



*Practical operation procedures,
for digital investigators,
using The Open Computer Forensics Architecture*

April 2009

KLPD, Driebergen

Author: J. van der Wal

“Practical Use of OCFA”



Copyright © 2008-2009, KLPD, Driebergen

The content of this document may be used and distributed freely, under the creative commons license, without modification, and for non-profit use only.

“Practical Use of OCFA”

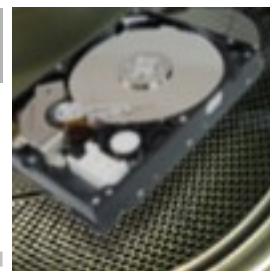


Table of contents

1 Introduction.....	4
1.1 Remarks.....	4
2 Starting a new case.....	5
2.1 Testing before running a case.....	8
2.2 Tuning your storage.....	8
2.3 Tuning your configuration.....	9
2.3.1 Tuning template.conf.....	9
2.3.2 Tuning casename.conf.....	9
2.4 Tuning your rulelist.....	9
2.5 Entering evidence (Kickstarting).....	10
2.5.1 Kicktree.....	10
2.5.2 Single file dd images.....	11
2.5.3 Multi part encase files.....	11
2.5.4 Multi part dd files.....	12
2.5.5 Read only mounted filesystems.....	12
2.5.6 Encase exports.....	13
3 Monitoring a running case.....	14
3.1 When is washing process finished?.....	14
3.2 Stopping a running case.....	14
3.3 Preparing the case for the DUIF frontend.....	15
3.4 Deleting an existing case.....	16
4 Troubleshooting.....	16
4.1 Resubmitting evidence.....	17
5 Abbreviation.....	18

“Practical Use of OCFA”



1 Introduction

This document should help the ocfa-operator to use the Open Computer Forensics Architecture (OCFA). The chapters describe the different task an operator could perform on the OCFA-machine.

1.1 Remarks

Not all aspects of ocfa are described yet. Please feel free to contact the author for suggestions.

This document assumes the usage of **ocfa version 2.2.0** or higher.

“Practical Use of OCFA”



2 Starting a new case

In this chapter, we assume the presence of an installed and well configured ocfa-machine.

1. Make the hostname [*http://casnema.ocfa.loc*] accessible for inspection of results in your local web browser. Only after the step below when the operator has restarted the apache webserver, this url is accessible. There are several ways to do this:
 - A. This could be done by adding the case-url to the local `/etc/hosts`
 - B. The network administrator adds the url to the dns
2. Login with user ocfa to the ocfa-linux-application-server.
3. Give the case name
The user should be prompted to fill in a casename.
If this is the first time the casename is given, the operator gets a hint to use `createcase.pl` as next step.
The case floep does not yet exist, you may want to run `createcase.pl`
4. Create the case
`createcase.pl <casename> <password>`
 - A. Check for existence of `/var/ocfa/<casename>` directory. If it is a huge case, the operator has at this point the opportunity to move the case folder to a SAN or NAS and replace the original location with a symlink.
 - B. Check for existence of database for the case.
[In the “`psql -l`” output list the casename should occur]
5. Make sure no other cases are in progress at the moment.
Check for a running “`casemon.pl`” and kill it,
Stop the case with `ocfahalt.sh <casename>`
Check for any running processes with `top -u ocfa` and kill them.
6. Review “`casename.conf`” file in `$OCFAROOT/etc`. See below the chapter “Tuning the configuration”.
7. Check whether “`/var/ocfa/windows`” direcotory is empty. If not, check the Microsoft Windows server for any remaining case-data. This directory should be empty before starting a new case.
8. Restart apache web service
 - A. `sudo /etc/init.d/apache2 restart`

“Practical Use of OCFA”



- B. Check now your local web browser for existence of the case:
URL: <http://casename.ocfa.loc> (replace the casename with the actual name of the case)
9. Start casemon.pl <casename> script
Because in most cases, the operator uses a remote ocfa-machine with a ssh connection, it is wise to use a “screen session” to run the casemon.pl script in. In the case the connection get lost, casemon will continue and the operator has the opportunity to reconnect to the screen session later. The same holds for the use of the kickstart command. The prompt will not return after the casemon.pl script, until killed. To kill, use the 'CTRL-C'

```
casemon.pl <casename>
```

 - A. Check whether the anycast is alive. [ps -ef | grep anycast]
10. OPTIONAL: Start outlook and outlookexpress modules on the Windows Application Server. Procedures for the Washbrushtools like outlook and outlookexpress are not part of this document. See the Washbrush documentation.
11. Kickstart the evidence
Use the kicktree-command to kickstart evidence into the washing process. It is preferred to use a “screen session” to run the kicktree-command in, because the command could take a while to finish. See the paragraph about “Entering Evidence” for detailed usage information of the kicktree-command or simply enter the kicktree command to get usage information in your terminal.

```
Example:  
kicktree -C testcase -I x13c9d44 -p /mnt/images/x13c9d44.dd
```

 - A. The kicktree command should return “Evidence has been entered”.
Remark: The current automatic generated help message of kicktree pretends to have a “*mmls*” module. This module is not implemented yet.
12. Check in local web browser the queues [ppq-overview]

“Practical Use of OCFA”



Open Computer Forensics Architecture

Home Index Overview PPQ

PPQ Overview

Module	prio 0	prio 1	prio 2	prio 3	prio 4	prio 5	prio 6	prio 7	never
7z	0	0	<u>0</u>	0	0	0	0	0	0
dsm	0	<u>0</u>	<u>0</u>	0	<u>0</u>	0	0	0	0
pgp	0	<u>2</u>	0	0	0	0	0	0	0
pkcr	0	0	<u>0</u>	0	0	0	0	0	0
tar	0	0	<u>0</u>	0	0	0	0	0	0
zip	0	0	<u>4</u>	0	0	0	0	0	0
bzip	0	0	<u>2</u>	0	0	0	0	0	0
file	0	<u>0</u>	0	<u>0</u>	0	0	0	0	0
gzip	0	0	<u>1</u>	0	0	0	0	0	0
pdftotext	0	0	<u>6</u>	0	0	0	0	0	0
antiword	0	0	<u>5</u>	0	0	0	0	0	0
mailwash	0	0	<u>1</u>	0	0	0	0	0	0
digest	0	<u>0</u>	0	0	<u>0</u>	0	0	0	0
sleuthkit	0	0	<u>1</u>	0	0	0	0	0	0
objdump	0	0	<u>3</u>	0	0	0	0	0	0
router	0	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	0	0	0
kickstart	0	0	0	0	0	0	0	0	0
indexer	0	0	<u>11</u>	0	0	0	0	0	0
mailsplit	0	0	<u>2</u>	0	0	0	0	0	0

“Practical Use of OCFA”



2.1 *Testing before running a case*

Two items have to be tested in all cases before starting a real-case:

1. Well configured connection to the windows machine (outlook&outlookexpress)
2. Well installed NIST databases for digest module
3. Enough disk space for repository.
There should be at least two times the amount of the source data be available.
4. The place where the ppq resides. (/var/ocfa/<casename>/ppq)
When the operator chooses to place the case directory on an external storage device like a SAN, the actual place of the ppq (ppq = “persistent priority queue” used by the anycast relay) directory should be on a local storage device. We encountered corrupted persistent queues in the case of storage of the queues on an external device.

2.2 *Tuning your storage*

After you created your case, you should have a directory /var/ocfa/<casename>, for example /var/ocfa/testcase. All the storage for each underlying directory will initially be within the same filesystem, that is, the filesystem that is used for /var/ocfa.

If /var/ocfa is a real big SAN slice that can hold multiple investigations, or if you only have one active investigation at a time, then this would be the right option. If however you use a separate slice for each investigation, you should move the content of your /var/ocfa/<casename> directory to that slice, and use the now empty /var/ocfa/<casename> as mountpoint for that slice.

If for some reason LVM and (software)RAID are no options, you may wish to set up a somewhat slower multi-filesystem repository. In order to build a multi filesystem repository, you should take a number of big filesystems, and populate each with directories with all the possible 2 digit lowercase hexadecimal directory names. Now you should go to your /var/ocfa/<casename>/repository/ directory, and create symbolic links to these directories. You should give the symbolic links the same name as the directories they link to. Please take care to make sure that each filesystem is referenced from a comparable set of symbolic links.

“Practical Use of OCFA”



2.3 Tuning your configuration

After you have created your new case, you will have a configuration file in your `${OCFAROOT}/etc` directory that has the same name as your investigation. This file is created from a standard configuration template, and has some sane values for many configuration parameters. It is still essential for you to validate if you need to update the configuration to your own specific needs.

2.3.1 Tuning template.conf

The `template.conf` in the `${OCFAROOT}/etc` directory contains a template from which the different `casename.conf` files will be created.

Important parameters:

- `storedbhost`
Specify the IP-address of the postgres database server.
- `storedbuser`
Specify the username of the postgres database server.

2.3.2 Tuning casename.conf

The `casename.conf` file resides in the `${OCFAROOT}/etc` directory and is generated from the `template.conf` file.

Please check the following parameters:

- `storedbhost`
Specify the IP-address of the postgres database server.
- `storedbuser`
Specify the username of the postgres database server.

2.4 Tuning your rulelist

Next to your basic config, the standard rulelist may not be fully suitable for your specific

“Practical Use of OCFA”



needs. Chances are that you are especially interested in particular file types and need to use your own modules for that. You should not be afraid to try and create your own modules, especially if you already have tools to convert these files or to extract specific metadata from them. The open computer forensics architecture was designed with the idea of 'wrap-it' development, thus creating a wrapper module around your own tool or library is relatively simple.

If you create your own modules, or have an other reason for wanting other routing than that bundled in the default rulelists, then you will want to create your own rulelist. The best way to do this is by making a copy of an existing rulelist that most matches your need. You could call this rulelist `rulelist_<casename>.conf`, and set a symlink from `/usr/local/digiwash2.0/etc/rulelist.csv` to this rulelist. Make sure to always keep `/usr/local/digiwash2.0/etc/rulelist.csv` a symbolic link. That way you are able to keep a wide range of individually tuned router rulelists for your specific needs.

2.5 Entering evidence (Kickstarting)

This text describes how to process images, mounted filesystems and encase exports within the context of an active investigation. If you do not have an active investigation, please consult the preceding paragraaf. There are different possibilities for inserting image data into the open computer forensics architecture. These possible ways are described in the following paragraphs. A few of them use the `kicktree`-command. So let us start explaining it.

2.5.1 Kicktree

To use the `kicktree`-command, you should first log in or change the current user to “ocfa”, because the `ocfa`-user should run the 'kicktree' command. It is advisable to use a screen session for the `kicktree`-command. The `kicktree` command requires the following set of arguments:

- x `-M --module <modname>` :
Specify the treegraph module to load. (optional). Example:
 - x “e01” is the treegraph module for kickstarting direct encase images.
- x `-C --case <casename>` :
Specify the name of the active investigation. (optional)

“Practical Use of OCFA”



- x `-S --source <srcname>` :
Specify the formal source of the evidence. Typically this will be the identifying name specified on a label or tag attached to the item or box or computer containing the item. (required)
- x `-I --item <itemname>` :
Specify the name of the item. If the label or tag specified by `-S` or `-source` refers to more than one actual items, the `itemname` specifies the individual item its individual name. (optional)
- x `-n --name <nickname>` :
Specify an optional non formal nickname for the item. This nickname is part of the path naming shown in the evidence browser web interface. If no nickname is specified, the actual kick path will be uses.(optinal)
- x `-p --path <path>` :
Specify the path of the item to kickstart. (required).
- x `-a --attribute <k>=<v>`:
Specify a module specific attribute. You can specify multiple attributes this way by repeating the `-a/--attribute` multiple times. (optional)

2.5.2 Single file dd images

One simple way of inserting evidence in the open computer forensics architecture, is the direct copy insertion of dd images. You can insert a dd image by running the `kicktree` command on this image.

Thus an example of a `kicktree` invocation could be:

```
kicktree -C inv124 -S server2 -I hd1 -p /var/img/inv124/serv2hd1.dd
```

Please note that the copy insertion is currently the best tested, but slower than the still experimental `carvpath` kickstart described later on.

2.5.3 Multi part encase files

Both Encase and FTK-Imager can produce multi part encase files. In fact, this is currently

“Practical Use of OCFA”



the preferred way of our digital investigators to make a 1-to-1 copy of seized evidence.

The same way as before, you can use the kicktree-command, but now with the use of the “e01” treegraph module. The command will look like:

```
kicktree -M e01 -C inv124 -S server2 -I hd1 -p /var/disk123/img.e01
```

Remarks: You have only to specify the first encase image of the sequence (e01).

2.5.4 Multi part dd files

With FTK Imager it is also possible to create multi part dd files. To enter this multi part file collection, the operator can use the carvfs tool to build a pseudo dd file. This pseudo dd file can be used with the carvfs tool to construct one pseudo dd ready to use with the kicktree command, as described earlier.

Carvfs

The operator can use the carvfs tool to convert the multipart dd files collection to a single pseudo dd image. This pseudo dd image can be kickstarted as described before.

```
carvfs [-d] <mountpoint> <imgtype> (auto|<digest>) <file> [<file> *]  
imgtype can be one of: "ewf" or "raw"
```

2.5.5 Read only mounted filesystems

While kickstarting image files is a good option, and has the advantage of getting access to unallocated space as separate entities, some filesystems are not recognized by the sleuthkit, and thus can not be processed using the above kickstart methods. If the result of one of the above kickstarts is a fully identified filesystem partition, but the result is only a single big partition file, then changes are that either the sleuthkit does not know how to handle the particular type of filesystem, or the filesystem has been corrupted and can not be read by the sleuthkit for that reason.

A good alternative for unsupported filesystems, but also an excellent alternative if you are not interested in unallocated space (and a long-shot for corrupted filesystems) is using linux itself to mount the filesystem in read only mode. The use of mounted filesystems can be an excellent alternative if metadata from the filesystem is relevant, which it often is. The current sleuthkit module does not add many relevant metadata, and neither does the

“Practical Use of OCFA”



usage of encase exports do so.

The config allows the setting of a flag 'staticmounts' to either true or false. With static mounts on true, the performance of the repository for kickstarting will go up, as files will not get copied to the repository, but symlinks will get used instead. This however means that you can not remove the original image after it has been processed, what may be an option if staticmounts is set to false.

Normally you should mount the filesystem using using the 'uid' and 'gid' flags and both setting them to the id of the ocfa user and the ocfa group respectively. This does mean however that owner metadata will become meaningless. An experimental alternative is adding the root user to the ocfa group and running kicktree as root. This requires setting staticmounts to false, given that symlinks may point to files that ocfa can not access. Please note that kickstart as root is an untested feature in the current release, If you use this feature, please supply us with your (positive or negative) feedback on it.

For kickstarting the mounted filesystem you can simply run something like:

```
kicktree -C inv124 -S server2 -I hd1 /mnt/inv124/server2hd1/
```

By default kicktree will assume the filesystem has its character set as LATIN1. If however, as may be useful when mounting ntfs, you have mounted the filesystem in a way that ensures all directory and file names to be in for example UTF8, you may indicate this by supplying an additional argument indicating the file/dir name encoding used on this mount.

It is important to note that you should not specify UTF8 unless you are sure your filesystem uses unicode and you mounted it UTF8, otherwise the system will throw an exception on the first filename that violates being a valid UTF8 sequence.

2.5.6 Encase exports

For filesystems, especially NTFS filesystems that were found to be corrupted by either the failed usage of the sleuthkit module, or the failed usage of linux mounts, the use of encase as a tool to create directory tree exports becomes quite viable. For this, the 'full image' should get imported into encase, and encase should be used to create an export. It is recommended to export to a samba share on the ocfa linux application server, if possible on the same filesystem that the repository is on. After encase has made the export to the linux system, a special version of kickstart can be used. This kickstart has some knowledge of special filenames that encase uses in its exports for things like unallocated clusters, and it uses this knowledge to add appropriate metadata to allow the router to do

“Practical Use of OCFA”



appropriate routing of these special files.

The 'eekickstart' uses the same arguments as the regular kickstart.

3 Monitoring a running case

1. Checking for errors:
Review `/var/log/ocfa.log` for ERROR's and WARNING's
`[tail /var/log/ocfa.log]`
2. Check for progress:
Review in web browser the PPQ overview. Special attention should be spend to numbers in the “NEVER” column.

3.1 *When is washing process finished?*

The users has to check a view places, to determine whether or not the washing process is finished with the case under investigation:

1. PPQ overview
All values should be zero [0]
Special attention for the never-queue. Messages in this queue indicate the failure of the module for this queue.
2. Working-direcotory
The working directory should be empty
`[du -hs /var/ocfa/casename/work/default]`
3. Windows-directory
The windows directory should be empty
`[du -hs /var/ocfa/windows]`
4. Running processes
Check the activity of the running ocfa processes. All processes should be almost idle.
`[top -u ocfa]`

3.2 *Stopping a running case*

After the operator has checked whether the case has finished, the case can be stopped.

“Practical Use of OCFA”



1. Kill the running casemon.pl script
Check with `ps -ef | grep casemon` if any casemon script is running and kill it
2. Use `ocfahalt <casename> scripg`
3. Check with `top -u ocfa` if all ocfa processes are stopped
4. OPTIONAL
Kill running processes manually.

3.3 Preparing the case for the DUIF frontend

After you have kickstarted all the evidence files, and the open computer forensics architecture has fully processed all data, you will want to prepare your case for usage by the frontend. The procedure for this is currently being reconsidered as there are some problems with this setup, but currently the following needs to be done to fully prepare your case to be used by the frontend effectively. First we should stop the running architecture. For this we should reattach to our casemon.pl screen, and with `ctrl-c` stop its execution. With casemon stoped, the architecture is still running, thus we use `ocfahalt` to stop the architecture.

```
ocfahalt <casename>
```

It should take some seconds before all modules and the anycast have stopped. You can check if anything is still running using:

```
ps -u ocfa
```

At this point comes the “config-tool” in. This is a java base GUI-application to prepare a case for use with the DUIF web frontend. The Configtool will be described in a separate document.

The parkcase tool:

The parkindex tool

Next to tools that optimise the database for basic queries using db indexes, there are currently some essential overview queries that prove to be too time consuming to be

“Practical Use of OCFA”



acceptable in the user interface. At some point we hope to be able to add additional database tables, and a database based user interface to do so, but currently we have a rather underperforming script that creates a static html tree with these overviews. This tool 'makeoverview.pl', should create a large set of overview pages with distinctive metadata values, thumbnail views, calendar views etc. The tool is however not very fast, so we should again run it in the screen session, and only run it as last step, or at any point in time that there are no images being processed.

3.4 *Deleting an existing case.*

First of all make sure there are no more processes running for this case. You may want to shutdown your webserver also to make sure no web access keeps the database locked from being deleted also.

If you want to delete a case, normally all you would have to do is to log in as ocfa and run the ***deletecase.sh*** script.

```
deletecase.sh <casename>
```

There may however be another action required.

If you have kickstarted evidence using the cp-kickstart tool, than you will have active fuse mountpoints in your repository that deletecase.sh does not take kindly to. In that case you should use the following command on each mountpoint.

```
fusermount -u <mountpoint>
```

You can use the `df` or `mount` command to check for mountpoints in your repository.

4 Troubleshooting

“Practical Use of OCFA”



4.1 *Resubmitting evidence*



5 Abbreviation

dsm	data store module.
duif	digital user interface for forensics
Klpd	Korps Landelijke Politiediensten (Netherlands Police Agency)
ocfa	Open Computer Forensics Architecture
ppq	persistent priority queue. Part of the anycast relay.