

Windows Disk Image

- Assumes basic setup from [setup](#)
- All command below should be
 - copied to a “IR log” text/md file
 - edited as needed
 - executed with output copied back to “IR log”
 - recommendation: move commands/output from “useless commands” to second “IR dump” text/md file
- ## is a placeholder for a number

setup

```
echo 'source $HOME/.case.env' >> ~/.bashrc
```

Env variables

```
pwd
WD=/case/working/folder && echo 'WD=$WD' >> ~/.case.env
E=/case/working/folder/evidence/evidence.E01 && echo 'E=$E' >> ~/.case.env
```

Disk Image basics

```
mmls -r $E
sudo -E ~/.local/bin/imount -p -o ## $E
# -p has 'pretty' & predictable folder name, fails if in use
# new tab or ctrl-z
DISK_C=<imount_dir> & echo 'DISK_C=$DISK_C' >> ~/.case.env
```

Start with plaso

```
docker run -it --rm --entrypoint=/bin/bash -v ./:/work log2timeline/plaso
# docker run -it --rm --user :$(id -g) --entrypoint=/bin/bash -v ./:/work
log2timeline/plaso

# one step
psteal.py --source /work/evidence/<filename> -o dynamic,json_line -w
/work/data/plaso_#.json

# two step
log2timeline.py --storage-file /work/data/timeline.plaso
/work/evidence/<filename>
psort.py -o json_line -w /work/output/plaso_out.json
/work/data/timeline.plaso
```

```
#psort.py -o json_line -w /work/output/plaso_out.json  
/work/data/timeline.plaso
```

Extended disk image analysis

```
fsstat -o <offset> $E  
istat -o <offset> $E 5 # root node  
istat -o <offset> $E <inode from fsstat>  
  
fls -o <offset> -m C: -r $E > data/bodyfile  
mactime -b data/bodyfile -d -z UTC yyyy-mm-ddThh:mm:ss >  
output/disk_timeline.csv  
  
mactime -b data/bodyfile -d -z UTC yyyy-mm-ddThh:mm:ss..yyyy-mm-dd >  
output/disk_timeline.csv  
  
# dissect  
  
target-query -f hostname,domain,version,ips,install_date,timezone "$E"  
target-query -j -f services "$E" | jq -r '.name'  
# JSON output → jq  
target-query --list | grep -iE  
'userassist|shimcache|amcache|services|powershell_history|browser.history'
```