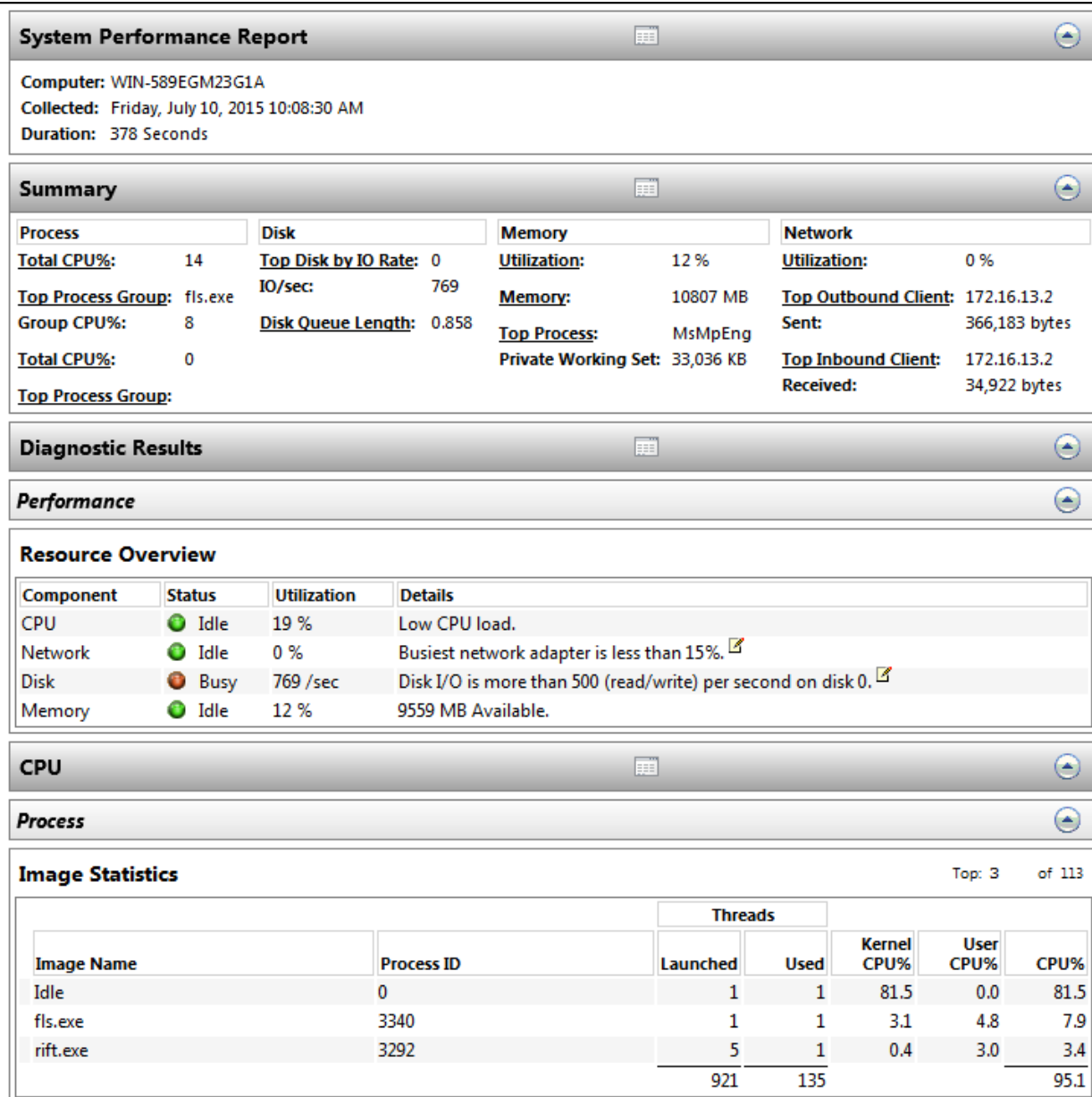


Figure 8: Performance Monitor Stats

Winexe Install on OSX

The easiest way to install Winexe is through brew ([HTTP://brew.sh](http://brew.sh)).

To install bring up a terminal and enter:

```
$ brew install winexe
```

Brew will download and compile Winexe. Currently the version installed is 1.0.

Sleuthkit-APFS Notes

Blackbag Technologies created a fork off of Sleuthkit to parse APFS. While it appears to work good for dead box forensics, it does not completely work for live box forensics. At this time I cannot recommend use of Sleuthkit-APFS with FRAC/RIFT. I figured I'd leave my notes here in case they help someone else.

Brief notes on using APFS. This could be used with FRAC/RIFT. However, there have been some issue observed accessing the entire filesystem with FLS.

1) Need to find the APFS filesystem. Use mmls:

```
$ sudo mmls /dev/rdisk0
Password:
GUID Partition Table (EFI)
Offset Sector: 0
Units are in 512-byte sectors

Slot Start End Length Description
000: Meta 0000000000 0000000000 0000000001 Safety Table
001: ----- 0000000000 0000000039 0000000040 Unallocated
002: Meta 0000000001 0000000001 0000000001 GPT Header
003: Meta 0000000002 0000000033 0000000032 Partition Table
004: 000 0000000040 0000409639 0000409600 EFI System Partition
005: 001 0000409640 0782929919 0782520280 IMAGE_READY
006: 002 0782929920 0783192063 0000262144 Microsoft reserved partition
007: 003 0783192064 1192529919 0409337856 Basic data partition
008: 004 1192529920 1192939519 0000409600 Linux HFS+ ESP
009: 005 1192939520 1193349119 0000409600 Linux HFS+ ESP
010: 006 1193349120 1194373119 0001024000
011: 007 1194373120 1954209791 0759836672
012: ----- 1954209792 1954210119 0000000328 Unallocated
```

From the examples I've seen and my own filesystem. 409640 is the start of the filesystem.

2) Next you need to run the pstat command. This is a new one within the sleuthkit-APFS toolset.

```
$ sudo pstat -o 409640 /dev/rdisk0 | grep "APSB"
| APSB Block Number: 775346
| APSB oid: 1027
| APSB xid: 58929
| APSB Block Number: 778644
| APSB oid: 1030
| APSB xid: 52029
| APSB Block Number: 777074
| APSB oid: 269472
| APSB xid: 58710
| APSB Block Number: 930584
| APSB oid: 310983
| APSB xid: 58560
```

3) Now the fls command can be ran. There is an new option called -B for the APSB Block number. The command looks like this:

```
sudo fls -r -o 409640 -B 775346 /dev/rdisk0
```

You may have to try each of the APSB Block number until you find the right one.

Note: In regards to the APS Block number, keep in mind that it changes every time you run pstat. Also, you cannot use the previous APSB Block numbers. At least this is the behavior I am observing on a live mac.

To put everything in a single command outside of the mmls command:

```
sudo fls -r -l -o 409640 -B `sudo pstat -o 409640 /dev/rdisk0 | grep "APSB" | cut -f2 -d: | head -n 1` /dev/rdisk0
```


“Compiling” FRAC and RIFT

The process below describes how to “compile” the Perl code of FRAC and RIFT into a binary executable. In order to “compile” the code, each of the modules required by the Perl scripts (see the use lines in the code) need to be installed on the box that will be doing the compile. Secondly, the PAR::Packer module needs to be installed. Thirdly, you will need to have CPAN setup and installed on the box before installing any Perl modules.

My FATE tool has an automated Perl module installer bash script that can be used on *NIX boxes. The script can be downloaded at:

[HTTPS://raw.githubusercontent.com/chaoticmachinery/fate/master/install_perl_modules.sh](https://raw.githubusercontent.com/chaoticmachinery/fate/master/install_perl_modules.sh)

To run the “install_perl_modules.sh” script just put a copy of in the FRAC directory and run it. It will go through all the files with a “.pl” extension and pull the modules names out. Then start installing the found Perl modules.

Once the modules have been installed, FRAC/RIFT can be “compiled”. The command line to “compile” the Perl code is:
`pp -o {output filename} {Perl script}`

For example:

`pp -o rift rift.pl`

Please note that “compiling” the scripts for Windows is the same way as it is done for *NIX.

License

FATE, RIFT, and FRAC are GNU GPL v2 licensed. Please see below for a copy of the license.

GNU GENERAL PUBLIC LICENSE Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS