

# Análisis Forense en **Linux**



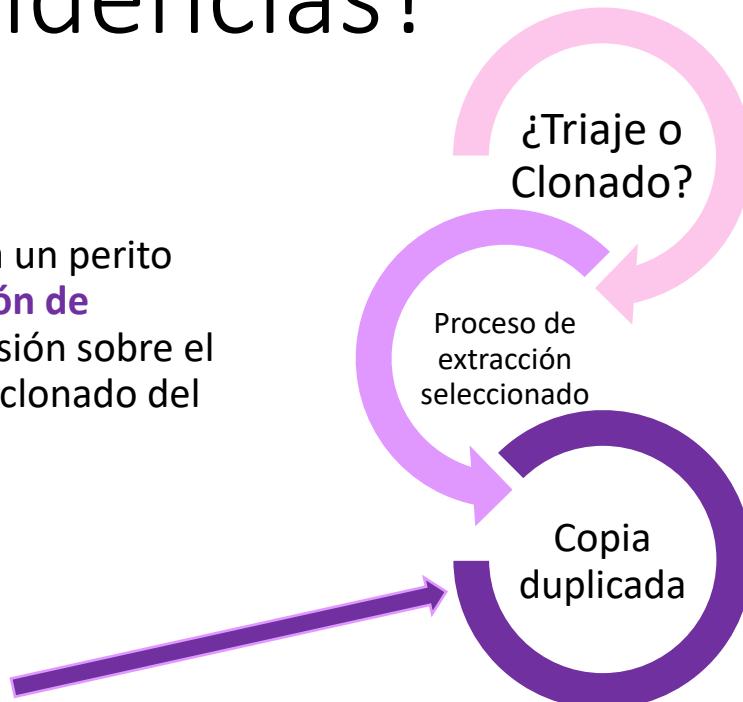
# Metodología:



# ¿Adquisición de evidencias?

En el primer escalón de la metodología que seguirá un perito informático, es decir, durante la fase de “**Adquisición de evidencias**”, el investigador deberá tomar una decisión sobre el modo de adquisición de los datos, optando por un clonado del soporte de almacenamiento o por un triaje.

La copia duplicada será la sometida a análisis por parte del investigador.



## ¿Cómo hacemos un triaje en Linux?

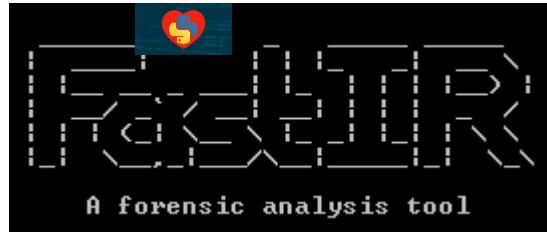
[es.wikipedia.org > wiki > Triage](https://es.wikipedia.org/wiki/Triage)

### Triage - Wikipedia, la enciclopedia libre

El **triaje**, trillaje o cribado (del francés **triage**, "cribado o clasificación"; con la misma etimología que el español trillado, "separación del grano de la paja") o ...

Clasificación de triaje · MANCHESTER TRIAGE... · Manchester Triage System

Al igual que para realizar un **triaje** en S.O de la familia Microsoft Windows, en Linux encontramos diferentes herramientas que nos permitirán realizar el proceso de triaje de manera automática, o al menos, automatizando gran parte de las tareas de extracción. De entre ellas hacemos especial mención a:



```
[*] Copyright Hristyan Lazarov [*]
[!] TriageResponse
[*] Linux Incident Response framework written in bash [*]

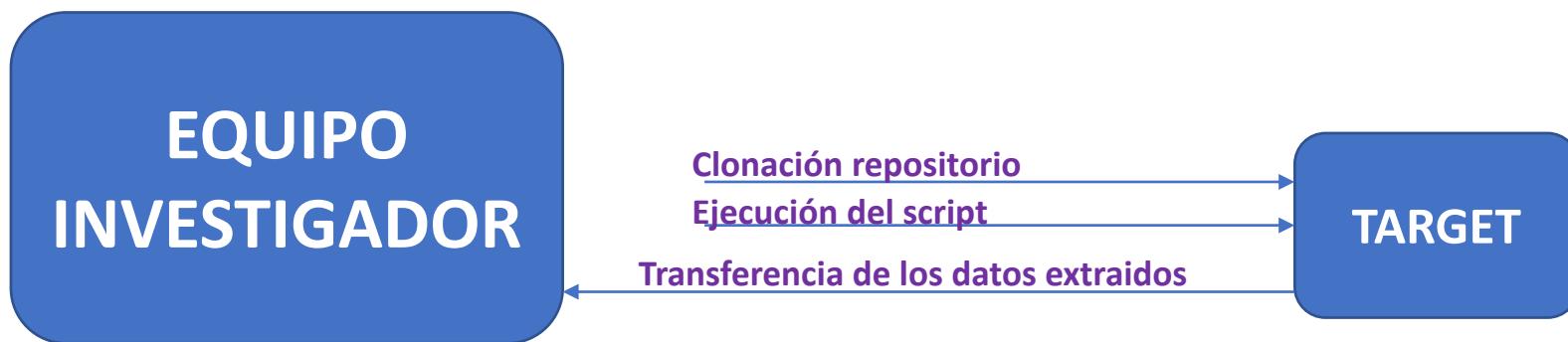
# @localhost # Home/
1) Live response ->           4) Take disk image (DD) ->
2) Connect to target (SSH) ->   5) Generate HTML report
3) Take memory dump (LKM LME)-> 6) Install Software ->
Please enter your choice[1-6, 99 to Quit]:
```



## Demostración Triaje con **FastIR Collector**

**El proceso de triaje tipo mediante FastIR Collector consta de los siguientes pasos**

- Clonación del repositorio FastIR Collector
- Configuración/Ejecución
- Transferencia de datos extraídos



## Clonación del repositorio

Tras cerciorarnos de que disponemos de la herramienta “git” instalada en al máquina pasaremos a clonar el repositorio oficial de FastIR para obtener la herramienta en el sistema target.

Comando:  
git clone  
[https://github.com/SekoiaLab/  
Fastir\\_Collector\\_Linux.git](https://github.com/SekoiaLab/Fastir_Collector_Linux.git)

Fast TIPS: Instalación GIT



sudo apt install git

```
ajfernandez@elasticdeb: ~/forensicTools
File Edit View Search Terminal Help
ajfernandez@elasticdeb:~$ mkdir forensicTools
ajfernandez@elasticdeb:~$ cd forensicTools/
ajfernandez@elasticdeb:~/forensicTools$ git clone https://github.com/SekoiaLab/Fastir_Collector_Linux.git
Cloning into 'Fastir_Collector_Linux'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 31 (delta 0), reused 0 (delta 0), pack-reused 28
Unpacking objects: 100% (31/31), done.
ajfernandez@elasticdeb:~/forensicTools$ ls
Fastir_Collector_Linux
ajfernandez@elasticdeb:~/forensicTools$ ls Fastir_Collector_Linux/
fastIR collector linux.py LICENSE README.md
ajfernandez@elasticdeb:~/forensicTools$
```

## Comprobación de la instalación:

```
ajfernandez@elasticdeb: ~/forensicTools/Fastir_Collector_Linux
File Edit View Search Terminal Help
root@elasticdeb:/home/ajfernandez/forensicTools/Fastir_Collector_Linux# python
fastIR_collector_linux.py -h

[Logo]

A Fast forensic analysis tool

usage: fastIR_collector_linux.py [-h] [--profiles PROFILES]
                                 [--output_dir OUTPUT_DIR] [--dir_zip DIR_ZIP]
                                 [--debug]

FastIR

optional arguments:
  -h, --help            show this help message and exit
  --profiles PROFILES  List of profiles: fast,dump,all use: --profiles fast
                       or --profiles dump --profiles all
  --output_dir OUTPUT_DIR
                        Directory to extract data
  --dir_zip DIR_ZIP    directory to store zip
  --debug              debug level
root@elasticdeb:/home/ajfernandez/forensicTools/Fastir_Collector_Linux#
```

### Fast TIP:

Para la ejecución correcta de esta herramienta hemos de ser o tener permisos de superusuario

```
root@elasticdeb:/home/ajfernandez/forensicTools/Fastir_Collector_Linux# whoami
root
root@elasticdeb:/home/ajfernandez/forensicTools/Fastir_Collector_Linux# python
fastIR_collector_linux.py -h

[Logo]

A Fast forensic analysis tool
```

## Ejecución de triaje

Con permisos de super **ROOT** para la correcta ejecución de la herramienta procedemos a extraer en la carpeta **TriajeLinux\_Demo** la información extraída y generada por FastIR Collector:

**TIP: ERROR sobre “ifconfig -a”**



Puesto que **net-tools** ya se considera desactualizado, debes instalarlo manualmente en caso de querer usarlo con “apt install net-tools”

```
root@elasticdeb:/home/ajfernandez/forensicTools/FastIR_Collector_Linux# python fastIR collector  
linux.py --profiles all --output dir TriajeLinux Demo  
  
[REDACTED]  
  
A Fast forensic analysis tool  
  
2021-03-08 07:37:17,195 - FastIR - INFO : uname -r  
2021-03-08 07:37:17,198 - FastIR - INFO : hostname  
2021-03-08 07:37:17,200 - FastIR - INFO : ifconfig -a  
2021-03-08 07:37:17,205 - FastIR - ERROR : ifconfig -a command failed  
2021-03-08 07:37:17,205 - FastIR - INFO : cat /proc/version  
2021-03-08 07:37:17,206 - FastIR - INFO : who am i  
2021-03-08 07:37:17,211 - FastIR - INFO : lsof -R  
2021-03-08 07:37:17,560 - FastIR - INFO : Write in text file TriajeLinux_Demo/elasticdeb/2021-03-08_073717/handles.txt  
2021-03-08 07:37:17,565 - FastIR - INFO : last -Faixw  
2021-03-08 07:37:17,566 - FastIR - INFO : Write in csv file TriajeLinux_Demo/elasticdeb/2021-03-08_073717/logon.csv  
2021-03-08 07:37:17,566 - FastIR - INFO : lsmod  
2021-03-08 07:37:17,572 - FastIR - INFO : Write in csv file TriajeLinux_Demo/elasticdeb/2021-03-08_073717/modules.csv
```

## Resultados

```
root@elasticdeb:/home/ajfernandez/forensicTools/Fastir_Collector_Linux# ls
fastIR_collector_linux.py LICENSE README.md TriajeLinux_Demo
root@elasticdeb:/home/ajfernandez/forensicTools/Fastir_Collector_Linux# ls TriajeLinux_Demo/
elasticdeb Hostname del target
root@elasticdeb:/home/ajfernandez/forensicTools/Fastir_Collector_Linux# ls TriajeLinux_Demo/elasticdeb/2021-03-08 073717/← Fecha extracción
additional_information.txt FastIR.log logon.csv process.csv users_home.zip
autorun.zip handle.txt modules.csv ss_sockets.csv var_log.zip
root@elasticdeb:/home/ajfernandez/forensicTools/Fastir_Collector_Linux#
```

```
root@elasticdeb:/home/ajfernandez/forensicTools/Fastir_Collector_Linux# cat TriajeLinux_Demo/elasticdeb/2021-03-08 073717/logon.csv
User,Way of connection,Date,Remote host
runlevel,(to lvl 5),Mon Mar  8 06:54:34 2021  still running,0.0.0.0
reboot,system boot,Mon Mar  8 06:54:22 2021  still running,0.0.0.0
shutdown,system down,Mon Feb 22 11:45:00 2021 - Mon Mar  8 06:54:22 2021 (13+19:09),0.0.0.0
ajfernandez,tty2,Mon Feb 22 10:44:11 2021 - Mon Feb 22 11:45:00 2021 (01:00),0.0.0.0
runlevel,(to lvl 5),Mon Feb 22 10:43:56 2021 - Mon Feb 22 11:45:00 2021 (01:01),0.0.0.0
reboot,system boot,Mon Feb 22 10:43:44 2021 - Mon Feb 22 11:45:00 2021 (01:01),0.0.0.0
shutdown,system down,Mon Nov  9 05:50:37 2020 - Mon Feb 22 10:43:44 2021 (105+04:53),0.0.0.0
ajfernandez,tty2,Mon Nov  9 05:48:18 2020 - Mon Nov  9 05:50:36 2020 (00:02),0.0.0.0
runlevel,(to lvl 5),Mon Nov  9 05:48:11 2020 - Mon Nov  9 05:50:37 2020 (00:02),0.0.0.0
reboot,system boot,Mon Nov  9 05:47:58 2020 - Mon Nov  9 05:50:37 2020 (00:02),0.0.0.0
root@elasticdeb:/home/ajfernandez/forensicTools/Fastir_Collector_Linux#
```

# Diferencia System V y SystemD



systemd

# De qué se trata:

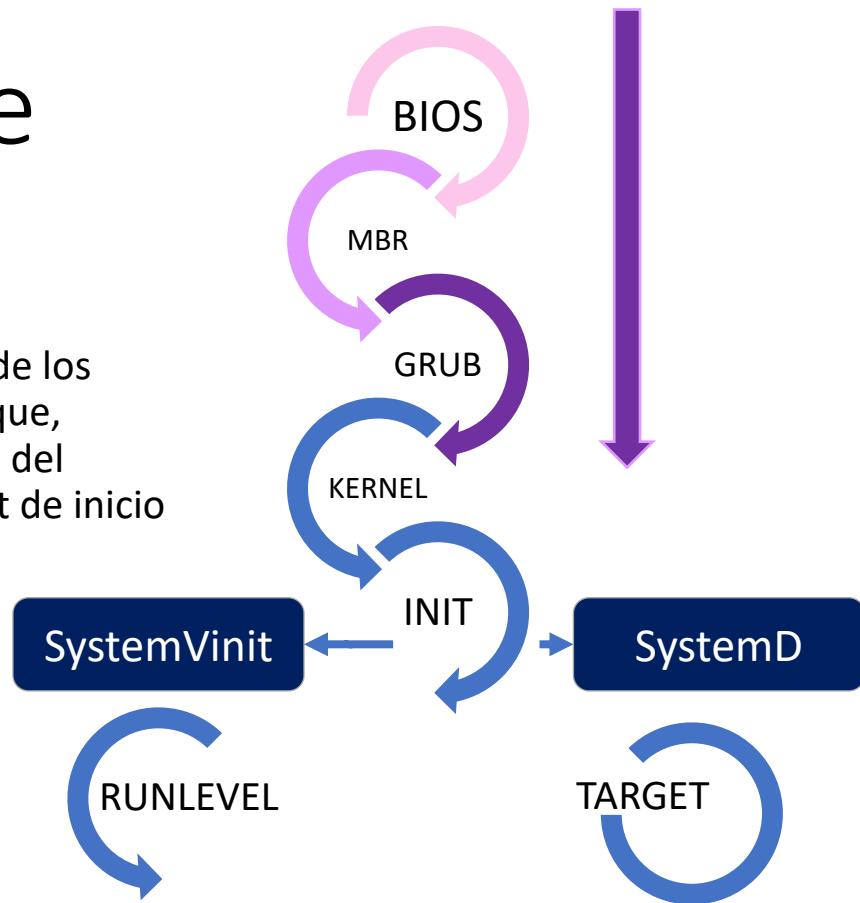
Para comprender la importancia que supone la adopción de uno de estos servicios hemos de comprender las fases por las que pasa un sistema informático con S.O Linux cuando lo arrancamos hasta llegar a la ventana de login.

- BIOS: Interfaz de bajo nivel entre el equipo y sus periféricos. Realiza verificación de integridad de memoria y busca instrucciones en la MBR.
- La MBR enlazará al gestor de arranque, GRUB en la mayoría de casos.
- El GRUB leerá las etiquetas de sistemas operativos disponibles en las cuales se añade información sobre el kernel que usará y el dispositivo de almacenamiento en el cual se encuentra instalado.
- Seleccionado en el GRUB el S.O que se utilizará para el arranque del sistema informático, se cargará el kernel seleccionado y es este último quien cargará el **script de inicio que se encargará de ejecutar los siguientes procesos**, todos dependientes de él, necesarios para el funcionamiento correcto del sistema

...del proceso de arranque, de quién gobernará el resto de procesos, de cómo se interactúa con ellos...

# Proceso de arranque

Una vez cargado el kernel en memoria, y tras la carga de los módulos necesarios para las siguientes fases de arranque, desde el initramfs se montará el filesystem persistente del sistema operativo target y, se pasará el control al script de inicio usado en el S.O



## ¿Cuáles son las diferencias entre ambos Init script?

Se puede resumir en la filosofía adoptada por cada uno de ellos.

**SystemVinit** aboga por un modelo de funcionamiento al más puro estilo UNIX, es decir, “haz una cosa, **una sola cosa**, pero hazla **bien**”.

**SystemD** no solo fue concebido para controlar el orden de prelación entre los servicios o scripts de inicio, sino que ha sido dotado de multitud de nuevas características y funcionalidades como son la gestión de registros, gestión del home en sus últimas versiones, etc.

**TIP:** sobre los registros..



SystemD a, a diferencia de SystemVinit, integra un sistema de registros que no necesita de SYSLOGD, como en el caso de SystemV, para recolectar información, es por esto que es una fuente de datos en fases más temprana.

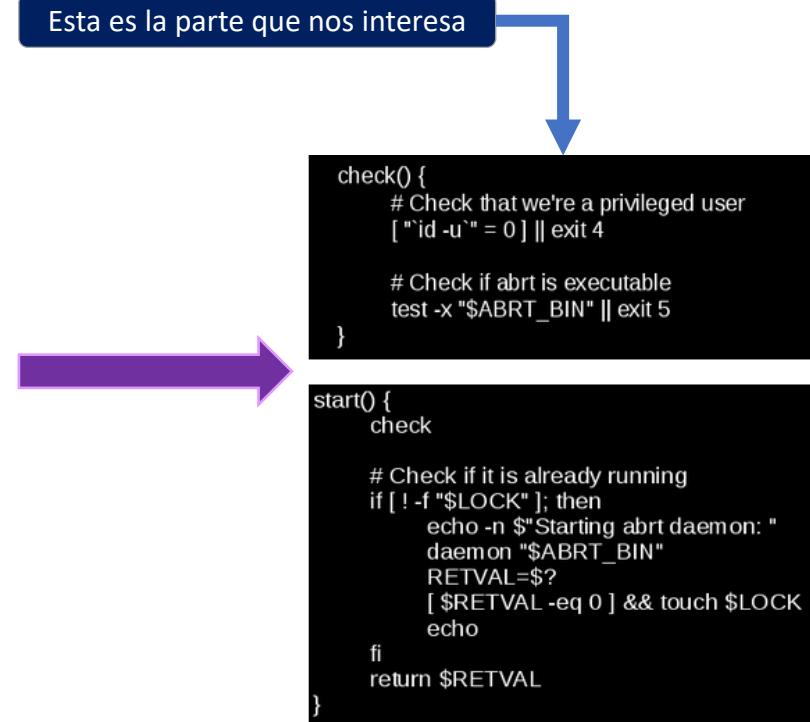
## Veamos un ejemplo **algo más esclarecedor**

### Archivo init script del servicio ABRT

```
#!/bin/bash
# Start the ABRT daemon
#
# chkconfig: 35 82 16
# description: Saves segfault data, kernel oopses, fatal exceptions
# processname: abrt
# pidfile: /var/run/abrt.pid
### BEGIN INIT INFO
# Provides: abrt
# Required-Start: $syslog $local_fs messagebus
# Required-Stop: $syslog $local_fs
# Default-Stop: 0 1 2 6
# Default-Start: 3 5
# Short-Description: Saves segfault data, kernel oopses, fatal exceptions
# Description: Saves segfault data, kernel oopses, fatal exceptions
### END INIT INFO

# Source function library.
. /etc/rc.d/init.d/functions
ABRT_BIN="/usr/sbin/abrtd"
```

Esta es la parte que nos interesa



```
check() {
    # Check that we're a privileged user
    [ "id -u" = 0 ] || exit 4

    # Check if abrt is executable
    test -x "$ABRT_BIN" || exit 5
}

start0 {
    check

    # Check if it is already running
    if [ ! -f "$LOCK" ]; then
        echo -n "$Starting abrt daemon: "
        daemon "$ABRT_BIN"
        RETVAL=$?
        [ $RETVAL -eq 0 ] && touch $LOCK
        echo
    fi
    return $RETVAL
}
```

## ¿Qué hacen estas funciones?

**Check()**{ comprobará que somos un usuario con privilegios de superusuario o root y que el archivo al que apuntamos esté establecido como ejecutable

**Start()**{ comprobará que no haya ninguna instancia anterior del proceso que vamos a ejecutar

Debido a estas líneas, lógicamente replicadas en muchos de los script de inicio dependientes de SystemVinit, los desarrolladores de SystemD se preguntaron por qué dichas acciones debían estar a cargo del script de servicio y no del propio Init.

```
check() {
    # Check that we're a privileged user
    [ "$id -u" = 0 ] || exit 4

    # Check if abrt is executable
    test -x "$ABRT_BIN" || exit 5
}
```

```
start() {
    check

    # Check if it is already running
    if [ ! -f "$LOCK" ]; then
        echo -n $"Starting abrt daemon: "
        daemon "$ABRT_BIN"
        RETVAL=$?
        [ $RETVAL -eq 0 ] && touch $LOCK
        echo
    fi
    return $RETVAL
}
```

## ¿Cómo es un archivo de servicio de SystemD?

```
ajfernandez@jupiter ~ % cd /etc/systemd/system
ajfernandez@jupiter /etc/systemd/system % cat sshd.service
File: sshd.service
1 [Unit]
2 Description=OpenBSD Secure Shell server
3 After=network.target auditd.service
4 ConditionPathExists=!/etc/ssh/sshd_not_to_be_run
5
6 [Service]
7 EnvironmentFile=-/etc/default/ssh
8 ExecStartPre=/usr/sbin/sshd -t
9 ExecStart=/usr/sbin/sshd -D $SSHD_OPTS
10 ExecReload=/usr/sbin/sshd -t
11 ExecReload=/bin/kill -HUP $MAINPID
12 KillMode=process
13 Restart=on-failure
14 RestartPreventExitStatus=255
15 Type=notify
16
17 [Install]
18 WantedBy=multi-user.target
19 Alias:sshd.service
```

Se puede observar de manera rápida como el número de líneas requeridas por un archivo de servicio de **SystemD** es mucho menor que las de un servicio para **SystemVinit** (nótese que las capturas mostradas de **SystemVinit** eran truncadas).

Esto se debe principalmente en que no debemos preocuparnos de verificar si ya se inició el servicio o que, en los servicios que este último se basa, ya estén iniciados, es decir, no debemos crear un orden numérico de prelación para la ejecución de los servicios puesto que...

...lo que definimos para mantener el orden en la ejecución son **DEPENDENCIAS**.

Si **SystemD** nos ofrece más funciones...¿Cómo interactuamos con ellas?

Para ellos SystemD nos ofrece una serie de binarios o utilidades mediante las cuales interactuar con las diferentes funcionalidades que controla:

### **systemd Utilities**

systemctl journalctl notify analyze cgls cgtop logindctl nspawn

# Utilidades SystemD

**systemctl**: herramienta principal de control de los servicios.

**journalctl**: control del estado de registros.

**notify**: notificación entre procesos.

**analyze**: evaluación del sistema de arranque.

**ccls**: obtención de información sobre cgroups según su pertenencia a grupos.

**cgtop**: obtención de información sobre cgroups según consumo de recursos.

**loginctl**: información sobre los inicios de sesión de usuarios.

**nspawn**: herramienta principal de control de jaulas mediante namespaces.

## systemd Utilities

systemctl journalctl notify analyze ccls cgtop loginctl nspawn

## Utilidades SystemD

```
ajfernandez@jupiter ~ $ systemd-analyze blame
 3.352s apt-daily.service
 3.020s apt-daily-upgrade.service
 3.019s nginx.service
 2.330s cloud-config.service
 2.154s docker.service
 1.820s cloud-init-local.service
 1.611s fail2ban.service
 1.047s cloud-init.service
 930ms dev-vdal.device
 823ms ufw.service
 772ms cloud-final.service
 763ms certbot.service
 506ms apache2.service
 368ms mariadb.service
 249ms systemd-journald.service
 248ms ntp.service
 231ms ssh.service
 224ms systemd-udev-trigger.service
 177ms openvpn@client.service
 161ms systemd-logind.service
 155ms rsyslog.service
 124ms netfilter-persistent.service
 90ms systemd-remount-fs.service
 87ms networking.service
 67ms systemd-udevd.service
 58ms containerd.service
 54ms systemd-tmpfiles-setup-dev.service
```

```
ajfernandez@jupiter ~ $ systemctl cgroups
Control group /:
└─.slice
  └─user.slice
    └─user-1001.slice
      └─session-3050.scope
        └─19513 sshd: ajfernandez [priv]
          ├─19526 sshd: ajfernandez@pts/0
          ├─19528 -zsh
          └─19991 systemd-cgls
            └─19992 systemd-cgls
              └─user@1001.service
                └─init.scope
                  ├─19519 /lib/systemd/systemd --user
                  └─19520 (sd-pam)
                └─init.scope
                  └─1 /sbin/init
                └─system.slice
                  └─fail2ban.service
                    └─1083 /usr/bin/python3 /usr/bin/fail2ban-server -s /var/run/fail2ban/fail2ban.sock -p /var/run/fail2ban/fail2ban.pid -x -b
                  └─containerd.service
                    └─953 /usr/bin/containerd
                      └─do-agent.service
                        └─900 /opt/digitalocean/bin/do-agent -log_syslog
                  └─dbus.service
```

```
ajfernandez@jupiter ~ $ logind session-status
3050 - ajfernandez (1001)
  Since: Fri 2021-03-12 09:20:33 CET; 1h 37min ago
  Leader: 19513 (sshd)
  Remote: 5
  Service: sshd; type tty; class user
  State: active
  Unit: session-3050.scope
    └─19513 sshd: ajfernandez [priv]
      ├─19526 sshd: ajfernandez@pts/0
      ├─19528 -zsh
      └─20065 logind session-status
        └─20066 logind session-status
```

## ¿Dónde encontramos SystemD?

En la actualidad son múltiples las distribuciones que usan SystemD como Init por defecto, entre ellas, y como máximo exponente de sus impulsores encontramos a RHEL y Debian.

- Debian GNU/Linux desde la versión 8.
- Fedora 15.
- Frugalware 1.5.
- Mageia desde la versión 2.
- Mandriva desde 2011.
- openSUSE 12.1.
- Arch Linux desde octubre de 2012.
- RHEL desde la versión 7
- CentOS 7 desde julio de 2014.
- Ubuntu desde su versión 15.04 en 2015.



### TIP: ...detractores constructivos..

SystemD ha generado gran controversia en el panorama GNU/Linux, fruto de la cual, los detractores de la adopción de SystemD como init han desarrollado forks de sus distros favoritas manteniendo SystemVinit como script de arranque.

# Análisis de la memoria RAM



# Contexto: investigación FORENSE

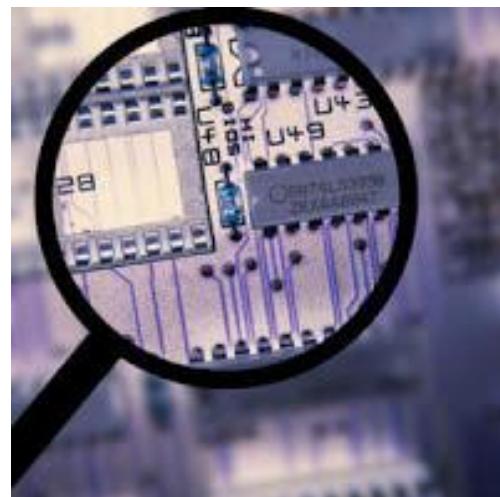
Debido a las características funcionales de la memoria RAM sabemos que una vez se apaga el equipo informático target u objeto de análisis **se perderá** toda la información contenida en la memoria RAM.



- Se creará un dump o volcado completo de la memoria RAM usada en un equipo informático activo hacia un archivo en un soporte de **almacenamiento persistente** como un disco duro o un pendrive USB.

# Es de **VITAL** importancia

- Detección de eventos que no dejan registros en disco.
- Mapa de procesos en ejecución.
- Archivos abiertos por los procesos.
- Existencia de malware en ejecución.



# ¿Adquisición de evidencias?

El investigador o perito encargado de realizar un análisis forense de la memoria RAM de un dispositivo informático podrá realizar la investigación *in situ*, trabajando directamente con los datos aún contenidos en la memoria RAM o bien a través de un volcado o dump obtenido con anterioridad al análisis con las condiciones del sistema informático sin alterar.

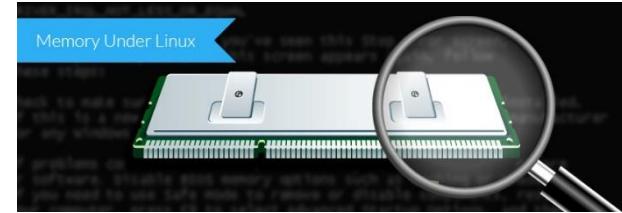
En I.R podemos encontrar casos que por la celeridad del proceso requieran que se realice un análisis *in situ*

...sin embargo...

Como **norma general** se intentará trabajar siempre con una imagen forense o volcado

Fast TIPS: ¿¿¿ I.R ???  
**(I)ncident (R)eponse**

## ¿Cómo hacemos un volcado de RAM en Linux?



La comunidad de peritos informáticos tienen a sus disposición diferentes herramientas de volcado de datos RAM, desde herramientas comerciales (Mandiant Memoryze o MoonSols) a herramientas libres y multiplataforma como Lime, Volatility o Rekall.



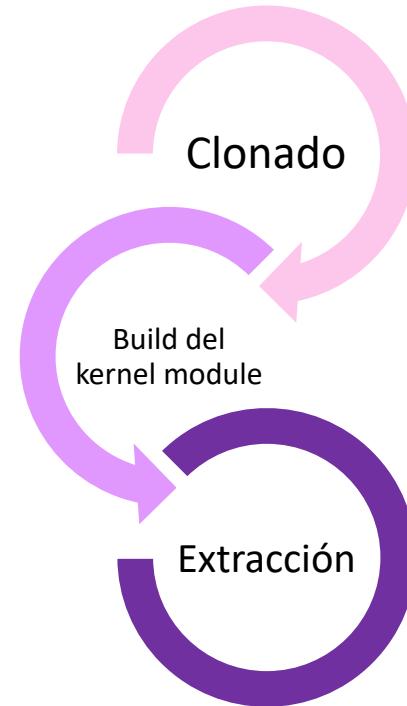
 FIRE EYE™  
Memoryze  
Find evil in live memory

### LiME ~ Linux Memory Extractor

A Loadable Kernel Module (LKM) which allows for volatile memory acquisition from Linux and Linux-based devices, such as Android. This makes LiME unique as it is the first tool that allows for full memory captures on Android devices. It also minimizes its interaction between user and kernel space processes during acquisition, which allows it to produce memory captures that are more forensically sound than those of other tools designed for Linux memory acquisition.



## Demostración extracción con LiME



**El proceso de extracción tipo mediante LiME consta de los siguientes pasos**

- Clonación del repositorio de LiME
- Compilar el módulo
- Extracción

## Clonación del repositorio

Tras cerciorarnos de que disponemos de la herramienta “git” instalada en al máquina pasaremos a clonar el repositorio oficial de LiME para obtener la herramienta en el sistema target.

Comando:  
git clone  
<https://github.com/504ensicsLabs/LiME.git>

Fast TIPS: Instalación GIT

`sudo apt install git`

```
ajfernandez@elasticdeb:~$ uname -r
4.19.0-12-amd64
ajfernandez@elasticdeb:~$ cd forensicTools/
ajfernandez@elasticdeb:~/forensicTools$ git clone https://github.com/504ensicsLabs/LiME.git
Cloning into 'LiME'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 349 (delta 0), reused 1 (delta 0), pack-reused 344
Receiving objects: 100% (349/349), 1.61 MiB | 733.00 KiB/s, done.
Resolving deltas: 100% (185/185), done.
ajfernandez@elasticdeb:~/forensicTools$ ls LiME/src/
deflate.c disk.c hash.c lime.h main.c Makefile Makefile.sample tcp.c
ajfernandez@elasticdeb:~/forensicTools$
```

# QUANTIKA<sup>14</sup>

## Build para crear el módulo del kernel:

```
ajfernandez@elasticdeb: ~/forensicTools/LiME/src
File Edit View Search Terminal Help
ajfernandez@elasticdeb:~$ cd forensicTools/LiME/src/
ajfernandez@elasticdeb:~/forensicTools/LiME/src$ su
Password:
root@elasticdeb:/home/ajfernandez/forensicTools/LiME/src# make
make -C /lib/modules/4.19.0-14-amd64/build M="/home/ajfernandez/forensicTools/LiME/src" modules
make[1]: Entering directory '/usr/src/linux-headers-4.19.0-14-amd64'
  CC [M]  /home/ajfernandez/forensicTools/LiME/src/tcp.o
/home/ajfernandez/forensicTools/LiME/src/tcp.c: In function 'setup_tcp':
/home/ajfernandez/forensicTools/LiME/src/tcp.c:75:5: warning: ISO C90 forbids mixed declarations and code [-Wdeclaration-after-statement]
    int opt = 1;
    ^
CC [M]  /home/ajfernandez/forensicTools/LiME/src/disk.o
CC [M]  /home/ajfernandez/forensicTools/LiME/src/main.o
CC [M]  /home/ajfernandez/forensicTools/LiME/src/hash.o
CC [M]  /home/ajfernandez/forensicTools/LiME/src/deflate.o
LD [M]  /home/ajfernandez/forensicTools/LiME/src/lime.o
Building modules, stage 2.
MODPOST 1 modules
CC      /home/ajfernandez/forensicTools/LiME/src/lime.mod.o
LD [M]  /home/ajfernandez/forensicTools/LiME/src/lime.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.19.0-14-amd64'
strip --strip-unneeded lime.ko
mv lime.ko lime-4.19.0-14-amd64.ko
root@elasticdeb:/home/ajfernandez/forensicTools/LiME/src#
```

### Fast TIP:

Para la construcción correcta del módulo debemos tener instalado el paquete “build-essential”



```
ajfernandez@elasticdeb: ~/forensicTools
File Edit View Search Terminal Help
root@elasticdeb:/home/ajfernandez/forensicTools# apt install build-essential
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86_64-linux-gnu dpkg-dev fakeroot g++
  g++-8 gcc gcc-8 libalgorithm-diff-perl libalgorithm-diff-xs-perl
  libalgorithm-merge-perl libbasan5 libbinutils libc-dev-bin libc6-dev libcc1-0
  libdpkg-perl libfakeroot libfile-fcntllock-perl libgcc-8-dev libitm1
  liblsan0 libmpx2 libstdc++-8-dev libtsan0 libubsan1 linux-libc-dev make
  manpages-dev
Suggested packages:
  binutils-doc debian-keyring g++-multilib g++-8-multilib gcc-8-doc
  libstdc++-8-dbg gcc-multilib autoconf automake libtool flex bison gdb
  gcc-doc gcc-8-multilib gcc-8-locales libgcc1-dbg libgomp1-dbg libitm1-dbg
  libatomic1-dbg libasan5-dbg liblsan0-dbg libubsan1-dbg libumtx0-dbg
  libumtx2-dbg libquadmath0-dbg libc-doc bzr libstdc++-8-doc make-doc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86_64-linux-gnu build-essential dpkg-dev
  fakeroot g++ g++-8 gcc gcc-8 libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libbasan5 libbinutils
  libc-dev-bin libc6-dev libcc1-0 libdpkg-perl libfakeroot
  libfile-fcntllock-perl libgcc-8-dev libitm1 liblsan0 libmpx2 libstdc++-8-dev
  libtsan0 libubsan1 linux-libc-dev make manpages-dev
0 upgraded, 31 newly installed, 0 to remove and 99 not upgraded.
Need to get 38.9 MB of archives.
After this operation, 155 MB of additional disk space will be used.
Do you want to continue? [Y/n] #
```

## Precauciones previas a la ejecución: carga de módulos del kernel

Antes de proceder a la creación el volcado completa de la memoria RAM hemos de verificar que:

- Dispongamos de espacio suficientes para el archivo generado
- Podamos cargar correctamente el módulo del kernel generado mediante la orden “insmod”

**TIP: conocer espacio libre**



**df -h**

```
ajfernandez@elasticdeb:~/forensicTools/LiME/ramLinux_Demo$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            5.5G   0    5.5G  0% /dev
tmpfs           1.2G  9.3M  1.1G  1% /run
/dev/sdal        486   17G  29G  38% /
tmpfs            5.6G  46M  5.5G  1% /dev/shm
tmpfs            5.0M  4.0K  5.0M  1% /run/lock
tmpfs            5.6G   0    5.6G  0% /sys/fs/cgroup
tmpfs            1.2G  3.5M  1.1G  1% /run/user/1000
/dev/sr0         377M  377M   0  100% /media/cdrom0
ajfernandez@elasticdeb:~/forensicTools/LiME/ramLinux_Demo$
```

**TIP: carga del módulo de kernel creado**



**insmod**

```
root@elasticdeb:/home/ajfernandez/forensicTools/LiME/src# insmod
bash: insmod: command not found
root@elasticdeb:/home/ajfernandez/forensicTools/LiME/src# echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
root@elasticdeb:/home/ajfernandez/forensicTools/LiME/src# export PATH=$PATH:/sbin
root@elasticdeb:/home/ajfernandez/forensicTools/LiME/src# echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/sbin
root@elasticdeb:/home/ajfernandez/forensicTools/LiME/src# insmod
insmod: ERROR: missing filename
root@elasticdeb:/home/ajfernandez/forensicTools/LiME/src#
```

## Ejecución de un volcado de RAM con LiME

Con permisos de superusuario o **ROOT** para la correcta ejecución de la herramienta procedemos a volcar en la carpeta **ramLinux\_Demo** la información completa contenida en la memoria RAM:

**TIP: tipos de extracción**

- Format=raw
- Format=lime
- Format=padded

```
ajfernandez@elasticdeb: ~/forensicTools/LiME/src
File Edit View Search Terminal Help
root@elasticdeb:/home/ajfernandez/forensicTools/LiME/src# ls
deflate.c hash.c           lime.mod.c main.o          Module.symvers
deflate.o hash.o           lime.mod.o Makefile      tcp.c
disk.c   lime-4.19.0-14-amd64.ko lime.o    Makefile.sample  tcp.o
disk.o   lime.h             main.c   modules.order
root@elasticdeb:/home/ajfernandez/forensicTools/LiME/src# insmod lime-4.19.0-14-
amd64.ko "path=/home/ajfernandez/forensicTools/LiME/ramLinux_Demo/demoDump.lime
format=lime"
root@elasticdeb:/home/ajfernandez/forensicTools/LiME/src# NO ERROR IS "OK"
```

**COMANDO:** insmod lime-4.19.0-14-amd64.ko "path=/home/ajfernandez/forensicTools/LiME/ramLinux\_demo.lime format=lime"

```
ajfernandez@elasticdeb:~/forensicTools/LiME/ramLinux_Demo$ ls
demoDump.lime
ajfernandez@elasticdeb:~/forensicTools/LiME/ramLinux_Demo$ du -hs demoDump.lime
12G  demoDump.lime
ajfernandez@elasticdeb:~/forensicTools/LiME/ramLinux_Demo$ grep MemTotal /proc/meminfo
MemTotal:      11568640 kB
ajfernandez@elasticdeb:~/forensicTools/LiME/ramLinux_Demo$
```

## Análisis del dump con Volatility

Requisitos de la herramienta:

```
ajfernandez@elasticdeb: ~/forensicTools/LiME/src
File Edit View Search Terminal Help
root@elasticdeb:/home/ajfernandez/forensicTools/LiME/ramLinux_Demo# python -V
Python 2.7.16 CORRECT VERSION
root@elasticdeb:/home/ajfernandez/forensicTools/LiME/ramLinux_Demo# apt install python-pip
python-setuptools python-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  linux-image-4.19.0-9-amd64
Use 'apt autoremove' to remove it.
The following additional packages will be installed:
  libexpat1-dev libjs-jquery libjs-sphinxdoc libjs-underscore libpython-all-dev
  libpython-dev libpython2-dev libpython2.7-dev python-all python-all-dev python-asn1crypto
  python-cffi-backend python-configparser python-crypto python-cryptography python-dbus
  python-entrypoints python-enums python-gi python-ipaddress python-keyring
  python-keyrings.alt python-pip-whl python-pkg-resources python-secretstorage python-six
  python-wheel python-xdg python2-dev python2.7-dev
Suggested packages:
  python-crypto-doc python-cryptography-doc python-cryptography-vectors python-dbus-dbg
  python-dbus-doc python-enums-doc python-gi-cairo libkf5wallet-bin gir1.2-gnomekeyring-1.0
  python-gdata python-keyczar python-secretstorage-doc python-setuptools-doc
The following NEW packages will be installed:
  libexpat1-dev libjs-jquery libjs-sphinxdoc libjs-underscore libpython-all-dev
  libpython-dev libpython2-dev libpython2.7-dev python-all python-all-dev python-asn1crypto
  python-cffi-backend python-configparser python-crypto python-cryptography python-dbus
  python-dev python-entrypoints python-enums python-gi python-ipaddress python-keyring
  python-keyrings.alt python-pip python-pip-whl python-pkg-resources python-secretstorage
  python-setuptools python-six python-wheel python-xdg python2-dev python2.7-dev
0 upgraded, 33 newly installed, 0 to remove and 98 not upgraded.
Need to get 36.2 MB of archives.
After this operation, 68.9 MB of additional disk space will be used.
Do you want to continue? [Y/n] 
```

 **COMANDO:** apt install python-pip python-setuptools python-dev

```
Setting up python-dev (2.7.16-1) ...
Setting up libpython-all-dev:amd64 (2.7.16-1) ...
Setting up python-entrypoints (0.3-1) ...
Setting up python-all-dev (2.7.16-1) ...
Setting up python-keyring (17.1.1-1) ...
Processing triggers for man-db (2.8.5-2) ...
root@elasticdeb:/home/ajfernandez/forensicTools/LiME/ramLinux_Demo# python -m pip install distorm3==3.4.4
Collecting distorm3==3.4.4
  Downloading https://files.pythonhosted.org/packages/68/11/17cc480c1338bea2a223688fcaa04974d203e3d5223044677c288fe1261d/distorm3-3.4.4.tar.gz (134kB)
    100% |██████████| 143kB 4.4MB/s
Building wheels for collected packages: distorm3
  Running setup.py bdist wheel for distorm3 ... done
  Stored in directory: /root/.cache/pip/wheels/42/ab/15/12b7215b87c6298993b20c5cc4e5c3e3a2aeba6aac54dde99
Successfully built distorm3
Installing collected packages: distorm3
Successfully installed distorm3-3.4.4
root@elasticdeb:/home/ajfernandez/forensicTools/LiME/ramLinux_Demo# 
```



**COMANDO:** python -m pip install distorm3=3.4.4

## Análisis del dump con Volatility

Instalación de la herramienta:

```
ajfernandez@elasticdeb: ~/forensicTools/LiME/src
File Edit View Search Terminal Help
root@elasticdeb:/home/ajfernandez/forensicTools# ls
Fastir_Collector_Linux LiME
root@elasticdeb:/home/ajfernandez/forensicTools# git clone https://github.com/volatilityfoundation/volatility.git
Cloning into 'volatility'...
remote: Enumerating objects: 27411, done.
remote: Total 27411 (delta 0), reused 0 (delta 0), pack-reused 27411
Receiving objects: 100% (27411/27411), 21.10 MiB | 17.52 MiB/s, done.
Resolving deltas: 100% (19758/19758), done.
root@elasticdeb:/home/ajfernandez/forensicTools# chmod +x volatility/vol.py
root@elasticdeb:/home/ajfernandez/forensicTools# ls
Fastir_Collector_Linux LiME [volatility]
root@elasticdeb:/home/ajfernandez/forensicTools#
```



### COMANDOS:

- git clone <https://github.com/volatilityfoundation/volatility.git>
- chmod +x volatility/vol.py

```
ajfernandez@elasticdeb: ~/forensicTools/LiME/src
File Edit View Search Terminal Help
root@elasticdeb:/home/ajfernandez/forensicTools# ls
Fastir_Collector_Linux LiME [volatility]
root@elasticdeb:/home/ajfernandez/forensicTools# mv volatility/ /opt/
root@elasticdeb:/home/ajfernandez/forensicTools# ln -s /opt/volatility/vol.py /usr/bin/vol.py
root@elasticdeb:/home/ajfernandez/forensicTools# echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/sbin
root@elasticdeb:/home/ajfernandez/forensicTools# vol.py --info
Volatility Foundation Volatility Framework 2.6.1

Profiles
-----
```



### COMANDOS:

- mv volatility /opt
- ln -s /opt/volatility/vol.py /user/bin/vol.py
- vol.py --info

## Análisis del dump con Volatility

Creación del perfil:

```
ajfernandez@elasticdeb: ~/forensicTools/LiME/src
File Edit View Search Terminal Help
root@elasticdeb:/opt/volatility/tools/linux# ls
kcore Makefile.enterprise module.dwarf module.mod.c module.o Module.symvers
Makefile module.c module.ko module.mod.o modules.order
root@elasticdeb:/opt/volatility/tools/linux# apt install dwarfdump
Reading package lists... Done
Building dependency tree
Reading state information... Done
dwarfdump is already the newest version (20180809-1).
The following package was automatically installed and is no longer required:
 linux-image-4.19.0-9-amd64
Use 'apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 98 not upgraded.
root@elasticdeb:/opt/volatility/tools/linux# make
make -C /lib/modules/4.19.0-14-amd64/build CONFIG_DEBUG_INFO=y /opt/volatility/tools/linux/modules
make[1]: Entering directory '/usr/src/linux-headers-4.19.0-14-amd64'
  Building modules, stage 2...
MODPOST 1 modules
WARNING: modpost: missing MODULE_LICENSE() in /opt/volatility/tools/linux/module.o
see include/linux/module.h for more information
make[1]: Leaving directory '/usr/src/linux-headers-4.19.0-14-amd64'
dwarfdump -di module.ko > module.dwarf
make -C /lib/modules/4.19.0-14-amd64/build M="/opt/volatility/tools/linux" clean
make[1]: Entering directory '/usr/src/linux-headers-4.19.0-14-amd64'
  CLEAN  /opt/volatility/tools/linux/.tmp_versions
  CLEAN  /opt/volatility/tools/linux/Module.symvers
make[1]: Leaving directory '/usr/src/linux-headers-4.19.0-14-amd64'
root@elasticdeb:/opt/volatility/tools/linux# NO ERROR IS "OK"
```

### COMANDOS:

- apt install dwarfdump
- make

**TIP:** archivo module.dwarf



Se trata del archivo de configuración del Kernel usado en el dump

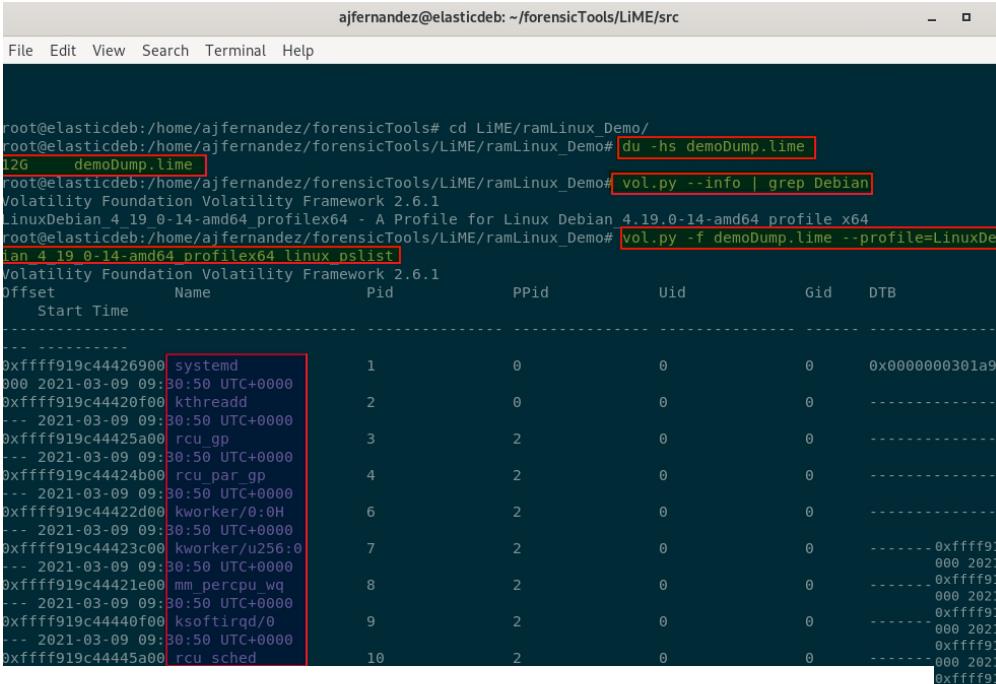
```
ajfernandez@elasticdeb: ~/forensicTools/LiME/src
File Edit View Search Terminal Help
root@elasticdeb:/opt/volatility/tools/linux# zip ${lsb_release -i -s} ${uname -r} profile.zip module.dwarf /boot/System.map-${uname -r}
updating: boot/System.map-4.19.0-14-amd64 (deflated 79%)
adding: module.dwarf (deflated 91%)
root@elasticdeb:/opt/volatility/tools/linux# ls
Debian 4.19.0-14-amd64.profile.zip kcore Makefile Makefile.enterprise module.c module.dwarf
root@elasticdeb:/opt/volatility/tools/linux# cp Debian 4.19.0-14-amd64.profile.zip /opt/volatility/volatility/plugins/overlays/linux/
root@elasticdeb:/opt/volatility/tools/linux# vol.py --info | grep Debian
Volatility Foundation Volatility Framework 2.6.1
LinuxDebian 4.19.0-14-amd64 profilex64 A Profile for Linux Debian 4.19.0-14-amd64 profile x64
root@elasticdeb:/opt/volatility/tools/linux#
```

### COMANDOS:

- zip \${lsb\_release -i -s} \${uname -r} profile.zip ./volatility/tools/linux/module.dwarf /boot/System.map-\${uname -r}
- Cp (archivoZip) /opt/volatility/volatility/plugins/overlays/linux
- Vol.py --info | grep (NombreDistribución)

## Ejemplo - Procesos

A continuación mostramos, mediante el uso del plugins de procesos, los procesos que estaban corriendo en el host target cuando realizamos el dump de la memoria RAM usando **LiME** gracias a nuestro perfil y a **Volatility**:



**COMANDO:**

- `vol.py -f (archivoDump) – profile=(profileCreado) linux_pslist`

Offset	Name	Pid	PPid	Uid	Gid	DTB	Start Time
0xfffff919c44426900	systemd	1	0	0	0	0x00000000301a90	000 2021-03-09 09:30:50 UTC+0000
0xfffff919c44420f00	kthreadd	2	0	0	0	-	-- 2021-03-09 09:30:50 UTC+0000
0xfffff919c44425a00	rcu_gp	3	2	0	0	-	-- 2021-03-09 09:30:50 UTC+0000
0xfffff919c44424b00	rcu_par_gp	4	2	0	0	-	-- 2021-03-09 09:30:50 UTC+0000
0xfffff919c44422d00	kworker/0:0H	6	2	0	0	-	-- 2021-03-09 09:30:50 UTC+0000
0xfffff919c44423c00	kworker/u256:0	7	2	0	0	-	-- 2021-03-09 09:30:50 UTC+0000
0xfffff919c44421e00	mm_percpu_wq	8	2	0	0	-	-- 2021-03-09 09:30:50 UTC+0000
0xfffff919c44440f00	ksoftirqd/0	9	2	0	0	-	-- 2021-03-09 09:30:50 UTC+0000
0xfffff919c44445a00	rcu_sched	10	2	0	0	-	-- 2021-03-09 09:30:50 UTC+0000
0xfffff919c00ac2d00	evolution-alarm	1088	804	1000	1000	0x00000002be8e	000 2021-03-09 09:31:01 UTC+0000
0xfffff919c40edbc00	tracker-store	1089	783	1000	1000	0x00000002b603	000 2021-03-09 09:31:01 UTC+0000
0xfffff919c3988bdc00	fwupd	1133	0	0	0	0x00000002b603	000 2021-03-09 09:31:10 UTC+0000
0xfffff919c39895a00	chromium-terminal	1423	0	0	0	0x00000002b603	000 2021-03-09 09:31:13 UTC+0000
0xfffff919c11e76900	bash	1429	0	0	0	0x00000002b603	000 2021-03-09 09:31:13 UTC+0000
0xfffff919c20a9e900	su	1616	1616	0	0	0x00000002fd91	000 2021-03-09 09:31:24 UTC+0000
0xfffff919c20a9e8f00	bash	1668	0	0	0	0x00000002fd91	000 2021-03-09 09:31:24 UTC+0000
0xfffff919c20a9e900	gvfsd-metadata	2305	783	1000	1000	0x00000002b629	000 2021-03-09 09:37:08 UTC+0000
0xfffff919b0feaa3bc00	firefox-esr	2333	836	1000	1000	0x000000030066	000 2021-03-09 09:50:53 UTC+0000
0xfffff919c421a3c00	unattended-upgr	508	1	0	0	0x0000000301780	000 2021-03-09 09:30:54 UTC+0000
0xfffff919c421a1e00	dbus	509	0	0	0	0x000000030175e	000 2021-03-09 09:30:54 UTC+0000
0xfffff919c410c0f00	sshd	523	0	0	0	0x0000000301322	000 2021-03-09 09:30:54 UTC+0000
0xfffff919c410c3c00	alsactl	528	1	0	0	0x00000003017b8	000 2021-03-09 09:30:54 UTC+0000

**INSTANCIA DEL NAVEGADOR FIREFOX ABIERTA AL MOMENTO DE CREACIÓN DEL DUMP DE MEMORIA RAM**

**SESIÓN SSH INICIADA AL MOMENTO DE CREACIÓN DEL DUMP DE MEMORIA RAM**

## Plugins disponibles:

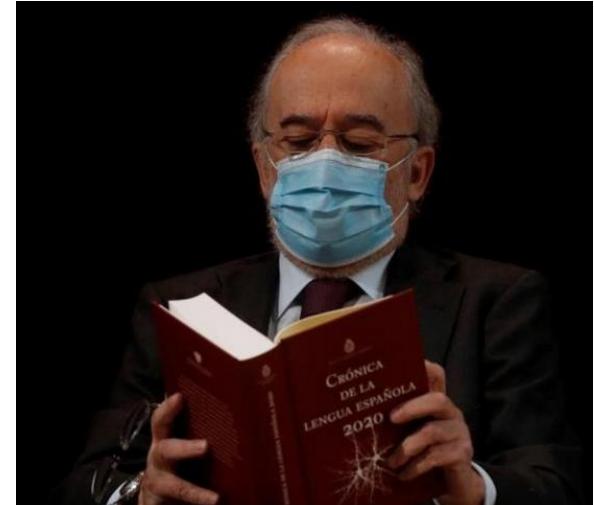
<code>limeinfo</code>	- Dump Lime file format information
<code>linux_apihooks</code>	- Checks for userland apihooks
<code>linux_arp</code>	- Print the ARP table
<code>linux_aslr_shift</code>	- Automatically detect the Linux ASLR shift
<code>linux_banner</code>	- Prints the Linux banner information
<code>linux_bash</code>	- Recover bash history from bash process memory
<code>linux_bash_env</code>	- Recover a process' dynamic environment variables
<code>linux_bash_hash</code>	- Recover bash hash table from bash process memory
<code>linux_check_afinfo</code>	- Verifies the operation function pointers of network protocols
<code>linux_check_creds</code>	- Checks if any processes are sharing credential structures
<code>linux_check_evt_arm</code>	- Checks the Exception Vector Table to look for syscall table hooking
<code>linux_check_fop</code>	- Check file operation structures for rootkit modifications
<code>linux_check_idt</code>	- Checks if the IDT has been altered
<code>linux_check_inline_kernel</code>	- Check for inline kernel hooks
<code>linux_check_modules</code>	- Compares module list to sysfs info, if available
<code>linux_check_syscall</code>	- Checks if the system call table has been altered
<code>linux_check_syscall_arm</code>	- Checks if the system call table has been altered
<code>linux_check_tty</code>	- Checks tty devices for hooks
<code>linux_cpuid</code>	- Prints info about each active processor
<code>linux_dentry_cache</code>	- Gather files from the dentry cache
<code>linux_dmesg</code>	- Gather dmesg buffer
<code>linux_dump_map</code>	- Writes selected memory mappings to disk
<code>linux_dynamic_env</code>	- Recover a process' dynamic environment variables
<code>linux_elfs</code>	- Find ELF binaries in process mappings
<code>linux_enumerate_files</code>	- Lists files referenced by the filesystem cache
<code>linux_find_file</code>	- Lists and recovers files from memory
<code>linux_getcwd</code>	- Lists current working directory of each process
<code>linux_hidden_modules</code>	- Carves memory to find hidden kernel modules
<code>linux_ifconfig</code>	- Gathers active interfaces
<code>linux_info_regs</code>	- It's like 'info registers' in GDB. It prints out all the
<code>linux_iomem</code>	- Provides output similar to /proc/iomem
<code>linux_kernel_opened_files</code>	- Lists files that are opened from within the kernel
<code>linux_keyboard_notifiers</code>	- Parses the keyboard notifier call chain

EXTRACTO LINUX PLUGINS

# Linux Artifacts



# ¿Artifacts?



Los artifacts son todas esas pequeñas piezas de información que se crean fruto de la interacción con un sistema informático durante el transcurso de una vulneración, ataque informático, exfiltración de datos, etc.

# Tipos principales:

A pesar de existir una miríada de *artifacts* en cualquier sistema basado en GNU/Linux estos se agrupan en 2 conjuntos principales:

- **System Artifacts**
  - **Apps Artifacts**



## System Artifacts

¿Cuáles son los logs del sistemas más importantes?:

La mayoría de archivos de registro  
son archivos de texto plano y por  
convención se encuentran en el  
path /var/log

...sin embargo...

*Algunos de estos archivos de  
registros pueden ser movidos a  
diferentes directorios si así lo  
decide el admin. del sistema*

### Fast TIPS: Logs

Algunos de ellos son  
dependientes de la distribución

# System Artifacts

¿Cuáles son los logs del sistemas más importantes?:

/var/log/auth.log: Proporciona un registro de todas las actividades que implican un proceso de autenticación. Por ejemplo registra los usuarios logueados al sistema operativo. Registra el día, hora, usuario y ordenes que se han ejecutado con el comando sudo, los cronjobs que se han ejecutado, los intentos fallidos de autenticación, etc.

/var/log/debug: Para registrar datos de los programas que están actuando en modo depuración. De esta forma los programadores pueden obtener información si sus programas están funcionando adecuadamente.

/var/log/syslog: Contiene la totalidad de logs capturados por rsyslogd. Por lo tanto en este fichero encontraremos multitud logs y será difícil de consultar y filtrar.

# System Artifacts

¿Cuáles son los logs del sistemas más importantes?:

/var/log/kern.log: Proporciona información detallada de mensajes del kernel. Por ejemplo si habéis compilado un kernel y tenéis problemas podréis ver los mensajes de error y advertencias en kern.log. También puede ser útil para intentar detectar y solucionar problemas con la detección de hardware.

/var/log/lastlog: Contiene información sobre la fecha y la hora en que cada usuario se ha conectado por última vez.

/var/log/cron: Registra la totalidad de información de las tareas realizadas por cron.

## App Artifacts

¿Cuáles son los logs de aplicaciones más importantes?:

Por el contrario que ocurría en los *system artifacts*, aquí dependerá de las aplicaciones/servicios instalados u ofrecidos en el sistema informático target, sin embargo, tomando un sistema tipo, podemos tener en cuenta los siguientes apps artifacts que con mayor probabilidad encontraremos en el sistema:

Artifacts de webservers (apache/nginx):

/var/log/[apache | nginx]

```
user nginx nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    server {
        listen 80;
        server_name example.com www.example.com;
        root /usr/share/nginx/html;
        index index.html;
        access_log /var/log/nginx/access.log;
    }
}
```

Podemos personalizar el path para ajustarlo a nuestras necesidades

# App Artifacts

¿Cuáles son los logs de aplicaciones más importantes?:

Fail2ban es uno de los sistemas más implementados en entornos administrados mediante SSH en los que no se deniega el acceso por contraseña y tiene un log/artifacts en el cual son reflejadas todas las acciones que lleva a cabo el servicio:

## Artifacts de Fail2ban:

/var/log/fail2ban.log

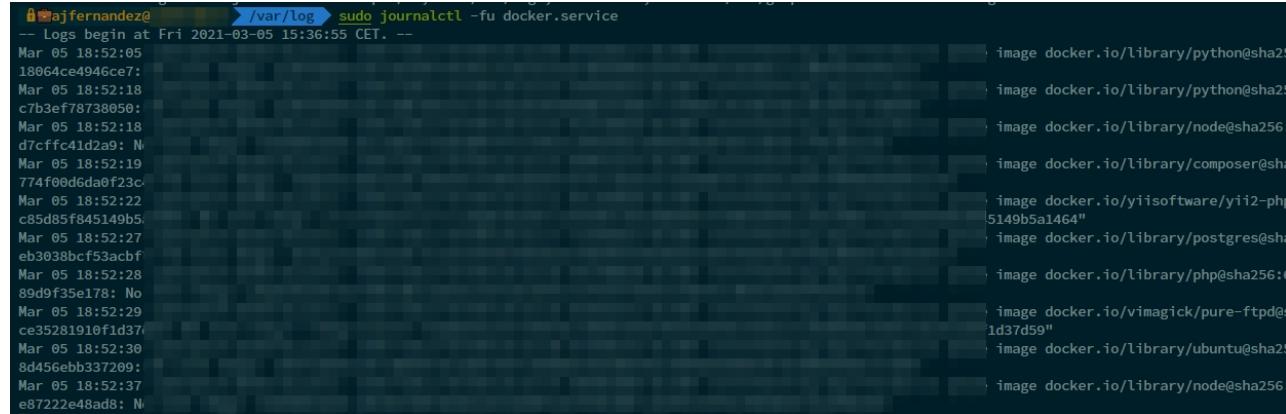
# App Artifacts

¿Cuáles son los logs de aplicaciones más importantes?:

Hoy en día es difícil no encontrar un servidor que no funcione como anfitrión para Docker containers, es por esto que cobran especial relevancia los logs/artifacts de este servicio que, dependiente de la distribución usada, se puede encontrar en:

Artifacts de Docker:

- Debian: /var/log/daemon.log
- RHEL: /var/log/messages



A screenshot of a terminal window titled 'majfernandez' showing Docker logs. The command used is 'sudo journalctl -fu docker.service'. The logs begin at Fri 2021-03-05 15:36:55 CET. The output lists several log entries, each with a timestamp, a SHA-256 hash, and a brief message. To the right of the log entries, there is a vertical list of Docker images, each preceded by a small blue square icon.

```
majfernandez@... /var/log ]# sudo journalctl -fu docker.service
-- Logs begin at Fri 2021-03-05 15:36:55 CET. --
Mar 05 18:52:05 18064ce4946ce7:
Mar 05 18:52:18 c7b3ef78738050:
Mar 05 18:52:18 d7cffc41d2a9: N
Mar 05 18:52:19 774fe00d6da0f23c:
Mar 05 18:52:22 c85d85f845149b5:
Mar 05 18:52:27 eb3038bcf53acbf:
Mar 05 18:52:28 89d9f35e178: No
Mar 05 18:52:29 ce35281910fd37:
Mar 05 18:52:30 8d456ebb337209:
Mar 05 18:52:37 e87222e48ad8: N
image docker.io/library/python@sha256:...
image docker.io/library/python@sha256:...
image docker.io/library/node@sha256:...
image docker.io/library/composer@sha256:...
image docker.io/yiisoft/yii2-php51@sha256:...
image docker.io/library/postgres@sha256:...
image docker.io/library/php@sha256:...
image docker.io/vimagick/pure-ftpd@sha256:...
image docker.io/library/ubuntu@sha256:...
image docker.io/library/node@sha256:...
```

## Entonces...¿Solo LOGS?

Definitivamente **NO**, además de los archivos de registro, otros archivos especiales pueden contener trazas de datos/información útil para una investigación forense, por ejemplo, todos los archivos de configuración del sistema/app pueden ayudarnos a crear un contexto rico para nuestra investigación.

	/
<b>bin</b>	Binarios de usuario
<b>boot</b>	Ejecutables y archivos requeridos para el arranque
<b>dev</b>	Archivos de información de todos los volúmenes
<b>etc</b>	Archivos de configuración del sistema y de aplicaciones
<b>home</b>	Directorio personal con las carpetas de usuario
<b>lib</b>	Bibliotecas necesarias para la ejecución de binarios
<b>media</b>	Directorio de montaje de volúmenes extraíbles
<b>opt</b>	Ficheros de aplicaciones externas que no se integran en /usr
<b>proc</b>	Ficheros de información de procesos
<b>root</b>	Directorio personal de superusuario
<b>sbin</b>	Binarios de sistema
<b>srv</b>	Archivos relativos a servidores web, FTP, etc.
<b>sys</b>	Archivos virtuales con información de eventos del sistema
<b>tmp</b>	Directorio de ficheros temporales
<b>usr</b>	Archivos de programas y aplicaciones instaladas
<b>var</b>	Archivos de variables, logs, emails de los usuarios del sistema, etc.

Según el acervo/jerarquía Linux, el directorio **/etc** contendrá los archivos de configuración.

Un "archivo de configuración" es un archivo local que se utiliza para controlar el funcionamiento de un programa; debe ser estático y no puede ser un binario ejecutable.

## ¿Qué información podemos obtener de los archivos de .conf?

A continuación mostramos solo unos ejemplo de información que podemos recabar mediante los artifacts .conf del directorio /etc:

### /etc(exports:

lista de control de acceso a NFS, mantiene una relación de los directorios que compartimos así como de los clientes (y permisos) para los que son accesibles.

### /etc/ftpusers:

lista de control de acceso a FTP, se especifican los usuarios que podrán acceder al servicio así como los hosts desde los cuales podrá realizarse la conexión para cada usuario.

### /etc/hosts:

mantiene información estática sobre relaciones hostname / direccionamiento IP.

### /etc/profile:

establece una configuración uniforme para los inicios de sesión en shell sh aplicable a todos los usuarios del sistema que hagan uso de la misma.

### /etc/crontab:

mantiene un sistema de lanzamiento programado y persistente de scripts, servicios, etc.

```

ajfernandez@jupiter ~ cd /etc
ajfernandez@jupiter /etc cat exports
File: exports
1   /          master(rw)  trusty(rw,no_root_squash)
2   /projects   proj*.local.domain(rw)
3   /usr        *.local.domain(ro) @trusted(rw)
4   /home/joe   pc001(rw,all_squash,anonuid=150,anongid=100)
5   /pub        *(ro,insecure,all_squash)
6   /srv/www    -sync,rw server @trusted @external(ro)
7   /foo         2001:db8:9:e54::/64(rw) 192.0.2.0/24(rw)
8   /build      buildhost[0-9].local.domain(rw)

ajfernandez@jupiter /etc

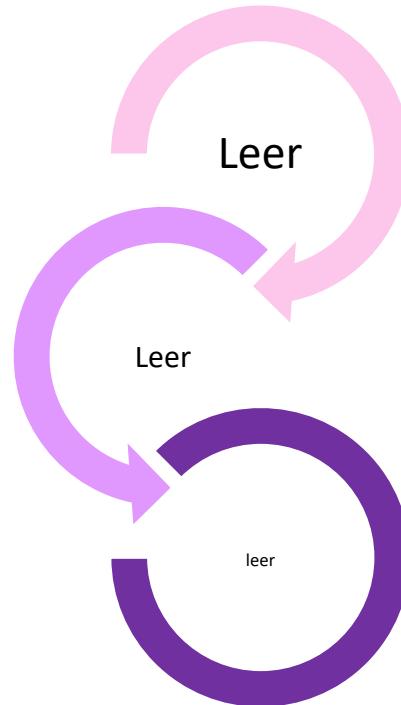
```

## Esto no acaba aquí...

Cualquier archivo de configuración puede convertirse en el artifact perfecto para dar arrojar luz sobre un proceso de investigación forense pericial.

**...muy a tener en cuenta:**

- Archivos ocultos (precedidos por un punto .bashrc)
- Archivo authorized\_host con clave pública
- Archivos de servicio para el init
- etc



# Carving en Linux

# ¿Qué es el file carving?

En el contexto de un análisis forense informático, el file carving es una técnica utilizada para encontrar archivos ocultos o eliminados de soportes de almacenamiento.

Un archivo puede estar oculto en áreas del soporte de almacenamiento como clústeres perdidos, no asignados e incluso en el espacio marcado como "libre" del soporte. Para utilizar este método de extracción o recuperación de datos, un archivo debe tener una marca o firma estándar llamada encabezado de archivo correspondiente al inicio del archivo así como una marca estándar de identificación del final del mismo.

En caso de disponer de la información anterior, podremos acotar los datos raw del archivo a bajo nivel para proceder a la generación o extracción del archivo resultante, como por ejemplo una imagen, un documento Word, etc.



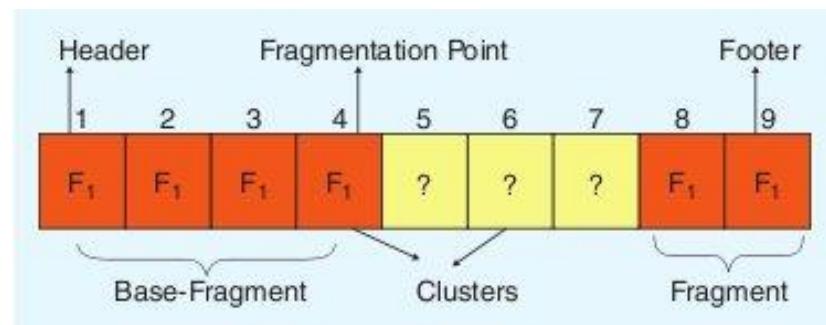
# Conceptos importantes

Header: cabecera estándar de un tipo de archivo específico.

Footer: pie estándar que identifica de manera particular el final de un tipo de archivo.

Bloque: unidad mínima de escritura en el soporte de almacenamiento

Fragmento: conjunto de bloques que forman parte de un mismo archivo.



# Técnicas actuales

**Basadas en identificadores:** el archivo se obtiene tratando los datos incluidos entre un identificador de comienzo (header) y un identificador de final (footer) coincidentes para un mismo tipo de archivo.

**Basado en estructura:** esta técnica, aunque avanzada, es útil para recuperar archivos fragmentados identificando puntos de fragmentación.

**Basado en metadatos:** usando particularidades de algunos sistemas de fichero, como puede ser la tabla MFT en NTFS que aporta metainformación de los archivos que contribuyen a su localización.  
(archivos residentes)

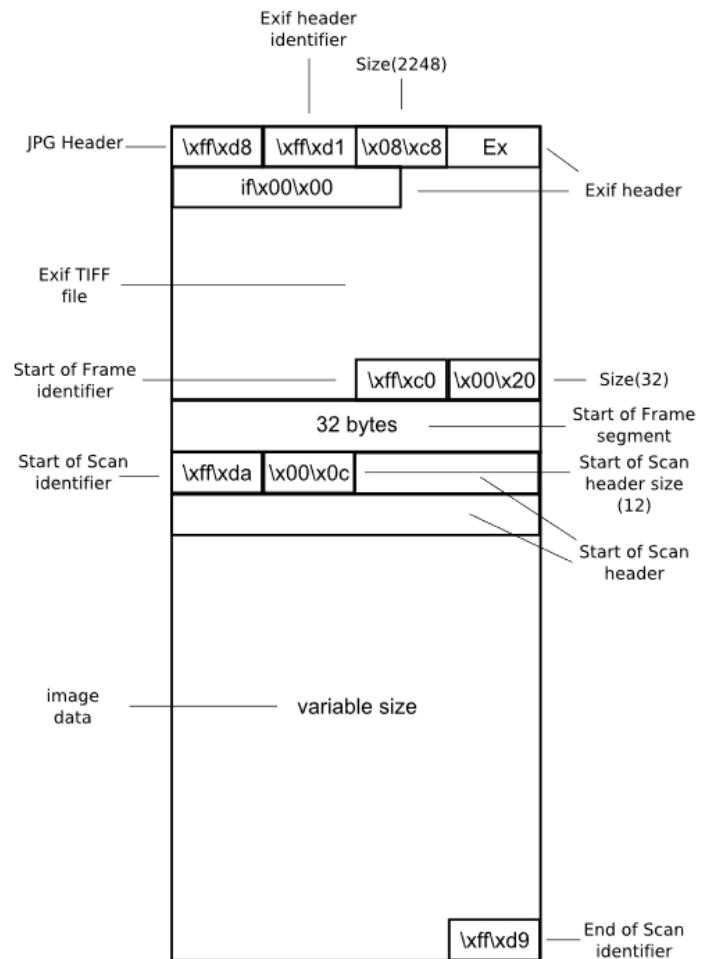
U N Y U U I E F P V K Y O Y  
B T O E O V F I C H E R O J  
F R U F C R O N D G A Z O A  
Q X D T A U P Z C K N E S A  
I A F N R L E E U N V X G E  
N B R H J E Q U U V T I O U  
**F I C H E R O A I B J F L Z**  
I U X T N U U G B X E E U Y

## ...Well known...cabeceras y footers:

Estos son algunos de los pares header-footer más conocidos y útiles:

<b>DOC</b> 	<b>D0 CF 11 E0 A1 B1 1A E1</b>	<b>57 6F 72 64 2E 44 6F 63 75 6D 65 6E 74 2E</b>	<b>JPG</b> 	<b>FF D8 FF E0 00 10 4A 46 49 46 00 01 01</b>	<b>D9 ("Mejor usar el chequeo de tamaño")</b>
<b>XLS</b> 	<b>D0 CF 11 E0 A1 B1 1A E1</b>	<b>FE FF FF FF 00 00 00 00 00 00 00 00 57 00 6F 00 72 00 6B 00 62 00 6F 00 6F 00 6B 00</b>	<b>PDF</b> 	<b>25 50 44 46 2D 31 2E</b>	<b>25 25 45 4F 46</b>
			<b>ZIP</b> 	<b>50 4B 03 04 14</b>	<b>50 4B 05 06 00</b>

**...Well known...estructura fichero:**



# Problemas inherentes al **Carving**:

- Elevado consumo de tiempo
  - Información parcial y/o datos ilegibles
  - Resultados dispares entre herramientas
  - Elevada complejidad en métodos manuales

# Carving

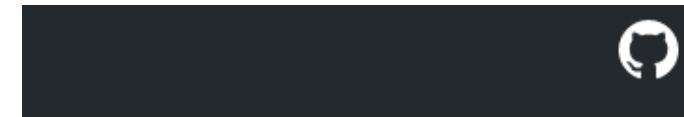


## TOP 3 Tools:

**Scalpel**: realiza operaciones de file carving basándose en patrones, es decir, usa una técnica basada en "estructuras" que describen fragmentos de datos en particular.

**Foremost**: programa de consola para recuperar archivos desarrollado originalmente por la Oficina de Investigaciones Especiales de la Fuerza Aérea de los Estados Unidos y el Centro de Estudios e Investigación de Seguridad de Sistemas de Información y luego abierto al público en general.

**PhotoRec**: programa de recuperación de datos de archivos de diferentes formatos y especializado en documentos ofimáticos y contenido multimedia. PhotoRec ignora el sistema de archivos y busca los datos subyacentes, es decir, se basa nuevamente en el conocimiento de la estructura interna de los archivos.



sleuthkit / scalpel

Code

Issues 27

Pull requests



Page

PhotoRec



Manos a la obra, demostración de carving con...

## Foremost



## Descarga del paquete Foremost

Comando: (Upd. repo)

apt-get update && apt-get install foremost

```
ajfernandez@elasticdeb:~$ su
Password:
root@elasticdeb:/home/ajfernandez# apt update && apt install foremost
Hit:1 http://security.debian.org/debian-security buster/updates InRelease
Hit:2 http://deb.debian.org/debian buster InRelease
Hit:3 http://deb.debian.org/debian buster-updates InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
98 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  linux-image-4.19.0-9-amd64
Use 'apt autoremove' to remove it.
The following NEW packages will be installed:
  foremost
0 upgraded, 1 newly installed, 0 to remove and 98 not upgraded.
Need to get 41.7 kB of archives.
After this operation, 102 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian buster/main amd64 foremost amd64 1.5.7-8 [41.7 kB]
Fetched 41.7 kB in 0s (753 kB/s)
```

## Preparación del entorno:

```
ajfernandez@elasticdeb: ~
File Edit View Search Terminal Help
root@elasticdeb:/home/ajfernandez# cd /media/ajfernandez/FOREMOST
root@elasticdeb:/media/ajfernandez/FOREMOST# lsblk
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda     8:0    0   60G  0 disk
└─sda1  8:1    0  48.7G 0 part /
sda2   8:2    0   1K  0 part
└─sda5  8:5    0 11.3G 0 part [SWAP]
sdb     8:16   1  7.5G  0 disk
└─sdb1  8:17   1  7.5G  0 part /media/ajfernandez/FOREMOST
sr0    11:0    1  377M  0 rom  /media/cdrom0
root@elasticdeb:/media/ajfernandez/FOREMOST#
```

Nuestra unidad de almacenamiento  
será un pendrive USB llamado  
FOREMOS

```
ajfernandez@elasticdeb: ~
File Edit View Search Terminal Help
root@elasticdeb:/media/ajfernandez/FOREMOST# ls
'System Volume Information'
root@elasticdeb:/media/ajfernandez/FOREMOST# cp /home/ajfernandez/index.jpeg .
root@elasticdeb:/media/ajfernandez/FOREMOST# cp /home/ajfernandez/file-sample_10
0kB.doc .
root@elasticdeb:/media/ajfernandez/FOREMOST# cp /home/ajfernandez/file-sample_15
0kB.pdf .
root@elasticdeb:/media/ajfernandez/FOREMOST# ls
file-sample_100kB.doc index.jpeg
file-sample_150kB.pdf 'System Volume Information'
root@elasticdeb:/media/ajfernandez/FOREMOST#
```

Coparemos y, posteriormente  
borraremos, 3 archivos diferentes  
que intentaremos rescatar.

## Visualización página del manual de Foremost

Con permisos de super **ROOT** para la correcta ejecución de la herramienta procedemos a extraer en la carpeta **TriajeLinux\_Demo** la información extraída y generada por FastIR Collector:

**TIP: Uso general**



**Foremost -t (tipo archivo) -o (destino) -i (fuente de datos)**

```
ajfernandez@elasticdeb: ~
File Edit View Search Terminal Help
FOREMOST(8)           System Manager's Manual           FOREMOST(8)

NAME
    foremost - Recover files using their headers, footers, and data structures

SYNOPSIS
    foremost [-h] [-V] [-d] [-vqwQT] [-b <blocksize>] [-o <dir>] [-t
    <type>] [-s <num>] [-i <file>]

BUILTIN FORMATS
    Recover files from a disk image based on file types specified by the
    user using the -t switch.

    jpg          Support for the JFIF and Exif formats including implementations
                 used in modern digital cameras.

    gif
    png
    bmp          Support for windows bmp format.
```

## Recuperación de archivos

Trataremos de recuperar los archivos previamente copiados y eliminados de la unidad USB.

Para ello crearemos una imagen mediante dd de la unidad de almacenamiento de datos.

Comando Imagen:



dd if=(origen) of=(destino)

Comando Foremost:



dd if=(origen) of=(destino)

```
ajfernandez@elasticdeb: ~
File Edit View Search Terminal Help
root@elasticdeb:/media/ajfernandez/FOREMOST# man foremost
root@elasticdeb:/media/ajfernandez/FOREMOST# ls
file-sample 100kB.doc index.jpeg
file-sample 150kB.pdf 'System Volume Information'
root@elasticdeb:/media/ajfernandez/FOREMOST# rm file-sample 100kB.doc file-sample 150kB.pdf index.jpeg
root@elasticdeb:/media/ajfernandez/FOREMOST# ls
'System Volume Information'
root@elasticdeb:/media/ajfernandez/FOREMOST# dd if=/dev/sdb of=/root/foremost_usb.img
15682559+0 records in
15682559+0 records out
8029470208 bytes (8.0 GB, 7.5 GiB) copied, 596.318 s, 13.5 MB/s
root@elasticdeb:/media/ajfernandez/FOREMOST#
```

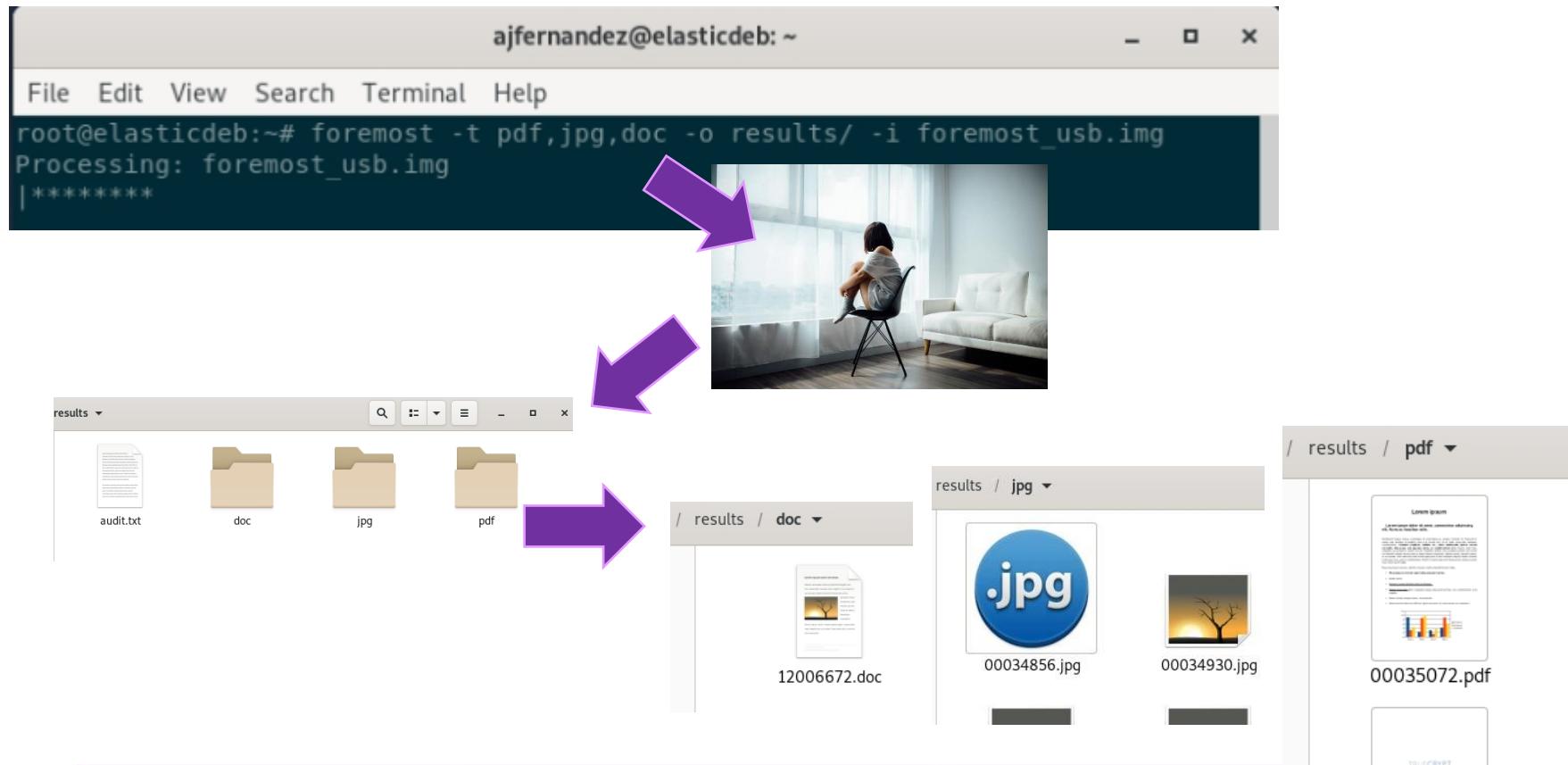
```
15682559+0 records in
15682559+0 records out
8029470208 bytes (8.0 GB, 7.5 GiB) copied, 596.318 s, 13.5 MB/s
root@elasticdeb:/media/ajfernandez/FOREMOST# ls -l /root/foremost_usb.img
-rw-r--r-- 1 root root 7.5G Mar 11 11:33 /root/foremost_usb.img
root@elasticdeb:/media/ajfernandez/FOREMOST#
```

## Resultados

Comando Foremost:



`foremost -t jpg,pdf,doc -o results -i /root/foremost_usb.img`



The screenshot illustrates the process of extracting files from a USB image using the `foremost` tool. The terminal window shows the command being run: `foremost -t jpg,pdf,doc -o results -i /root/foremost_usb.img`. The output indicates the processing of the image file `foremost_usb.img`. Below the terminal, a file browser displays the results. It shows four main folders: `audit.txt`, `doc`, `jpg`, and `pdf`. A large purple arrow points from the terminal output towards a preview of a photograph of a person sitting by a window. Another purple arrow points from the file browser's main view towards a detailed view of the `jpg` folder, which contains files like `12006672.doc`, `00034856.jpg`, and `00034930.jpg`.