Universidad del Cauca

ISO 9001: 2015 SC-CER450832

IQNet: CO- SC-CER450832

Una Acreditación con Rostro Humano

Hacia una Universidad comprometida con la paz territorial

WRITE BLOCKER WITH RASPBERRY PI

Universidad del Cauca

GROUP 1

Luis Felipe Bastidas Torres

Leidy Julieth Sarria Tintinago

Daniel E. Collazos García

Date: 5/08/2019

Electiva informática forense

Outline

Universidad del Cauca

- Background
  - Raspberry Pi 3 model B
  - Kali linux
  - Write Blocker
- Stage 1: automounting
- Stage 2: remote control
- Results

Background

Universidad del Cauca

- Raspberry PI 3 model B.
- Source: https://www.raspberrypi.org/documentation/setup/
- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- 4 USB 2 ports
- Micro SD port for loading your operating system and storing data

Background

Universidad del Cauca

- Kali Linux
- Source: https://www.offensive-security.com/kali-linux-arm-images/
- Kali Linux comes pre-loaded with the most popular open source forensic software, a handy toolkit when you need to do forensic work.

Background

Universidad del Cauca

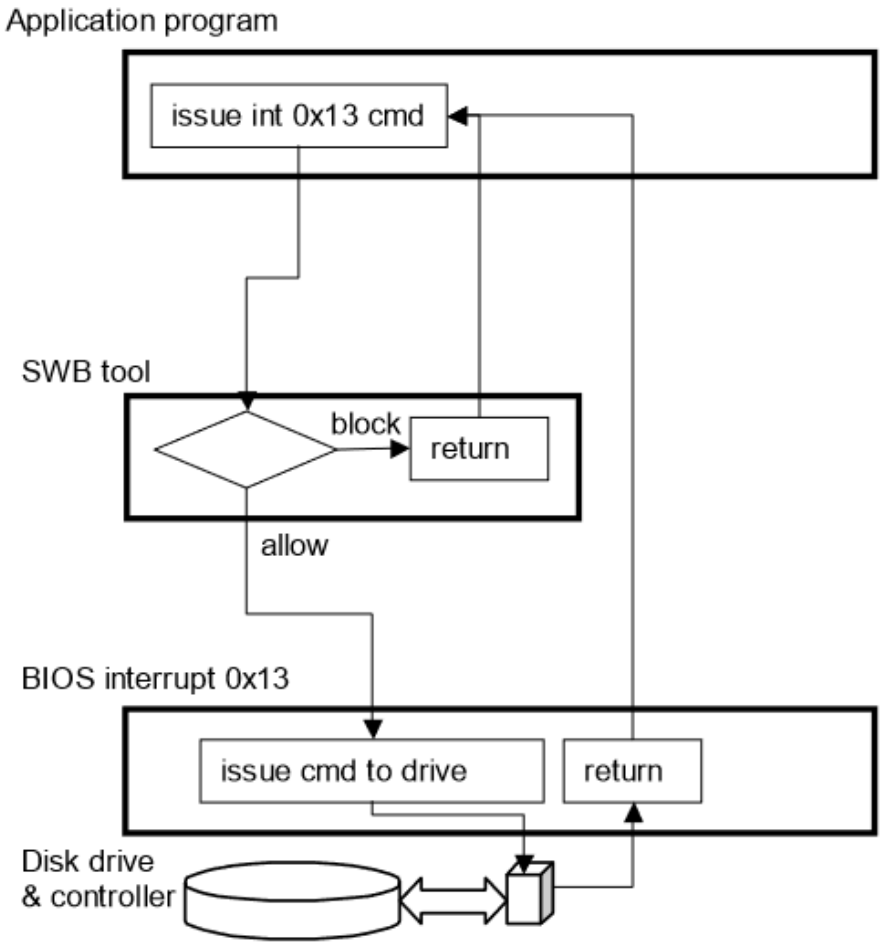
- **Write Blocker** are devices that allow acquisition of information on a drive without creating the possibility of accidentally damaging the drive contents. They do this by allowing read commands to pass but by blocking write commands, hence their name [1]. There are both hardware and software write blockers. Some software write blockers are designed for a specific operating system.

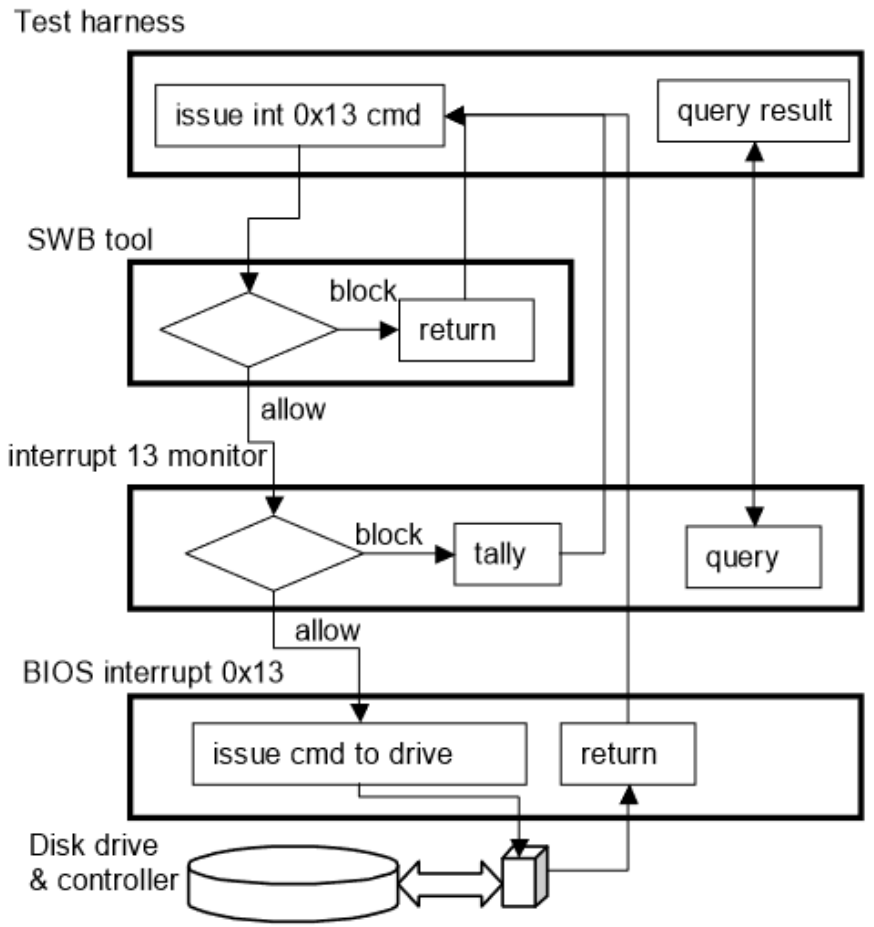
Software Write Blocker

Universidad del Cauca

- Their main upsides are with ease of use, since they are on a CD and do not require you to open the case, and speed since they do not become a bottle neck [1].
- A study of forensic imaging in the absence of write-blockers [2]
- Differences between HWB and SWB [3]
- OpenSource: Linux Software Write Blocker [4]
- Software Write Block Tool Specification & Test Plan [5]

SWB Tool operation





Automounting

Universidad del Cauca

- Preparation
- Previous concepts
- Source
- issues

Preparation

Universidad del Cauca

- Install Kali Linux ARM in Raspberry PI
  - Download Kali linux extension img.xz
  - Unzip with 7zip
  - Software: https://www.balena.io/etcher/ in microSD
  - Insert microSD in Raspberry PI

Previous Concepts

Universidad del Cauca

- Udev: https://www.linux.com/news/udev-introduction-device-management-modern-linux-system
- Blockdev: http://man7.org/linux/man-pages/man8/blockdev.8.html
- Mount: http://man7.org/linux/man-pages/man8/blockdev.8.html

Source

Universidad del Cauca

```
# set disk/device attributes
blockdev --setro "/dev/$bdev" || logger "forensics: blockdev --setro /dev/$bdev failed!"

dd if="/dev/$bdev" of="/home/swb/images/$bdev-${TIME}.dd" conv=sync,noerror bs=1M || logger "forensics: error with dd"
# get hash of image
md5sum /home/swb/images/"$bdev-${TIME}.dd" > "/home/swb/hash/hash-md5sum-$bdev-${TIME}.txt" || logger "forensics: error with md5sum"
# copy image for mount
cp "/home/swb/images/$bdev-${TIME}.dd" "/home/swb/images/copy-$bdev-${TIME}.dd" || logger "forensics: no file in copy proccess"
# get hash of copy images
md5sum /home/swb/images/"copy-$bdev-${TIME}.dd" >>  "/home/swb/hash/hash-md5sum-$bdev-${TIME}.txt" || logger "forensics: error with hash
# mount image copy for forensic activities
mount -o ro,loop,noexec /home/swb/images/"copy-$bdev-${TIME}.dd" "/mnt/$bdev" || logger "forensics: error mounting /dev/$bdev"
```

Modes of script

Universidad del Cauca

- Script iforense2019: set read-only devices, get image, copy image and hashes and mount copy image in /mnt how ro.

```
# get disk image with dd
TIME=`date +%s`
dd if="/dev/$bdev" of="/home/swb/images/$bdev-${TIME}.dd" conv=sync,noerror bs=1M || logger "forens
# get hash of image
md5sum /home/swb/images/"$bdev-${TIME}.dd" > "/home/swb/hash/hash-md5sum-$bdev-${TIME}.txt" || logg
# copy image for mount
cp "/home/swb/images/$bdev-${TIME}.dd" "/home/swb/images/copy-$bdev-${TIME}.dd" || logger "forensic
# get hash of copy images
md5sum /home/swb/images/"copy-$bdev-${TIME}.dd" >>  "/home/swb/hash/hash-md5sum-$bdev-${TIME}.txt"
# mount image copy for forensic activities
mount -o ro,loop,noexec /home/swb/images/"copy-$bdev-${TIME}.dd" "/mnt/$bdev" || logger "forensics:
```

Modes of Script

Universidad del Cauca

- Script iforense2019-3: set read-only devices, nothing more

```
#!/bin/sh
# Part 1. Set disk/device in read mode a
[ $# -eq 1 ] || exit
# get input device
[ ! -z "$1" ] || exit
# assign to bdev
bdev="$1"
# check device how block device
[ -b "/dev/$bdev" ] || exit
# Check variable is not null
[ ! -z ${bdev##loop*} ] || exit
# set disk/device attributes
blockdev --setro "/dev/$bdev" || logger
```

scripting uses

Universidad del Cauca

- Script iforense2019:
  - Mount in read-only and view its content.
  - Issue: save image and copy image in microSD.
  - Issue: same process for all devices that connect.
- Script iforense2019-3:
  - Only set device in read-only mode, nothing more.
  - The python server does the rest of the operations.

Issues

Universidad del Cauca

- Patch to kernel Linux
- Script in bash
- Bad rules in udev

```
# Mark new block devices as read-only.
#ACTION=="add", SUBSYSTEM=="block", KERNEL!="ram*", RUN+="/usr/sbin/iforense2019 $name"
```

- Change rules to:

#ACTION=="add", SUBSYSTEM=="block", KERNEL=="sd*", RUN+="/usr/sbin/iforense2019 $name"

Stage 2: Remote control

Universidad del Cauca

```
from flask import Flask,jsonify
import os
import subprocess
# App of example for remote control
# 2019 forense
# Felipe-Leidy-Daniel
app = Flask(__name__)

@app.route('/devices')
def list_devices():
    cmd = "ls -l /dev/sd* 2>null | wc | awk '{print $1}' > /tmp/server/devices.txt"
    os.system(cmd)
    x = open('/tmp/server/devices.txt','r')
    line = x.readlines()
    x.close()
    if(int(line[0][:-1]) > 0):
        cmd="lsblk | egrep -o sd[^:]* | awk '{print $1}' > /tmp/server/listDevices.txt"
        os.system(cmd)
        x= open('/tmp/server/listDevices.txt','r')
        line = x.readlines()
    else:
        line = [0]
    return jsonify(
        devices=line,
        action="list"
        )
```

Stage 2: Remote control

Universidad del Cauca

- Use of REST with a webservices.

- Create • HTTP PUT
- Read • HTTP GET
- Update • HTTP POST
- Delete • HTTP DELETE

- Create • HTTP PUT
- Read • HTTP GET
- Update • HTTP POST
- Delete • HTTP DELETE

- Raspberry control with Android device!

Operation modes

Universidad del Cauca

when a device is connected to the Raspberry PI it is marked as alone without mounting it. Two routes are established with the Flask Python server. One way saves the process in the microSD, the second way marks the process on a selected device marked in write mode. The files to be saved are image, image copy, hash and result of PhotoRec.

Status defines method: false-> save in MicroSD, true->save in external device.

Form 1: Save in microSD

Universidad del Cauca

- Routes in server: iphost: ip address, porthost: port of server

1. http://iphost:porthost/devices -> connected devices
2. http://iphost:porthost/getImage/<device>/false -> get image, copy image and hash
3. http://iphost:porthost/detailImage/<device>/false -> get metadata of image
4. http://iphost:porthost/photorec/<device>/false -> get deleted files

To see the files inside the image:

* http://iphost:porthost/mount/<device>/false -> mount /mnt
* http://iphost:porthost/umount -> umount /mnt

Form 2: Save in external device

Universidad del Cauca

- First, mount external device in /media:
  - http://iphost:porthost/setrw/<device>/
    - Note: not whole device only a partition
- Second, same routes of form1, change false by true
- Third, unmount external device:
  - http://iphost:porthost/umount -> unmount /media

Results: script iforense2019

Universidad del Cauca

Copy image.

```
root@kali:/home/swb/images# ls
copy-sda-1564971577.dd
```

* Hash of image

```
root@kali:/home/swb/hash# cat hash-md5sum-sda-1564971577.txt
8cf53a32a479d140c842267759b47b18  /home/swb/images/sda-1564971577.dd
root@kali:/home/swb/hash#
```

* Mount copy in /mnt/sda in ro mode

```
root@kali:~# mount | grep /home/
/home/swb/images/copy-sda-1564971577.dd on /mnt/sda type vfat (ro,noexec,relatime,fmask=0
022,dmask=0022,codepage=437,iocharset=ascii,shortname=mixed,errors=remount-ro)
```

* check mounted devices, copy image is mounted with loop devices

```
root@kali:~# df
Filesystem     1K-blocks     Used Available Use% Mounted on
/dev/root       29650404 13370656  14993136  48% /
devtmpfs          464460        0    464460   0% /dev
tmpfs             469964        0    469964   0% /dev/shm
tmpfs             469964      732    469232   1% /run
tmpfs               5120        0      5120   0% /run/lock
tmpfs             469964        0    469964   0% /sys/fs/cgroup
/dev/mmcblk0p1    124738    54564     70174  44% /boot
tmpfs              93992       16     93976   1% /run/user/0
tmpfs              93992        4     93988   1% /run/user/113
/dev/loop0        991956   502884    489072  51% /mnt/sda
```

* block of write commands in loop device

```
root@kali:~# dmesg | tail
[ 1961.873006] FAT-fs (loop0): Directory bread(block 1576787) failed
[ 1961.873023] FAT-fs (loop0): Directory bread(block 1576788) failed
[ 1961.873039] FAT-fs (loop0): Directory bread(block 1576789) failed
[ 1961.873055] FAT-fs (loop0): Directory bread(block 1576790) failed
[ 1961.873071] FAT-fs (loop0): Directory bread(block 1576791) failed
[ 1961.873083] FAT-fs (loop0): error, corrupted directory (invalid entries)
[ 1992.680958] print_req_error: I/O error, dev loop0, sector 1568
[ 1992.680998] Buffer I/O error on dev loop0, logical block 1568, lost async page write
[ 1992.681055] print_req_error: I/O error, dev loop0, sector 3512
[ 1992.681078] Buffer I/O error on dev loop0, logical block 3512, lost async page write
```

Results: script iforense2019-3

Universidad del Cauca

Receiver in Python server to /getImage

```
192.168.3.6 - - [05/Aug/2019 18:36:56] "GET /devices HTTP/1.1" 200 -
192.168.3.6 - - [05/Aug/2019 18:37:20] "GET /devices HTTP/1.1" 200 -
97+1 records in
97+1 records out
1017773056 bytes (1.0 GB, 971 MiB) copied, 81.0377 s, 12.6 MB/s
```

Created image and copy image

```
root@kali:/home/swb/images# ls
copy-sdb.dd  sdb.dd
```

Created hash of image and copy image

```
root@kali:/home/swb/hash# cat hash-sdb.txt
4f9e8fee6865148e3a396a974ebedbd9  /home/swb/images/sdb.dd
4f9e8fee6865148e3a396a974ebedbd9  /home/swb/images/copy-sdb.dd
```

OpenSource

Universidad del Cauca

- Read-only script bash:
  - https://github.com/dancollager/script-linux-swb
- Server for control remote:
  - https://github.com/dancollager/server_rbp
- Script for sanitize:
  - https://github.com/dancollager/SanitizeUSB-script

References

- [1] https://www.forensicswiki.org/wiki/Write_Blockers
- [2] https://commons.erau.edu/jdfsl/vol9/iss3/4/?utm_source=commons.erau.edu%2Fjdfsl%2Fvol9%2Fiss3%2F4&utm_medium=PDF&utm_campaign=PDFCoverPages
- [3] https://github.com/msuhanov/Linux-write-blocker/blob/master/research/2017-01_Write_blockers.pdf
- [4] https://github.com/msuhanov/Linux-write-blocker
- [5] https://www.nist.gov/sites/default/files/documents/2017/05/09/swb-stp-v3_1a.pdf

Universidad del Cauca

¡Gracias por su atención!

Universidad del Cauca

www.unicauca.edu.co