

Lab8

ID: 11912614

Name: 张睿豪

Q1 & Q2 are finished with my project members: 谢岳臻, 李家奥, 刘晟淇, 王奕童

Q1

Prove that you have replaced the kernel

Before replacement

```
Linux raspberrypi 5.4.51-07+ #1333 SMP Mon Aug 10 16:45:19 BST 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

pi@raspberrypi:~$ uname -r
5.4.51-07+
pi@raspberrypi:~$
```

After replacement

```
The exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
-bash: warning: setlocale: LC_ALL: cannot change locale (en_US.UTF-8)

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

pi@raspberrypi:~$ uname -r
4.14.114-07+
pi@raspberrypi:~$
```

Prove that you have built the nailgun module with new headers

We compile the nailgun module successfully

```
(base) zrh@zrh-Lenovo-Legion-Y7000P2021:/media/zrh/RepoWin/Academic/computer_security/week10/nailgundefense/Read_SCR$ cd .
(base) zrh@zrh-Lenovo-Legion-Y7000P2021:/media/zrh/RepoWin/Academic/computer_security/week10/nailgundefense/Read_SCR$ vim Makefile
(base) zrh@zrh-Lenovo-Legion-Y7000P2021:/media/zrh/RepoWin/Academic/computer_security/week10/nailgundefense/Read_SCR$ make
make ARCH=arm -C /home/zrh/Repository/gitrepo/modulespath/lib/modules/4.14.114-v7+/build M=/media/zrh/RepoWin/Academic/computer_security/week10/nailgundefense/Read_SCR CROSS_COMPILE=/home/zrh/Repository/gitrepo/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian-x64/bin/arm-linux-gnueabihf- modules
make[1]: 进入目录"/home/zrh/Repository/gitrepo/linux"
CC [M] /media/zrh/RepoWin/Academic/computer_security/week10/nailgundefense/Read_SCR/nailgun.o
/media/zrh/RepoWin/Academic/computer_security/week10/nailgundefense/Read_SCR/nailgun.c: 在函数'nailgun_init'中:
/media/zrh/RepoWin/Academic/computer_security/week10/nailgundefense/Read_SCR/nailgun.c:222:5: 警告: ISO C90 不允许混合使用声明和代码 [-Wdeclaration-after-statement]
    struct nailgun_param *param = kmalloc(sizeof(t_param), GFP_KERNEL);
    ^
Building modules, stage 2.
MODPOST 1 modules
CC /media/zrh/RepoWin/Academic/computer_security/week10/nailgundefense/Read_SCR/nailgun.mod.o
LD [M] /media/zrh/RepoWin/Academic/computer_security/week10/nailgundefense/Read_SCR/nailgun.ko
make[1]: 离开目录"/home/zrh/Repository/gitrepo/linux"
(base) zrh@zrh-Lenovo-Legion-Y7000P2021:/media/zrh/RepoWin/Academic/computer_security/week10/nailgundefense/Read_SCR$ uname -r
5.15.0-52-generic
(base) zrh@zrh-Lenovo-Legion-Y7000P2021:/media/zrh/RepoWin/Academic/computer_security/week10/nailgundefense/Read_SCR$ ls
Makefile  Module.symvers  nailgun.ko  nailgun.mod.o
modules.order  nailgun.c  nailgun.mod.c  nailgun.o
```

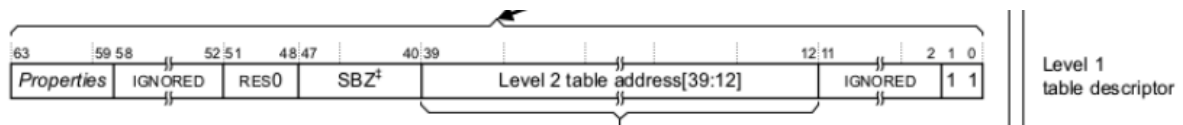
Q2

The Nailgun Attack works.

```
[ 4.500040] systemd[1]: Proceeding without the setting in effect! (this warning is only shown for the first
[ 4.838067] systemd[1]: /etc/systemd/system/teamviewerd.service:8: PIDFile= references path below legacy di
cordingly.
[ 5.140238] random: systemd: uninitialized urandom read (16 bytes read)
[ 5.180212] random: systemd: uninitialized urandom read (16 bytes read)
[ 5.198048] systemd[1]: Listening on udev Control Socket.
[ 5.225742] random: systemd: uninitialized urandom read (16 bytes read)
[ 5.243442] systemd[1]: Listening on Journal Socket (/dev/log).
[ 5.284186] systemd[1]: Condition check resulted in Journal Audit Socket being skipped.
[ 5.304136] systemd[1]: Set up automount Arbitrary Executable File Formats File System Automount Point.
[ 6.213902] EXT4-fs (mmcblk0p2): re-mounted. Opts: (null)
[ 6.372883] systemd-journald[106]: Received request to flush runtime journal from PID 1
[ 9.283450] random: crng init done
[ 9.283459] random: 7 urandom warning(s) missed due to ratelimiting
[ 9.612706] uart-pl011 3f201000.serial: no DMA platform data
[ 10.201186] Adding 102396k swap on /var/swap. Priority:-2 extents:1 across:102396k SSFS
[ 341.810509] nailgun: loading out-of-tree module taints kernel.
[ 341.811013] Nailgun Attack Start
[ 341.811066] Using smp_call_function
[ 341.811080] Step 1: Unlock debug and cross trigger registers
[ 341.811088] Step 2: Enable halting debug
[ 341.811092] Step 3: Halt the target processor
[ 341.811098] Step 4: Wait the target processor to halt
[ 341.811102] Step 5: Save context
[ 341.811108] Step 6: Switch to EL3
[ 341.811113] Step 7: Read SCR
[ 341.811118] Step 8: Restore context
[ 341.811124] Step 9: Send restart request to the target processor
[ 341.811129] Step 10: Wait the target processor to restart
[ 341.811136] All done! The value of SCR is 0x00000131
pi@raspberrypi:~/Desktop$ cat /dev/ptmx &
```

Q3 & Q4

The translation table base address is VTTBR: 0x32000000



As the picture above shows, size of a descriptor is 2^3 B.

According to structure of IPA, I know followings:



1. The number of entries for each page table:

- level-1 page table: $2^{31-30+1} = 2^2$
- level-2 page table: $2^{29-21+1} = 2^9$
- level-3 page table: $2^{20-12+1} = 2^9$

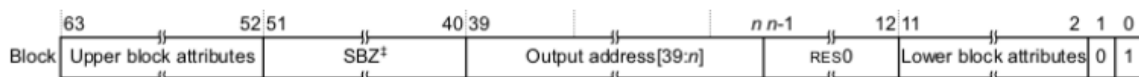
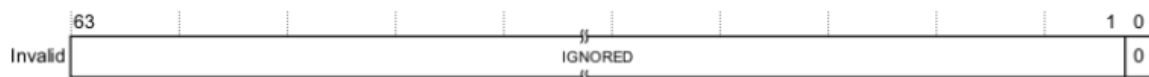
2. Space (B) needed to store each level page table:

- level-1 page table
Since $2^3 \times 2^2 = 2^5 < 4KB$, it needs $4KB = 2^{12}B$
- level-2 and level-3 page table
 $2^3 \times 2^9 = 2^{12}B$

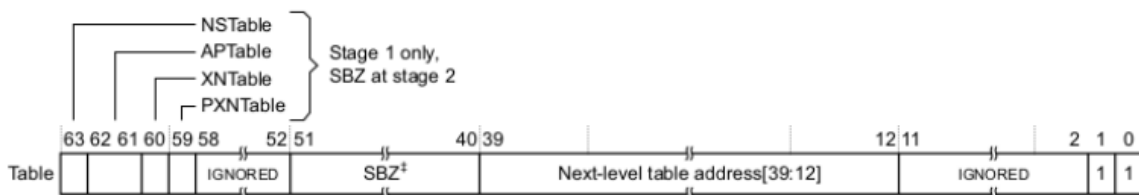
Design

Descriptor's value

1. For level-1 & level-2 descriptor:



For the level 1 descriptor, n is 30. For the level 2 descriptor, n is 21.

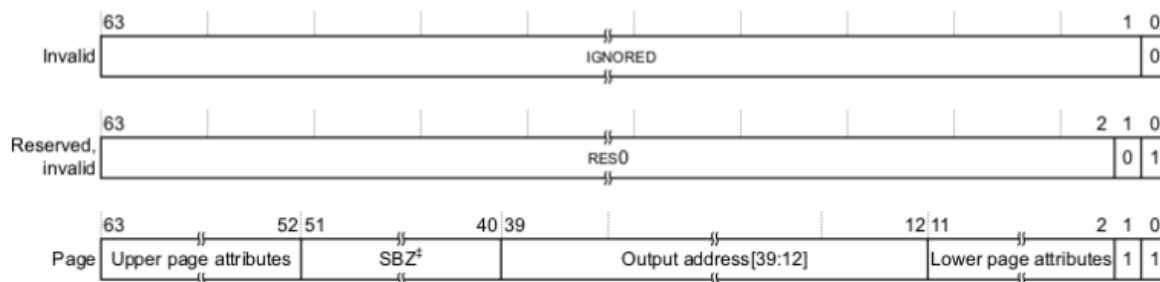


The level 1 descriptor returns the address of the level 2 table.
The level 2 descriptor returns the address of the level 3 table.

For simplicity, if it is

- invalid: value is 0
- block: only care about output address, value is output address $\ll 30 + 1$ for level 1 and output address $\ll 21 + 1$ for level 2
- table: only care about next-level table address, value is next-level table base address $\ll 12 + 3$

2. For level-3 descriptor:



For simplicity, if it is

- invalid: value is 0
- page: only care about output address, value is output address $\ll 12 + 3$

Address range of each page table

1. Level-1 Page Table: $[0x3200\ 0000, 0x3200\ 0000 + 2^{12}) =$

$$[0x32000000, 0x32001000)$$

Base of level-1 page is VTTBR.

2. Level-2 Page Table: $[0x3200\ 1000, 0x3200\ 1000 + 2^2 \times 2^{12}) =$

$$[0x32001000, 0x32005000)$$

Base of level-2 page is $0x3200\ 1000$, called BA2.

3. Level-3 Page Table:

Theoretically, the range should be $[0x3200\ 5000, 0x3200\ 5000 + 2^2 \times 2^9 \times 2^{12}) = [0x3200\ 1000, 0x3280\ 5000)$ but $0x3280\ 5000 > 0x321f\ ffff$ (upper bound of the reserved space). Therefore, actual range is

$$[0x32005000, 0x321fffff)$$

Base of level-3 page is $0x0x3200\ 5000$, called BA3.

Q3

What I know:

- VTTBR: $0x3200\ 0000$
- input IPA: $0x40030614 = 0b0100\ 0000\ 0000\ 0011\ 0000\ 0110\ 0001\ 0100$
 - Level-1 Page Table Number: $IPA[31:30] = 0b01$
 - Level-2 Page Table Number: $IPA[29:21] = 0b0$
 - Level-3 Page Table Number: $IPA[20:12] = 0b0011\ 0000$
 - offset (4kb): $IPA[11:0] = 0b0110\ 0001\ 0100$
- output PA (PA== IPA): $0x40030614 = 0x40030 << 12 + 0x614$

According to my design,

1. level 1:

- address of descriptor:
 $\text{concat}(\text{VTTBR}[39:5], \text{IPA}[31:30], 0b000) = 0x3200\ 0008$
- value of descriptor:
it points to the second (index is $\text{IPA}[31:30]==1$) level-2 table, so address of the table is
 $A2 = (\text{BA2} + 1 \times 2^{12}) = 0x3200\ 2000$

$$A2[39:12] << 12 + 3 = 0x3200\ 2003$$

2.

2. level 2 (base $0x3200\ 1000$):

- address of descriptor:
 $\text{concat}(A2[39:12], \text{IPA}[29:21], 0b000) = 0x3200\ 2000$
- value of descriptor:
it points to the first (index is $\text{IPA}[29:21]==0$) level-3 table, so address of the table is
 $A3 = (\text{BA3} + 0 \times 2^{12}) = 0x3200\ 5000$

$$A3[39:12] << 12 + 3 = 0x3200\ 5003$$

3. level 3 (base $0x3200\ 5000$):

- address of descriptor:
 $\text{concat}(A3[39:12], \text{IPA}[20:12], 0b000) = 0x3200\ 5180$
- value of descriptor:
it points to output address PA, so the value is

$$PA[39:12] \ll 12 + 3 = 0x4003\ 0003$$

Q4

What I know:

- VTTBR: 0x3200 0000
- input IPA: 0x41912614 = 0b0100 0001 1001 0001 0010 0110 0001 0100
 - Level-1 Page Table Number: IPA[31:30] = 0b01
 - Level-2 Page Table Number: IPA[29:21] = 0b1100
 - Level-3 Page Table Number: IPA[20:12] = 0b1 0001 0010
 - offset (4kb): IPA[11:0] = 0b0110 0001 0100
- output PA (PA== IPA): 0x41912614 = 0x41912 << 12 + 0x614

According to my design,

1. level 1:

- address of descriptor:

$$\text{concat}(\text{VTTBR}[39:5], \text{IPA}[31:30], 0b000) = 0x3200\ 0008$$

- value of descriptor:

it points to the second (index is IPA[31:30]==1) level-2 table, so address of the table is
 $A2 = (BA2 + 1 \times 2^{12}) = 0x3200\ 2000$

$$A2[39:12] \ll 12 + 3 = 0x3200\ 2003$$

2. level 2 (base 0x3200 1000):

- address of descriptor:

$$\text{concat}(A2[39:12], \text{IPA}[29:21], 0b000) = 0x3200\ 2060$$

- value of descriptor:

it points to the 12th (index is IPA[29:21]==12) level-3 table, so address of the table is
 $A3 = (BA3 + 12 \times 2^{12}) = 0x3201\ 1000$

$$A3[39:12] \ll 12 + 3 = 0x3201\ 1003$$

3. level 3 (base 0x3200 5000):

- address of descriptor:

$$\text{concat}(A3[39:12], \text{IPA}[20:12], 0b000) = 0x3201\ 1890$$

- value of descriptor:

it points to output address PA, so the value is

$$PA[39:12] \ll 12 + 3 = 0x4191\ 2003$$