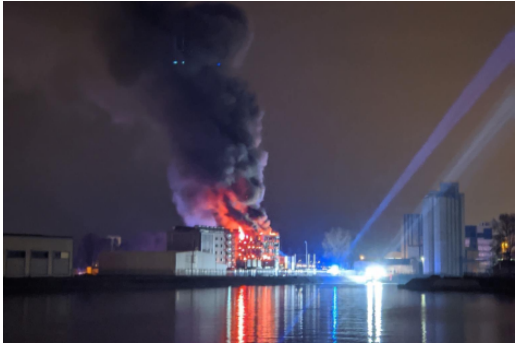
SYNACKTIV

Manger mes dossiers par la racine

Comment un rm -rf a dérapé et a supprimé l'intégralité de mon système



27 Mai 2021

Synacktiv

Lucas Georges

The command

```
156308 MESSAGE= lucasg : TTY=pts/23 ; PWD=/media/lucasg/5E08-D3D0/ ; USER=root ;
       COMMAND=/usr/bin/rm -r ./var/lib/chroots/nginx/[...]/css/images/btn
159858 MESSAGE= lucasg : TTY=pts/23 ; PWD=/media/lucasg/5E08-D3D0/ ; USER=root ;
       COMMAND=/usr/bin/rm -rf ./var/lib/chroots/nginx/[...]/css/images/btn
```

The result

```
|-----------d_time  Histogram----------------  after  --------------------  Thu Sep 24 16:11:15 2020
1601024355 :    23783 |*                                                |   Fri Sep 25 08:59:15 2020
1601084835 :      689 |*                                                |   Sat Sep 26 01:47:15 2020
1601145315 :        0 |                                                 |   Sat Sep 26 18:35:15 2020
1601205795 :      207 |*                                                |   Sun Sep 27 11:23:15 2020
1601266275 :        1 |*                                                |   Mon Sep 28 04:11:15 2020
1601326755 :    18955 |*                                                |   Mon Sep 28 20:59:15 2020
1601387235 :  2054247 |************************************************ |   Tue Sep 29 13:47:15 2020
1601447715 :        0 |                                                 |   Wed Sep 30 06:35:15 2020
1601508195 :        0 |                                                 |   Wed Sep 30 23:23:15 2020
1601568675 :        0 |                                                 |   Thu Oct  1 16:11:15 2020
```

```
$ ext4magic /dev/mapper/lucasg--vg-root -H -a $(date -d ”-7 days” +%s)
```

The result

```
|-----------d_time  k
1601024355 :    2378
1601084835 :     689
1601145315 :       0
1601205795 :     207
1601266275 :       1
1601326755 :    1895
1601387235 :  205424
1601447715 :       0
1601508195 :       0
1601568675 :       0
```

```
16:11:15 2020
08:59:15 2020
01:47:15 2020
18:35:15 2020
11:23:15 2020
04:11:15 2020
20:59:15 2020
13:47:15 2020
06:35:15 2020
23:23:15 2020
16:11:15 2020
```

```
$ ext4magic
```

```
" +%s)
```

Table des matières

whoami

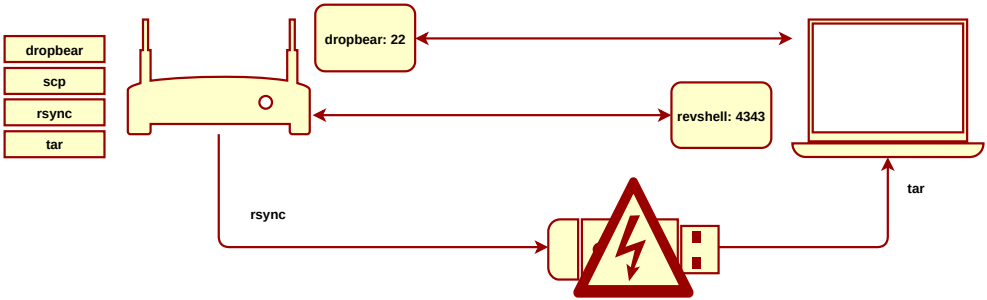
Lucas Georges
@_lucas_georges_

- Reverse Engineer @Synacktiv
- Vulnerability research & exploitation

Synacktiv

- Offensive security company (pentest, reverse, dev)
- ~ 90 ninjas
- We are always hiring

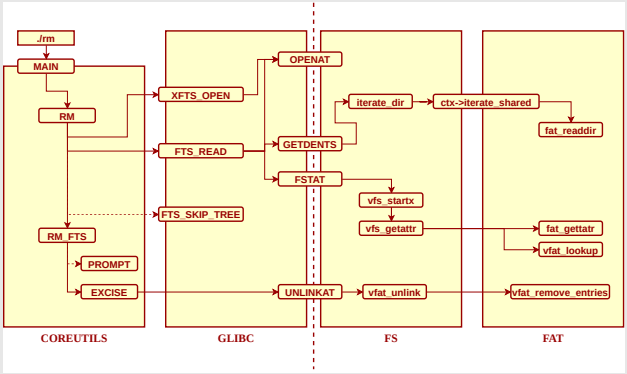
Context



Scenario

- Aim : extract a rootfs from an embedded device
  - rsync the whole rootfs to a pendrive
  - copy and tarball the rootfs using a laptop with all the correct options
- What happened :
  - rsync introduced write errors on the usb controlled and corrupted the pendrive's fs
  - When trying to remove the folder, it removed the whole system !

How rm works


./rm
MAIN
RM
RM_FTS
PROMPT
EXCISE
XFTS_OPEN
FTS_READ
FTS_SKIP_TREE
OPENAT
GETDENTS
FSTAT
UNLINKAT
iterate_dir
ctx->iterate_shared
fat_readdir
vfs_startx
vfs_getattr
vfat_unlink
fat_gettatr
vfat_lookup
vfat_remove_entries
COREUTILS
GLIBC
FS
FAT

Reproducing the bug

Linux fragmentation

Description	coreutils	glibc	kernel	BUG ?
Victim (debian unstable)	????	????	~ 5.2	YES
Debian Jessie 4.9.0-14	8.26	2.24	4.9.240-2	NO
Debian Stretch 4.19.0-11	8.30	2.28	4.19.146-1	NO
Debian Unstable 5.4.0-0.bpo.2	8.30	2.28	5.4.8-1~bpo10+1	NO
Ubuntu 18.04-3	8.28	2.27	4.18.0-25	YES
Ubuntu 18.04-3	8.28	2.27	5.0.0-23	YES
Ubuntu 18.04-5	8.28	2.27	5.4.0-51	NO
Ubuntu 20.04-1	8.30	2.31	5.4.0-53	NO

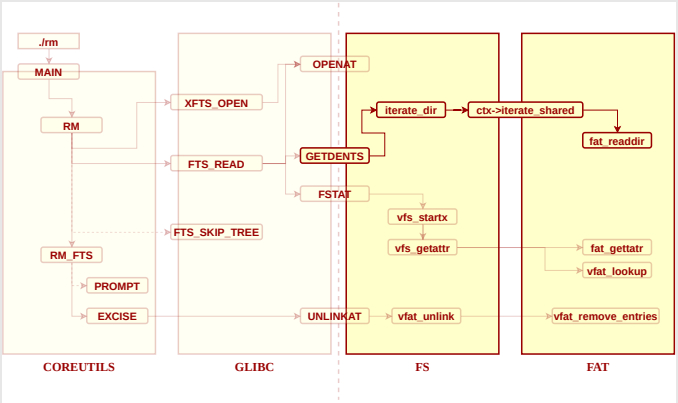
Reproducing the bug

strace output

```
write(2, "/home/arma/coreutils/src/rm: des"..., 1263) = 1263
read(0, "y\n", 1024)                    = 2
openat(17, "\3K\302\240\316\264.\1p\303\241", O_RDONLY|O_NOCTTY|O_NONBLOCK|O_NOFOLLOW|O_CLOEXEC|O_DIRECTORY) = 15
getdents(15, /* 931 entries */, 32768) = 32744
getdents(15, /* 748 entries */, 32768) = 25024
getdents(15, /* 0 entries */, 32768) = 0
close(15)                               = 0
newfstatat(20, "\1.\f", 0x5643647c58a8, AT_SYMLINK_NOFOLLOW) = -1 EIO (Input/output error)
newfstatat(20, "\0100\303\241\303\237\1\n\22\317\200.\1 \303\241", 0x564364b86f28, AT_SYMLINK_NOFOLLOW) = -1 EIO (Input/output error)
newfstatat(20, "\7\20\303\241\303\237\0010\303\242\22.\342\225\2252\342\224\200", 0x564364b8cc68, AT_SYMLINK_NOFOLLOW) = -1 ELOOP (Too
       many levels of symbolic links)
// [... snipped ...]
newfstatat(20, "\20\342\224\224\303\271\317\203", 0x564364ba9288, AT_SYMLINK_NOFOLLOW) = -1 EIO (Input/output error)
newfstatat(20, "\0010\303\241\23\21", 0x564364baa268, AT_SYMLINK_NOFOLLOW) = -1 ELOOP (Too many levels of symbolic links)
newfstatat(20, "/", {st_mode=S_IFDIR|0755, st_size=4096, ...}, AT_SYMLINK_NOFOLLOW) = 0
openat(20, "/", O_RDONLY|O_NOCTTY|O_NONBLOCK|O_NOFOLLOW|O_DIRECTORY) = 15
fstat(15, {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
fcntl(15, F_GETFL)                      = 0x38800 (flags O_RDONLY|O_NONBLOCK|O_LARGEFILE|O_NOFOLLOW|O_DIRECTORY)
fcntl(15, F_SETFD, FD_CLOEXEC)          = 0
getdents(15, /* 28 entries */, 32768) = 744
close(15)                               = 0
write(2, "/home/arma/coreutils/src/rm: des"..., 1265) = 1265
read(0, "y\n", 1024)                    = 2
write(2, "skipping '/media/arma/5E08-D3D0/"..., 1254) = 1254
```

The bug

__fat_readdir



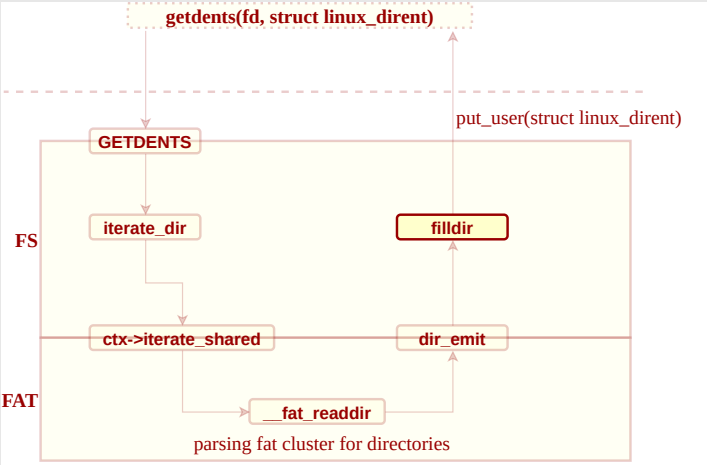
The bug

__fat_readdir

```
[  +0.000000] __fat_readdir
// [....]
[  +0.000017] dir_emit folder : 2c 20 c3 bc cf 83 44 20 c2 a5 15 2e 02 20 c3 a9 , ....D ..... ..
[  +0.000072] dir_emit folder : c3 a4 20 c2 a5 cf 83 2e c3 87 30 c2 a5          .. .......0..
[  +0.000072] dir_emit folder : 60 30 c2 a5 cf 83 01                            `0.....
[  +0.000071] dir_emit folder : 03 21 c3 a6 cf 84 2e 01 40 c3 a2                .!......@..
[  +0.000083] dir_emit folder : 10                                              .
[  +0.000015] dir_emit folder : 14 e2 94 94 c3 b6 cf 83 2e 10                   ..........
[  +0.000041] dir_emit folder : 01 10 ce b1 c3 9f 2e 01 21 c3 a2                ........!..
[  +0.000181] dir_emit folder : 04 10 c3 89 cf 83 01 2e 24 10 c2 a5             ........$...
[  +0.000040] dir_emit folder : e2 96 84 20 c2 a5 cf 83 63 2e e2 96 84 20 c2 a5 ... ....c.... ..
[  +0.000093] dir_emit folder : 24 30 c2 a5 e2 95 92 30 30 c3 ac e2 95 92 2e 2c $0.....00......,
[  +0.000001] dir_emit folder : 30 c3 ac                                        0..
[  +0.000164] dir_emit folder : 10 e2 94 94 c3 b9 cf 83                         ........
[  +0.000024] dir_emit folder : 01 30 c3 a1 13 11                               0....
[  +0.000092] dir_emit folder : 2f                                              /
[  +0.000091] dir_emit folder : 01 60 c3 a1 c3 9f 2e 4c 30 c3 a6                .`.....L0..
[  +0.003489] __fat_readdir
```

Explaining the bug

filldir



Explaining the bug

Ubuntu 18.04-3

```
static int filldir(struct dir_context *ctx, const char *name, int namlen,
          loff_t offset, u64 ino, unsigned int d_type)
{
    struct linux_dirent __user * dirent;
    struct getdents_callback *buf =
        container_of(ctx, struct getdents_callback, ctx);
    unsigned long d_ino;
    int reclen = ALIGN(offsetof(struct linux_dirent, d_name) + namlen + 2,
        sizeof(long));

    buf->error = -EINVAL; /* only used if we fail.. */
    if (reclen > buf->count)
        return -EINVAL;
    d_ino = ino;
    if (sizeof(d_ino) < sizeof(ino) && d_ino != ino) {
        buf->error = -EOVERFLOW;
        return -EOVERFLOW;
    }
    // [...]
```

Ubuntu 20.04-1

```
static int filldir(struct dir_context *ctx, const char *name, int namlen,
          loff_t offset, u64 ino, unsigned int d_type)
{
    struct linux_dirent __user *dirent, *prev;
    struct getdents_callback *buf =
        container_of(ctx, struct getdents_callback, ctx);
    unsigned long d_ino;
    int reclen = ALIGN(offsetof(struct linux_dirent, d_name) + namlen + 2,
        sizeof(long));
    int prev_reclen;

    buf->error = verify_dirent_name(name, namlen);
    if (unlikely(buf->error))
        return buf->error;
    buf->error = -EINVAL; /* only used if we fail.. */
    if (reclen > buf->count)
        return -EINVAL;
    d_ino = ino;
    if (sizeof(d_ino) < sizeof(ino) && d_ino != ino) {
        buf->error = -EOVERFLOW;
        return -EOVERFLOW;
    }
```

Explaining the bug

verify_dirent_name

```
/*
 * POSIX says that a dirent name cannot contain NULL or a '/'.
 *
 * It's not 100% clear what we should really do in this case. The filesystem is clearly corrupted, but returning a hard
 * error means that you now don't see any of the other names either, so that isn't a perfect alternative.
 *
 * And if you return an error, what error do you use? Several filesystems seem to have decided on EUCLEAN being the error
 * code for EFSCORRUPTED, and that may be the error to use. Or just EIO, which is perhaps more obvious to users.
 *
 * In order to see the other file names in the directory, the caller might want to make this a "soft" error: skip the
 * entry, and return the error at the end instead.
 * ...
 *
 * Note the PATH_MAX check - it's arbitrary but the real kernel limit on a possible path component, not NAME_MAX,
 * which is the technical standard limit.
 */
static int verify_dirent_name(const char *name, int len)
{
    if (len <= 0 || len >= PATH_MAX)
        return -EIO;
    if (memchr(name, '/', len))
        return -EIO;
    return 0;
}
```

Explaining the bug

Bug report

```
From: Jann Horn <jannh@google.com>
Date: Fri, 18 Jan 2019 17:14:39 +0100

When you e.g. run `find` on a directory for which getdents returns
”filenames” that contain slashes, `find` passes those ”filenames” back
to the kernel, which then interprets them as paths.

That could conceivably cause userspace to do something bad when
accessing something like an untrusted USB stick, but I’m not
aware of any specific example.
```

https://lore.kernel.org/lkml/20190118161440.220134-2-jannh@google.com/

Table des matières

dd is your friend

```
$ dd if=/dev/nvme0n1p3 of=/media/ubuntu/backup_dd_lucasg/nvme0n1p3.bak bs=1G status=
    progress
```

Loopdevice

cryptsetup automatically mount the device, no need to use losetup :

```
$ sudo cryptsetup open --type luks ./nvme0n1p3 backup_luks
$ ls -als /dev/mapper/
total 0
0 drwxr-xr-x 2 root root   140 Sep 30 08:29 .
0 drwxr-xr-x 23 root root 4900 Oct 1 14:26 ..
0 crw------- 1 root root 10, 236 Sep 29 16:30 control
0 lrwxrwxrwx 1 root root     7 Sep 30 09:30 lucasg--vg-root -> ../dm-1
0 lrwxrwxrwx 1 root root     7 Sep 30 08:29 lucasg--vg-swap_1 -> ../dm-2
0 lrwxrwxrwx 1 root root     7 Sep 30 08:29 luks-33fc53db-c0e8-4208-ae93-43a9a5136758 -> ../dm-3
0 lrwxrwxrwx 1 root root     7 Sep 30 08:28 backup_luks -> ../dm-0
```

ext4magic

`ext4magic` can restore deleted files thanks to `ext4` journalisation :

```
$ sudo ext4magic /dev/mapper/lucasg--vg-root -M -d /media/ubuntu/backup_dd_lucasg/ext4magic
Warning: Activate magic-scan or disaster-recovery function, may be some command line options ignored
"/media/ubuntu/backup_dd_lucasg/ext4magic" accept for recoverdir
Filesystem in use: /dev/mapper/lucasg--vg-root

Using internal Journal at Inode 8
Activ Time after : Tue Sep 29 13:03:24 2020
Activ Time before : Thu Oct 1 16:16:45 2020
Inode 2 is allocated
--------   /media/ubuntu/backup_dd_lucasg/ext4magic/boot
--------   /media/ubuntu/backup_dd_lucasg/ext4magic/home/lucasg/.cache/mozilla/firefox/adu3n5h2.default/cache2/doomed
--------   /media/ubuntu/backup_dd_lucasg/ext4magic/home/lucasg/.cache/mozilla/firefox/adu3n5h2.default/cache2
--------   /media/ubuntu/backup_dd_lucasg/ext4magic/home/lucasg/.cache/mozilla/firefox/adu3n5h2.default
--------   /media/ubuntu/backup_dd_lucasg/ext4magic/home/lucasg/.cache/mozilla/firefox
--------   /media/ubuntu/backup_dd_lucasg/ext4magic/home/lucasg/.cache/mozilla
--------   /media/ubuntu/backup_dd_lucasg/ext4magic/home/lucasg/.cache/google-chrome-beta/Profile 2/Cache/83db0be9d0d8d957_0
--------   /media/ubuntu/backup_dd_lucasg/ext4magic/home/lucasg/.cache/google-chrome-beta/Profile 2/Cache/83db0be9d0d8d957_s
...
(...)
Segmentation fault
```

sleuthkit

```
console
$ sudo ./tools/fstools/fls -F -p -d -m '/' -r /dev/mapper/lucasg--vg-root
0|/home/git (deleted)|51541506|d/rr--r--r--|33|33|0|1582119050|1601383833|1601383833|1572002125
0|/home/lucasg/.cache/mozilla/firefox/adu3n5h2.default/cache2/doomed/1ce91c99e9881667d35749bfba293accd26374b3.jd6892 (deleted)|52074044|
       r/rrw-r--r--|1000|1000|0|1586687532|1601383958|1601383958|1585213454
0|/home/lucasg/.cache/mozilla/firefox/adu3n5h2.default/cache2/doomed/1ce91c99e9881667d35749bfba293accd26374b3 (deleted)|52074044|r/rrw-r
       --r--|1000|1000|0|1586687532|1601383958|1601383958|1585213454
0|/home/lucasg/.cache/mozilla/firefox/adu3n5h2.default/cache2/doomed/3c597a80b51a24c02510e986e8c27bdb62e99ba6 (deleted)|52069943|r/rrw-r
       --r--|1000|1000|0|1586687064|1601383958|1601383958|1585213454
(...)
```

`fls` can carve back deleted files and folders, but everything is empty :(

```
$ sudo ./tools/fstools/istat /dev/mapper/lucasg--vg-root 51685967
Group: 6321
Generation Id: 3563593447
uid / gid: 1000 / 1000
mode: rrw-------
Flags: Extents,
size: 0
num of links: 0
Direct Blocks:
```

The only forensic tool that always work

```
ubuntu@ubuntu:/mnt/backup$ sudo strings -f -a -tx /dev/mapper/lucasg--vg-root > /mnt/
    backup/strings.txt
# (wait for the night)
ubuntu@ubuntu:/mnt/backup$ ls -alsh ./strings*
620G -rw-rw-r-- 1 ubuntu ubuntu 620G Sep 30 18:48 ./strings.txt
```

Bash helpers

```
cat_mem() { sudo dd if=/dev/mapper/lucasg--vg-root bs=1 skip=$((”$1”)) count=$((”$2”)) 2>/
    dev/null; }
show_mem() { sudo dd if=/dev/mapper/lucasg--vg-root bs=1 skip=$((”$1”)) count=$((”$2”))
    2>/dev/null | hd | less; }
truncate_using_marker() { sed -e '/'”$1”'/,${//i '”$1”'' -e 'd}';}
```

My personal Disaster Recovery Plan

- My professional keepass is regularly rsync'ed to an intranet server
- Slides and exercises for a training were backuped on the disk of a colleague … which just quit Synacktiv and wiped its disk :O
- No client data was lost since its not stored on work laptops.
- Some ongoing mission's files were archived and gpg encrypted on a shared external drive but … the GPG key itself wasn't backed up !
- Private notes and projects were not backuped at all :(

What to carve ?

- GPG key
- SSH keys
- other secrets
- private stuff

How to carve back a private GPG key ?

```
> i have really big problem because i accydently deleted /.gnupg, but still i have backuped
 /.gnupg/private-keys-v1.d so i have 4 “hashfile” name files with suffix .key

That good. Run gpg once to create a new .gnupg directory (or create it
manually). Then copy the four files to the new private-keys-v1.d
directory and you have restored the secret key material. Now you need
to get a copy of your two (I guess) public keys. They should be on the
keyservers or you have send them to other places, get a copy and gpg
--import them. Better restart the gpg-agent (gpgconf --kill gpg-agent).
That's it.

Private key files in gnupg/private-keys-v1.d have filenames of the pattern
  [0-9A-F]{40}.key
```

Source : https://lists.gnupg.org/pipermail/gnupg-users/2016-December/057246.html

1. Carve back the files under `~/.gnupg/private-keys-v1.d/*.key` folder
2. Retrieve the hash file patterns, also known as `”keygrip”` for gpg keys

Carving back the contents of private GPG keys

```
ubuntu@ubuntu:~/Desktop/$ grep -F "protected-private-key" /mnt/backup/strings.txt | tee /mnt/backup/
     strings.txt/gpg/grep_results.txt
/dev/mapper/lucasg--vg-root: 2700d38a2 protected-private-key
/dev/mapper/lucasg--vg-root: 2837fa922 protected-private-key
# (... etc ...)
```

```
ubuntu@ubuntu:~/Desktop/$ cat /mnt/backup/gpg/grep_results.txt | awk '{print $3}' | sort | uniq -c
      2 (21:protected-private-key(3:rsa(1:n513:
      1 -F
      1 https://stackoverflow.com/questions/25869207/unprotected-private-key-file
      3 https://stackoverflow.com/questions/25869207/unprotected-private-key-fileheroku
    179 protected-private-key
      1 (protected-private-key(d
      1 (protected-private-key(dmplac
     44 (protected-private-key(dsa(p%m)(q%m)(g%m)(y%m)(protected
    138 (protected-private-key(ecc(curve
     46 (protected-private-key(elg(p%m)(g%m)(y%m)(protected
     46 (protected-private-key(rsa(n%m)(e%m)(protecte
ubuntu@ubuntu:~/Desktop/$ cat ./backup/gpg/grep_results.txt | grep -F "(21:protected-private-key(3:rsa(1:n513:"
/dev/mapper/lucasg--vg-root: c6978d2000 (21:protected-private-key(3:rsa(1:n513:
/dev/mapper/lucasg--vg-root: c6978d3000 (21:protected-private-key(3:rsa(1:n513:
# (... etc ...)
```

Reconstruct back the ~/.gnupg/XXXX.key files

```
cat_mem() { sudo dd if=/dev/mapper/lucasg--vg-root bs=1 skip=$((”$1”)) count=$((”$2”)) 2>/
    dev/null; }
remove_trailing_null_bytes() { sed '$ s/\x00*$//';}
recover_gpg_key() {
       cat_mem 0x”$1” 0x1000 | remove_trailing_null_bytes > ./backup/gpg/
           recovered_gpg_key_”$1”.key;
       ls -als ./backup/gpg/recovered_gpg_key_”$1”.key;
}

$ cat ./backup/gpg/grep_results.txt | grep -F ”(21:protected-private-key(3:rsa(1:n513:” |
    awk '{print $2}' | while read offset; do recover_gpg_key $offset; done
4 -rw-rw-r-- 1 ubuntu ubuntu 2056 Oct 1 11:28 ./backup/gpg/recovered_gpg_key_c6978d2000.
    key
4 -rw-rw-r-- 1 ubuntu ubuntu 2056 Oct 1 11:28 ./backup/gpg/recovered_gpg_key_c6978d3000.
    key
```

Understand ~/.gnupg/XXXX.key file format

```
** Protected Private Key Format

A protected key is like this:

(protected-private-key
   (rsa
    (n #00e0ce9..[some bytes ]..51#)
    (e #010001#)
    (protected mode (parms)
         encrypted_octet_string)
    (protected-at <isotimestamp>)
   )
   (uri http://foo.bar x-foo:what)
   (comment whatever)
)
```

Source : https://lists.gnupg.org/pipermail/gnupg-devel/2017-December/033295.html

```
$ hd ./backup/gpg/recovered_gpg_key_c6978d2000.key
00000000 28 32 31 3a 70 72 6f 74 65 63 74 65 64 2d 70 72 |(21:protected-pr|
00000010 69 76 61 74 65 2d 6b 65 79 28 33 3a 72 73 61 28 |ivate-key(3:rsa(|
00000020 31 3a 6e 35 31 33 3a 00 d8 70 5a b9 2e 00 83 9b |1:n513:..pZ.....|
00000030 e3 d1 fd 41 79 75 28 a3 dd a9 43 0e b7 37 61 cd |...Ayu(...C..7a.|
(snipped)
00000220 7e f8 55 10 5b b1 82 b1 29 28 31 3a 65 33 3a 01 |~.U.[...)(1:e3:.|
00000230 00 01 29 28 39 3a 70 72 6f 74 65 63 74 65 64 32 |..)(9:protected2|
00000240 35 3a 6f 70 65 6e 70 67 70 2d 73 32 6b 33 2d 73 |5:openpgp-s2k3-s|
00000250 68 61 31 2d 61 65 73 2d 63 62 63 28 28 34 3a 73 |ha1-aes-cbc((4:s|
00000260 68 61 31 38 3a d3 1e 66 e2 87 90 37 ad 39 3a 31 |ha18:..f...7.9:1|
(snipped)
000007e0 69 9e 29 28 31 32 3a 70 72 6f 74 65 63 74 65 64 |i.)(12:protected|
000007f0 2d 61 74 31 35 3a 32 30 31 38 31 31 30 38 54 31 |-at15:20181108T1|
00000800 37 34 36 30 38 29 29 29                         |74608)))|
```

Compute back the corresponding keygrips

```
$ gpgsm --call-protect-tool --show-keygrip '/media/ubuntu/a733ad08-1f25-44b5-8dc6-38336
    ef027e7/backup/gpg/recovered_gpg_key_c6978d2000.key'
gpgsm: Note: '--show-keygrip' is not considered an option
DA39A3EE5E6B4B0D3255BFEF95601890AFD80709
$ gpgsm --call-protect-tool --show-keygrip '/media/ubuntu/a733ad08-1f25-44b5-8dc6-38336
    ef027e7/backup/gpg/recovered_gpg_key_c6978d3000.key'
gpgsm: Note: '--show-keygrip' is not considered an option
20EABE5D64B0E216796E834F52D61FD0B70332FC
```

Putting everything back in ~/.gnupg

```
$ mkdir -p ~/.gnupg/private-keys-v1.d && rm ~/.gnupg/pubring.kbx
$ cp recovered_gpg_key_c6978d2000.key ~/.gnupg/private-keys-v1.d/
     DA39A3EE5E6B4B0D3255BFEF95601890AFD80709.key
$ cp recovered_gpg_key_c6978d3000.key ~/.gnupg/private-keys-v1.d/20
     EABE5D64B0E216796E834F52D61FD0B70332FC.key

$ gpg --import lg.asc
gpg: keybox '~/.gnupg/pubring.kbx' created
gpg: key 0CB1775E1FC8AF64: public key "Lucas Georges <lucas.georges@synacktiv.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1

$ gpg -d /mnt/backup/p2o_miami/p2o_archive_21_07_2020.7z.gpg | tail -n 10 | hd
gpg: encrypted with 4096-bit RSA key, ID DC613035AC6B5FD5, created 2018-11-08
     "Lucas Georges <lucas.georges@synacktiv.com>"
00000000 a4 99 12 e5 48 3c 8a 02 37 b1 16 b0 16 84 5d 27 |....H<..7.....]'|
00000010 c8 bd af 43 33 fa b6 b7 af 6d e3 b7 aa 25 2f ee |...C3....m...%/.|
...
```

Carving back the contents of a private SSH key

```
$ cat_mem() {sudo dd if=/dev/mapper/lucasg--vg-root bs=1 skip=$(("$1")) count=$(("$2")) 2>/dev/null;}
$ truncate_using_marker() { sed -e '/'"$1"'/,${//i '"$1"'' -e 'd}';}

$ recover_ssh_key() {
    cat_mem 0x"$1" 0x8000 | truncate_using_marker '-----END OPENSSH PRIVATE KEY-----' > ./backup/ssh/
         recovered_ssh_key_"$1".key;
    ls -als ./backup/ssh/recovered_ssh_key_"$1".key;
}
ubuntu@ubuntu:~/$ grep -F "BEGIN OPENSSH PRIVATE KEY" /mnt/backup/strings.txt >
  /mnt/backup/ssh/other_grep_results.txt
ubuntu@ubuntu:~/$ cat ./backup/ssh/other_grep_results.txt | awk '{print $2}' | while read offset; do
     recover_ssh_key "$offset"; done
```

```
ubuntu@ubuntu:~/$ ls -als ./backup/ssh/recov* | grep " 34 " | awk '{print $10}' | xargs -I {} rm {}
ubuntu@ubuntu:~/$ ls -als ./backup/ssh/recov* | grep " 70 " | awk '{print $10}' | xargs -I {} rm {}
ubuntu@ubuntu:~/$ ls -als ./backup/ssh/recov* | grep " 32768" | awk '{print $10}' | xargs -I {} rm {}
```

Fingerprinting SSH public keys

```
$ sudo ssh-keygen -B -f /media/ubuntu/a733ad08-1f25-44b5-8dc6-38336ef027e7/backup/clés_lucasg.pub
4096 xuheg-bilud-bunoz-zocyz-codaz-dikic-zydel-herib-bipiz-turar-dixix lucasg@lucasg (RSA)
4096 xivin-buzih-decol-cyguz-vahec-dugut-repyv-hyzab-zygen-tudub-nexax lucasg@lucasg (RSA)
4096 xubep-pyvus-meren-doryd-bobiz-bomib-zidom-ravat-ducyt-manas-doxux lucasg@lucasg (RSA)
256  xecap-vyset-gusor-rukyr-ryhuh-kyvev-vidir-zudyd-givyf-mepel-zaxax lucasg@lucasg (ED25519)
256  xicif-pevok-caluc-putut-tanaf-lezaz-syzad-mubuz-dacov-kysyd-tyxax lucasg@lucasg (ED25519-SK)
```

Fingerprinting recovered SSH private keys

```
$ find ./backup/ssh/recovered_ssh_key_* | while read filepath ; do sudo ssh-keygen -B -f "$filepath" 2>/dev/null; done | sort -n | uniq
256 xebal-zogez-litep-bamaz-lagud-pubyt-reluz-vazoh-sagaz-gecuf-cuxox root@500a2f1385e9 (ED25519)
256 xecaz-vipec-coseg-pusik-bumyn-kerom-benem-gynyv-homuv-mebyl-fyxex nt authority\system@WinDev1811Eval (ED25519)
256 xupir-hosyk-moniz-hamab-damal-fakit-luhas-lihes-bebeg-syhav-gyxex root@photon-machine (ECDSA)
521 xupim-fuloz-lyzom-disez-tutan-hanag-vyvov-pynyz-filuk-fihit-kexox ettore@localhost.localdomain (ECDSA)
2048 xelom-higyp-zovuk-zuzoz-dazip-hilun-lyvol-rerah-cimeh-lopud-syxix kami@kami-dell-latitude (RSA)
2048 xerig-nudik-tezob-nuzoz-tovog-dezyc-bihim-guvov-pycez-zohap-dyxux lucasg (RSA)
2048 xutor-zykik-zured-rekut-mesat-gunod-makoc-sydum-teget-zybap-lixax root@debian (RSA)
3072 xovez-sapev-furud-robop-synin-dirid-tasyr-rolaz-pepyh-hakuv-zyxix root@ee23c13d64e9 (RSA)
...
```

Take a good look at private key filesizes

```
$ sudo ssh-keygen -B -f /home/ubuntu/.ssh/test_4096_rsa
4096 xogob-nipar-hetiz-zibir-lozym-gogeh-dibib-cusom-zumin-cibyn-gixyx root@ubuntu (RSA)
$ ls -als /home/ubuntu/.ssh/test_4096_rsa
4 -rw------- 1 root root 3434 Oct 2 09:33 /home/ubuntu/.ssh/test_4096_rsa

$ ls -als ./backup/ssh/recovered_ssh_key_* | awk '{print $6 " " $10}' | grep 3434      # rsa 4096
3434 ./backup/ssh/recovered_ssh_key_c59f6a5000.key
3434 ./backup/ssh/recovered_ssh_key_c5a37fe000.key
3434 ./backup/ssh/recovered_ssh_key_c5aa130000.key

$ ls -als ./backup/ssh/recovered_ssh_key_* | awk '{print $6 " " $10}' | grep 602       # ed55219-sk
602 ./backup/ssh/recovered_ssh_key_c5912d6000.key

$ ls -als ./backup/ssh/recovered_ssh_key_* | awk '{print $6 " " $10}' | grep 444       # ed55219
444 ./backup/ssh/recovered_ssh_key_c591374000.key
```

Just bruteforce it

```
$ lucasg@lazarus:~$ find ./ssh/ | while read file;do \
    ssh-keygen -p -P "toto" -N "" -f $file;\
done
Failed to load key ./ssh/recovered_ssh_encrypted_key_AAAAAA: incorrect passphrase supplied
      to decrypt private key
...
Saving key "./ssh/recovered_ssh_encrypted_key_BBBBB" failed: Permission denied.
...
```

Warning

If you're not careful with the previous command you can leak your SSH passphrase into your `.bash_history`.
Put a space before the command to prevent logging, or change your SSH passphrase again once you've identified which private key correspond to which passphrase.

Conclusion

GOOD

- Credentials
  - GPG key carved back
  - other authentication secrets
- Backups
  - Recent backup of my keepass
  - Backup of training found
  - GPG'ed backups can be deciphered
- "Nice to have"
  - SSH keys reconstructed

Conclusion

BAD

- 3 days lost for forensic + 2 days for re-setup
- Partial update of training : 5 days x 2 people.

Conclusion

BAD

- 3 days lost for forensic + 2 days for re-setup
- Partial update of training : 5 days x 2 people.

UGLY

- Lost nearly all my presentation's slides
- Lost all my `README.md` and `TODO.md` files
- Lost almost all my "personal" projects and dotfiles

Conclusion

- Data loss/Burning Datacenters/Ransomware attacks are traumatic events
  - Get help from friends and family
  - Also get help from professionals :)
- Explicitely prepare a Disaster Recovery Plan
  - for yourself
  - for your organization
- Encrypt your archives with a secondary recovery key (luks, gpg, etc.)

Plus d'informations sur :
https://www.synacktiv.com/publications/rm-rf-is-the-root-of-all-evil

MERCI DE VOTRE ATTENTION

SYNACKTIV