Lessons learned in Forensics

Based on real cases

CIRCL
Computer Incident
Response Center
Luxembourg

Michael Hamm - *TLP:GREEN*

info@circl.lu

Slides updated with Feedback

Modify data on "Read Only" mounted device

- Message in a forensic book

  "Nothing will prevent your Linux system to modify data on a read only mounted device"

- Leads to a new exercise for CIRCL DFIR 1.0.1 training
- I will
  - Targeted tamper of evidences
  - Only use on-board-tools
  - Be `root`
  - Cheat (A little bit)
- I will not
  - Remount the device in RW mode
- Any ideas?

Read Only Exercise: Play Script

1. Identify how the device is connected
2. Review mount options
3. Re-mount the device in "Read Only" mode
4. Review mount options
5. Open file, modify data and try to safe
6. Use `strings` to identify offest of the data
7. Calculate sector number
8. `dd` sector on to local disk
9. Modify local stored sector with a hexeditor
10. `dd` sector back on RO mounted block device
11. Validate results

Read Only Exercise: SCRIPT

```
dmesg
    sd 1:0:0:0: [sdb] Write Protect is off
    sdb: sdb1
    sd 1:0:0:0: [sdb] Attached SCSI removable disk
mount
    /dev/sdb1 on /media/michael/CIRCL-DFIR type vfat (rw,nosuid,

mount -o remount,ro /dev/sdb1 /media/michael/CIRCL-DFIR/
mount
    /dev/sdb1 on /media/michael/CIRCL-DFIR type vfat (ro,relatime,

strings -td /dev/sdb1 | grep Hello
    1050210 Hello World!
echo $(( 1050210 / 512 ))
    2051

dd if=/dev/sdb1 bs=512 skip=2051 count=1 of=2051.raw
ll
    -rw-r--r-- 1 root    root         512 Jan  3 11:20 2051.raw

hexer 2051.raw

dd if=2051.raw bs=512 seek=2051 count=1 of=/dev/sdb1
```

Read Only Exercise: Countermeasures

- Try on board methods:
  - `hdparm -r1 /dev/sdb`
  - `blockdev --setro /dev/sdb`
  - `udev` rules

    → Attack on block device still possible
- Try Forensics Linux Distributions:
  - Live Kali 2018_4 in forensic mode
  - SANS SIFT Workstation 3.0
  - DEFT X 8.2 DFIR Toolkit
    - Some distributions do not auto mount

      → Attack on block device still possible
- Kernel Patch: Linux write blocker (not tested)

  → `https://github.com/msuhanov/Linux-write-blocker`
- Hardware Write Blocker

  → Effectively block attack

Data recovery from damaged ZIP archives

- Ransomware case 2018 before infection
  - Daily full backup into ZIP archive
  - External drive connected full time
  - Backup folder on external drive:

```
2019-01-04.zip
2019-01-07.zip
2019-01-08.zip
2019-01-09.zip
```

- Backups accessible for ransomware
  - All ZIP archives got wiped partially

```
00000000: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000010: 0000 0000 0000 0000 0000 0000 0000 0000  ................
   .....
000001c0: 6d4c 250f 12b0 f1bd ac5a d0fd 0350 4b03  mL%......Z...PK.
000001d0: 0414 0000 0008 00ed 8227 4e2e f396 5bc0  .........'N...[.
000001e0: b304 006e aa05 0018 001c 006c 6f67 6f2d  ...n.......logo-
000001f0: 6369 7263 6c2d 466f 7265 6e73 6963 732e  circl-Forensics.
```

→ Topic not covered in this presentation

Hiding data in HPA

- ATA-3: Hard disk password
- ATA-4: HPA - Host Protected Area
  - Vendor area - benefit system vendors
  - Recovery data. persistent data
  - Controlled by firmware not OS
- ATA-6: DCO - Device Configuration Overlay
  - Benefit system vendors
  - Control reported capacity and disk features
  - Use disk from different manufacturers
  - Use disk with different number of sectors
    $\rightarrow$ Makes disks looking uniq

HPA Exercise: Create hidden data

- New disk

```
dmesg
    sd 1:0:0:0: [sdb] 3907029168 512-byte logical blocks: (2.00 TB/1.82 TiB)

hdparm -N /dev/sdb
    max sectors   = 3907029168/3907029168, HPA is disabled
```

- Create hidden data

```
echo -n 'MySecret 123456' | dd of=/dev/sdb seek=3900000000

dd if=/bin/dd of=/dev/sdb seek=3900000001
    148+1 records in
    148+1 records out
    76000 bytes
```

- Create HPA

```
hdparm --yes-i-know-what-i-am-doing -N p3896789168 /dev/sdb
    setting max visible sectors to 3896789168 (permanent)
    max sectors   = 3896789168/3907029168, HPA is enabled

Reboot disk to apply new settings
```

HPA Exercise: Normal disk usage

- Create partition and format disk

```
fdisk /dev/sdb
    primary
    First sector: default 2048
    Last sector:  default 3896789167

mkfs.ntfs -L CIRCL_DFIR -f /dev/sdb1
    Cluster size has been automatically set to 4096 bytes.
    Creating NTFS volume structures.
    mkntfs completed successfully. Have a nice day.
```

- Investigate disk layout

```
mmls /dev/sdb
             Slot      Start        End          Length       Description
    000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
    001:  ---------  0000000000   0000002047   0000002048   Unallocated
    002:  000:000   0000002048   3896789167   3896787120   NTFS / exFAT (0x07)
```

- Investigate last sector

```
dd if=/dev/sdb skip=3896789167 | xxd
    00000000: eb52 904e 5446 5320 2020 2000 0208 0000  .R.NTFS     .....
          ...
    000001f0: 0000 0000 0000 0000 0000 0000 0000 55aa  ..............U.
```

HPA Exercise: Access hidden data

- Search my secret string

```
hdparm -N /dev/sdb
    max sectors   = 3896789168/3907029168, HPA is enabled

hdparm --yes-i-know-what-i-am-doing -N p3907000000 /dev/sdb
    setting max visible sectors to 3907000000 (permanent)
    max sectors   = 3907000000/3907029168, HPA is enabled

Reboot disk to apply new settings

dd if=/dev/sdb skip=3900000000 count=150 | xxd | less
    00000000: 4d79 5365 6372 6574 2031 3233 3435 3600  MySecret 123456.
```

- Recover hidden binary data

```
dd if=/dev/sdb of=dd.exe bs=1 skip=$((512*3900000001)) count=76000

md5sum dd.exe
    36a70f825b8b71a3d9ba3ac9c5800683  dd.exe

md5sum /bin/dd
    36a70f825b8b71a3d9ba3ac9c5800683  /bin/dd
```

HPA Exercise: Feedback: Persistent malware

- Feeback: kaplan@cert.at
  - `https://www.schneier.com/blog/archives/2014/02/swap_nsa_exploi.html`
  - `https://en.wikipedia.org/wiki/Host_protected_area`
  - The Rootkit Arsenal; 1st Edition

    Escape and Evasion in the Dark Corners of the System
    Jones & Bartlett Publishers
    Blunden, Bill; 2009 p.538

- How it works
  - `IDENTIFY DEVICE`
  - `SET MAX ADDRESS`
  - `READ NATIVE MAX ADDRESS`

    $\rightarrow$ HPA aware software (like the BIOS)

Block device I/O Error

- `https://github.com/adulau/dcfldd/issues/1`

Known issues #1

Open **msuhanov** opened this issue Oct 24, 2018 · 3 comments

msuhanov commented Oct 24, 2018 • edited

1. Data becomes misaligned when a faulty sector is encountered on a source drive. NIST report, upstream patch.

Github

I/O error: Setup virtual block device

```
00000000: 4265 6769 6e5f 6f66 5f73 6563 746f 725f  Begin_of_sector_
00000010: 3120 2020 2041 4141 4141 4141 4141 4141  1    AAAAAAAAAAA
...
...
000001e0: 4141 4141 4141 4141 4141 4141 4145 6e64  AAAAAAAAAAAAAAEnd
000001f0: 5f6f 665f 7365 6374 6f72 5f31 2020 2020  _of_sector_1
00000200: 4265 6769 6e5f 6f66 5f73 6563 746f 725f  Begin_of_sector_
00000210: 3220 2020 2041 4141 4141 4141 4141 4141  2    AAAAAAAAAAA
...
...
009fffe0: 4141 4141 4141 4141 4141 4141 4145 6e64  AAAAAAAAAAAAAAEnd
009ffff0: 5f6f 665f 7365 6374 6f72 5f32 3034 3830  _of_sector_20480
```

```
losetup -f
losetup /dev/loop21 a.raw
losetup /dev/loop22 a.raw

# Table
# 0 10232 linear /dev/loop21 0
# 10232 1 error
# 10233 10240 linear /dev/loop22 10240

echo -e "0 10232 linear /dev/loop21 0\n10232 1 error\n10233 10240 linear /dev/loop22 1024(
blockdev --getsize64 /dev/mapper/dcfldd
```

I/O error: Results

- Create image files

```
dd     if=/dev/mapper/dcfldd of=a_dd.raw     conv=noerror,sync
dcfldd if=/dev/mapper/dcfldd of=a_dcfldd.raw conv=noerror,sync
```

- Compare `dd` and `dcfldd`

```
10482176 Jan 15 14:33 a_dd.raw
10518528 Jan 15 15:12 a_dcfldd.raw
```

- What about `dc3dd`

```
dc3dd /dev/mapper/dcfldd a_dc3dd.raw log=error_dc3dd.log
```

```
10482176 Jan 15 14:33 a_dd.raw
10482176 Jan 15 14:37 a_dc3dd.raw
10518528 Jan 15 15:12 a_dcfldd.raw
10485760 Jan 14 16:35 a.raw
```

I/O error: Investigate results

- What about `md5sum`

```
23211b0a0670e138b4b9880f66ff231b  a_dd.raw
3ae9377b678c03f441d98add58d447d6  a_dc3dd.raw
7671b9867b3749a3c78658e212d9c0ea  a_dcfldd.raw
```

- Investigate dd output

```
004fefd0: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
004fefe0: 4141 4141 4141 4141 4141 4141 4145 6e64  AAAAAAAAAAAAAAEnd
004feff0: 5f6f 665f 7365 6374 6f72 5f31 3032 3332  _of_sector_10232
004ff000: 0000 0000 0000 0000 0000 0000 0000 0000  ................
004ff010: 0000 0000 0000 0000 0000 0000 0000 0000  ................
004ff020: 0000 0000 0000 0000 0000 0000 0000 0000  ................
...
...
004fffe0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
004ffff0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00500000: 4265 6769 6e5f 6f66 5f73 6563 746f 725f  Begin_of_sector_
00500010: 3130 3234 3841 4141 4141 4141 4141 4141  10248AAAAAAAAAAA
00500020: 4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
```

- Conclusion:

```
--> Addressing:  8 Sectors empty
--> Content:    15 Sectors skipped
```

I/O error: Investigate results

- Investigate `dc3dd` output

```
004fefe0: 4141 4141 4141 4141 4141 4141 4145 6e64  AAAAAAAAAAAAAEnd
004feff0: 5f6f 665f 7365 6374 6f72 5f31 3032 3332  _of_sector_10232
004ff000: 0000 0000 0000 0000 0000 0000 0000 0000  ................
...
...
004ff1f0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
004ff200: 4265 6769 6e5f 6f66 5f73 6563 746f 725f  Begin_of_sector_
004ff210: 3130 3234 3141 4141 4141 4141 4141 4141  10241AAAAAAAAAAA
```

- Conclusion:

```
--> Addressing: 1 Sector empty
--> Content:    8 Sectors skipped
```

- Investigate `dcfldd` output

```
00a001f0: 5f6f 665f 7365 6374 6f72 5f32 3034 3830  _of_sector_20480
00a00200: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00a00210: 0000 0000 0000 0000 0000 0000 0000 0000  ................
...
```

- Conclusion:

```
--> Missmatch at I/O error
--> End of file blown up with 63 sectors of 0X00
--> Totoal output: 64 sectors to much
```

I/O error: Feedback

- Feedback D. Byers

```
David Byers and Nahid Shahmehri.
Contagious errors:
Understanding and avoiding issues with imaging drives containing faulty sectors.
The International Journal of Digital Forensics and Incident Response,
ISSN 1742-2876, E-ISSN 1873-202X, Vol. 5, no 1, p. 29-33
```

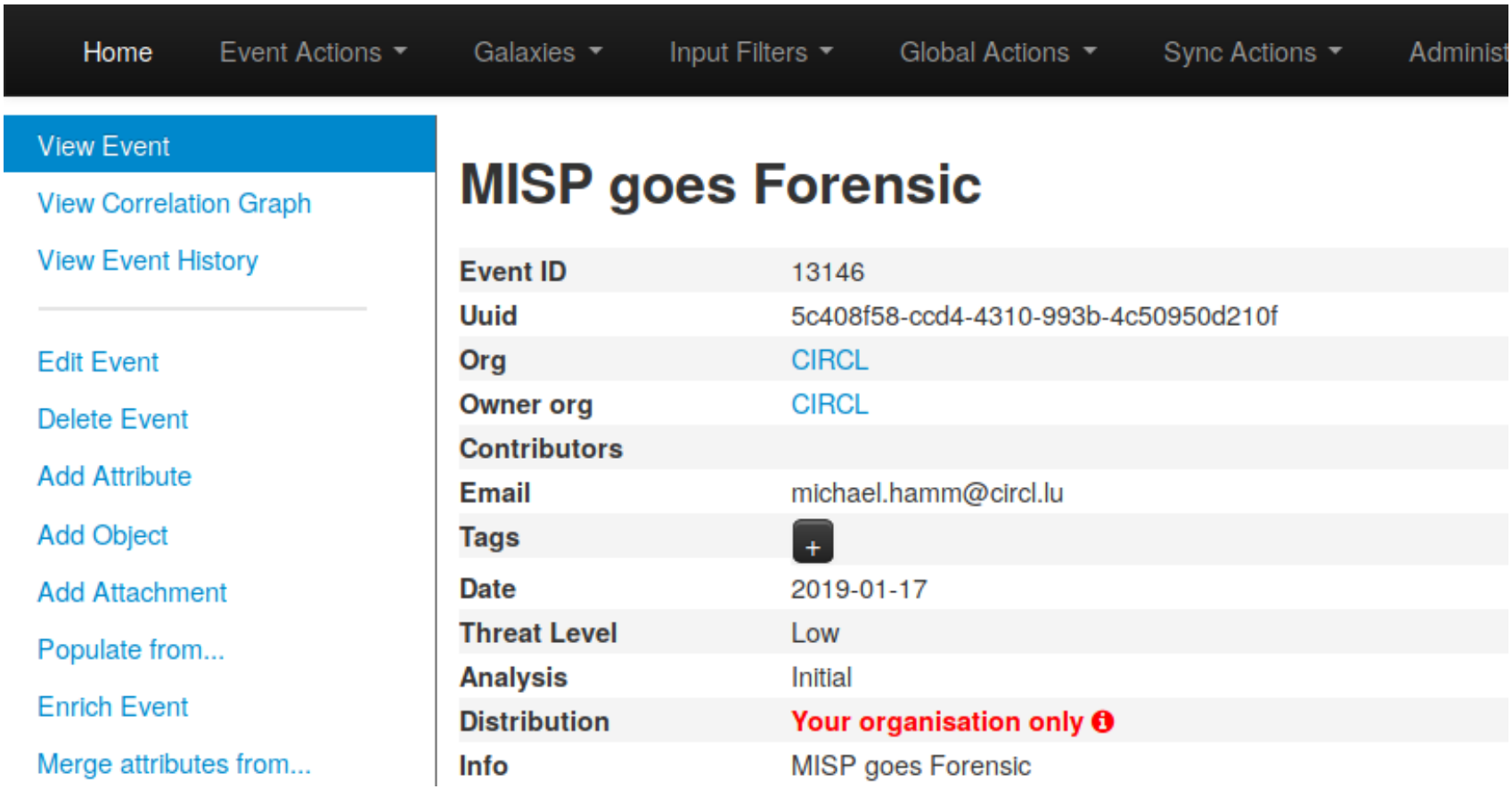
  - READ call goes through the kernel page cache
    → Linux reads 4096 bytes
    → 1 faulty sector → fail to read 8 sectors

  - Reading a device in direct I/O mode solve the problem
    → `dd if=/dev/mapper/dcfldd of=a_dd.raw iflag=direct conv=noerror,sync`

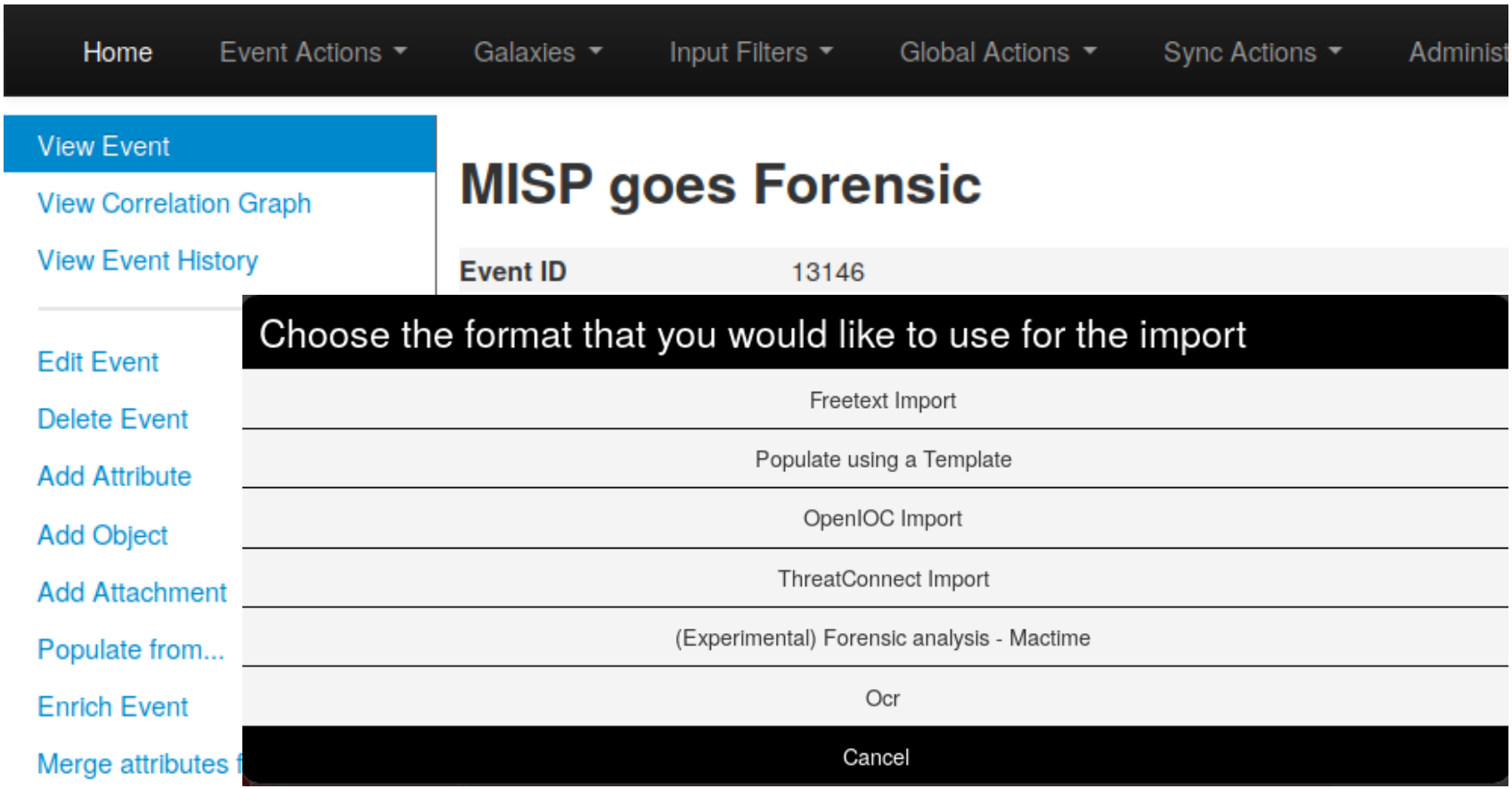
- Change loop device table: remove missing sectors

```
# 0 10232 linear /dev/loop21 0          --> 0 10239 linear /dev/loop21 0
# 10232 1 error                         --> 10239 1 error
# 10233 10240 linear /dev/loop22 10240  --> 10240 10240 linear /dev/loop22 10240
```

MISP goes Forensic



MISP goes Forensic

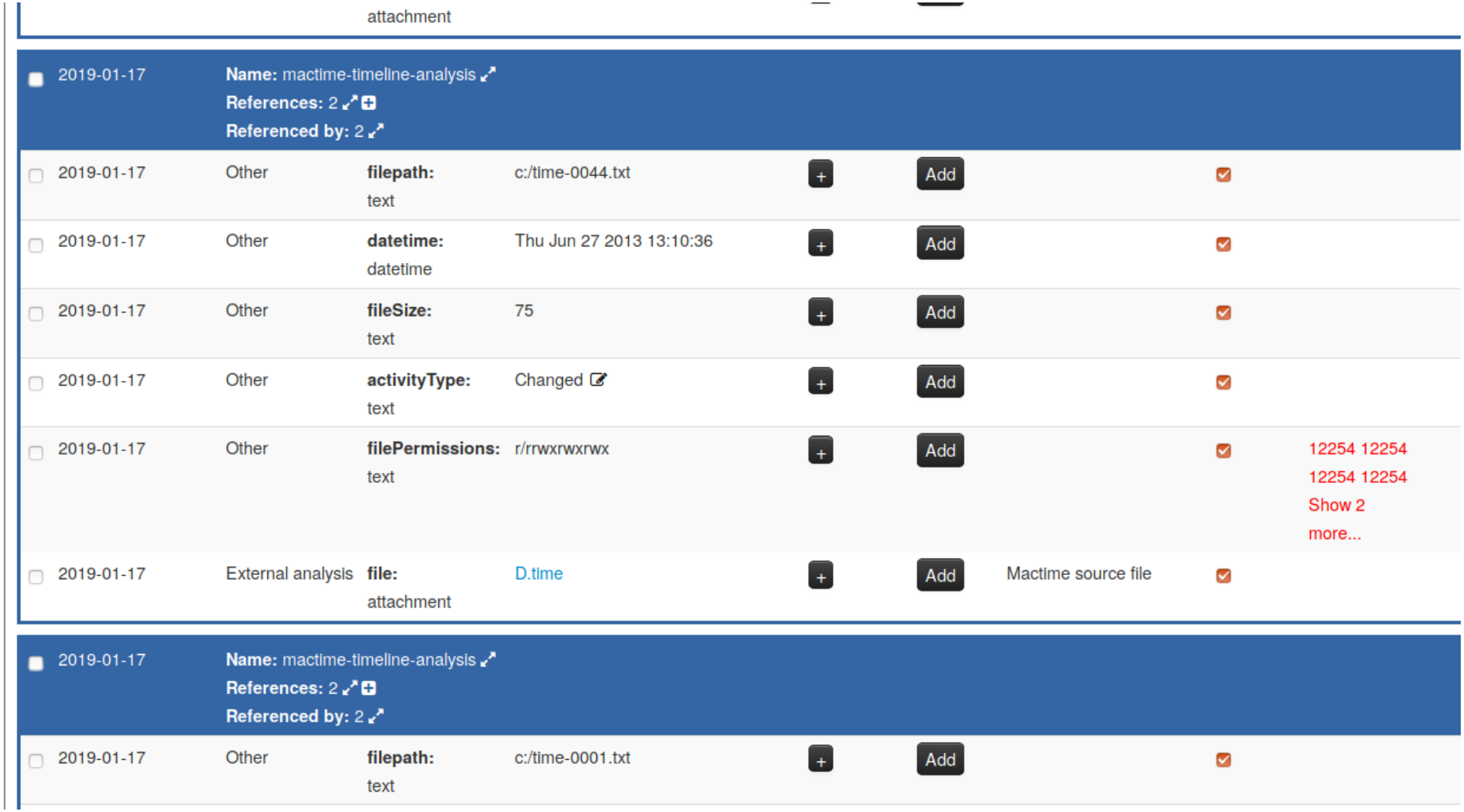


MISP goes Forensic

Select text for further analysis

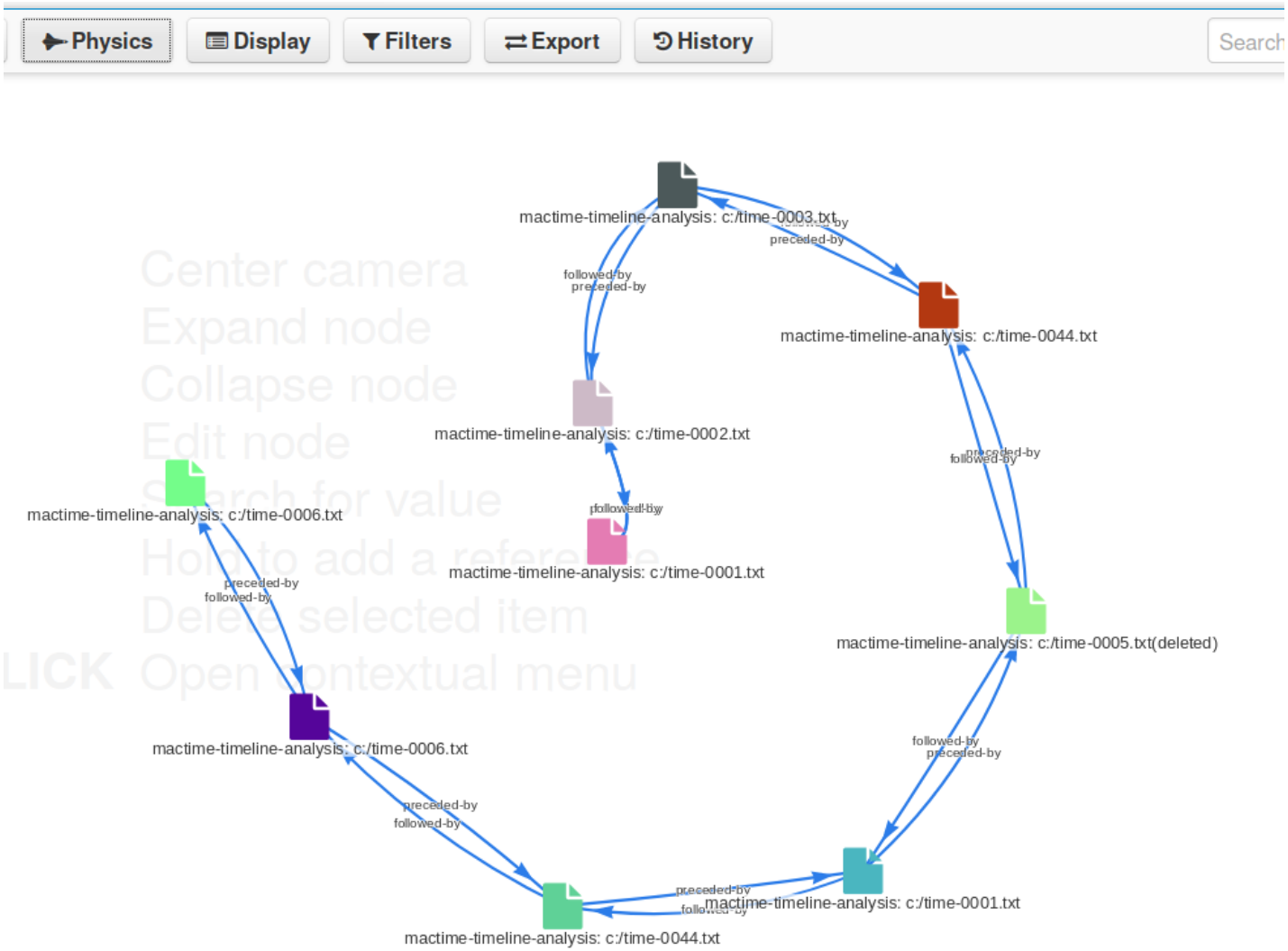
Select	Filepath	File Size	Activity Type
☐	c:/$Extend/$RmMetadata	336	Accessed,Created,Changed,Mo
☐	c:/$Extend/$RmMetadata/$TxfLog/$TxfLog.blf	65536	Accessed,Created
☐	c:/$Extend/$RmMetadata/$TxfLog/$TxfLogContainer00000000000000000002	10485760	Accessed,Created
☐	c:/$MFT	262144	Accessed,Created,Changed,Mo
☑	c:/time-0001.txt	113	Created
☑	c:/time-0002.txt	75	Created,Changed,Modified
☑	c:/time-0003.txt	75	Created,Changed,Modified
☑	c:/time-0044.txt	75	Created,Modified
☑	c:/time-0005.txt(deleted)	75	Accessed,Created,Changed,Mo
☑	c:/time-0001.txt	113	Changed,Modified
☐	c:/time-0003-	75	Accessed,Created,Changed
☑	c:/time-0044.txt	75	Changed
☑	c:/time-0006.txt	20	Modified
☑	c:/time-0006.txt	20	Accessed,Created,Changed
☐	c:/$Extend/$RmMetadata/$TxfLog/$TxfLogContainer00000000000000000002	10485760	Changed,Modified
☐	c:/time-0001.txt	113	Accessed
☐	c:/$Extend/$RmMetadata/$TxfLog/$TxfLog.blf	65536	Changed,Modified

MISP goes Forensic



		attachment					
2019-01-17	**Name:** mactime-timeline-analysis **References:** 2 **Referenced by:** 2						
2019-01-17	Other	**filepath:** text	c:/time-0044.txt	+	Add		
2019-01-17	Other	**datetime:** datetime	Thu Jun 27 2013 13:10:36	+	Add		
2019-01-17	Other	**fileSize:** text	75	+	Add		
2019-01-17	Other	**activityType:** text	Changed	+	Add		
2019-01-17	Other	**filePermissions:** text	r/rrwxrwxrwx	+	Add		12254 12254 12254 12254 Show 2 more...
2019-01-17	External analysis	**file:** attachment	D.time	+	Add	Mactime source file	
2019-01-17	**Name:** mactime-timeline-analysis **References:** 2 **Referenced by:** 2						
2019-01-17	Other	**filepath:** text	c:/time-0001.txt	+	Add		

MISP goes Forensic



Lessons learned in Forensics

- Modify data on "Read Only" mounted device
- Data recovery from damaged ZIP archives
- Hidding data in HPA
- Block device I/O Error
- MISP goes Forensic

Q & A

Thank you