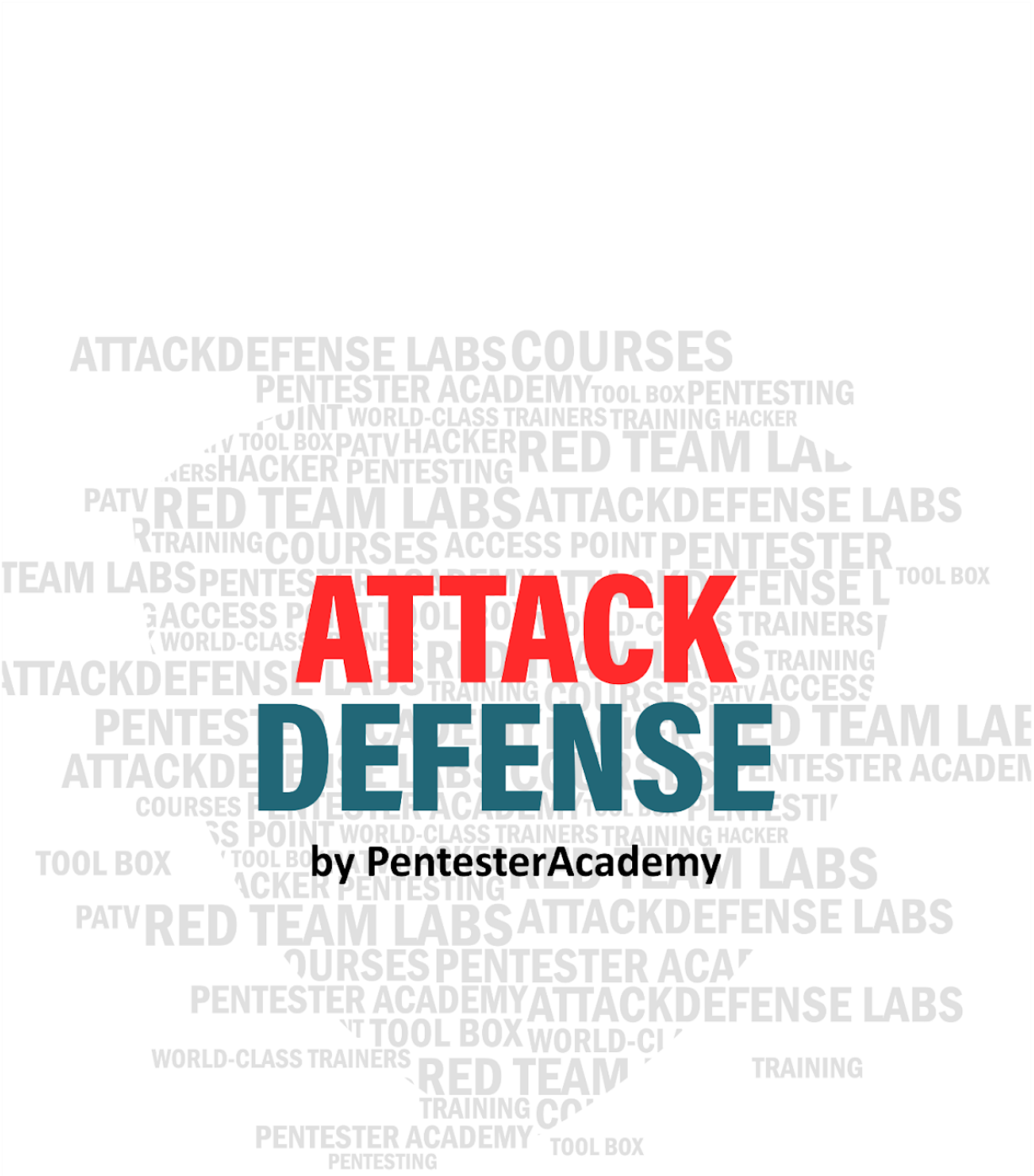
ATTACK
DEFENSE
by PentesterAcademy

Name	Local Boot using U-boot
URL	https://www.attackdefense.com/challengedetails?cid=1234
Type	IoT : Bootloader

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic.

Objective: Boot the machine using the files available on SD card and check the contents of the file system.

Step 1: One can observe that on lab start, u-boot runs and fails to find the location of kernel. This gives the user u-boot console.

```
TFTP error: trying to overwrite reserved memory...
smc911x: MAC 52:54:00:12:34:56
Wrong Image Format for bootm command
ERROR: can't get kernel image!
=>
```

It is given in the challenge statement that SD card is mounted to this machine. List available devices

Command: mmc list

```
=> mmc list
MMC: 0 (SD)
=>
```

One SD card is attached to it.

Step 2: Check the contents of this SD card.

Command: ext2ls mmc 0:1

Here,

- mmc : Device type
- 0 : Device number
- 1 : Partition number

```
=> ext2ls mmc 0:1
<DIR>       1024 .
<DIR>       1024 ..
<DIR>      12288 lost+found
<DIR>       3072 bin
<DIR>       1024 dev
<DIR>       1024 etc
<DIR>       1024 lib
<SYM>          3 lib32
<SYM>         11 linuxrc
<DIR>       1024 media
<DIR>       1024 mnt
<DIR>       1024 opt
<DIR>       1024 proc
<DIR>       1024 root
<DIR>       1024 run
<DIR>       1024 sbin
<DIR>       1024 sys
<DIR>       1024 tmp
<DIR>       1024 usr
<DIR>       1024 var
         4181384 zImage
           14430 vexpress-v2p-ca9.dtb
```

The kernel file and Device Tree Blob file is present on SD card.

Step 3: Check the board information get the memory information and address range.

Command: bdinfo

```
=> bdinfo
arch_number = 0x000008e0
boot_params = 0x60002000
DRAM bank   = 0x00000000
-> start    = 0x60000000
-> size     = 0x20000000
DRAM bank   = 0x00000001
-> start    = 0x80000000
-> size     = 0x00000004
eth0name    = smc911x-0
ethaddr     = 52:54:00:12:34:56
current eth = smc911x-0
ip_addr     = <NULL>
baudrate    = 38400 bps
TLB addr    = 0x7fff0000
relocaddr   = 0x7ff8b000
reloc off   = 0x1f78b000
irq_sp      = 0x7fe8aee0
sp start    = 0x7fe8aed0
=>
```

Step 4: Load kernel and Device Tree Blob file to valid memory addresses. Also, make sure that they don't overwrite each other.

Commands:
ext2load mmc 0:1 0x61000000 zImage
ext2load mmc 0:1 0x61a00000 vexpress-v2p-ca9.dtb

```
=> ext2load mmc 0:1 0x61000000 zImage
4181384 bytes read in 2830 ms (1.4 MiB/s)
=> ext2load mmc 0:1 0x61a00000 vexpress-v2p-ca9.dtb
14430 bytes read in 384 ms (36.1 KiB/s)
```

Step 5: Set kernel parameters.

Command: setenv bootargs 'console=ttyAMA0 console=tty0 root=/dev/mmcblk0p1 rw'

```
=> setenv bootargs 'console=ttyAMA0 console=tty0 root=/dev/mmcblk0p1 rw'
```

Step 6: Finally, machine is ready to boot.

Command: bootz 0x61000000 - 0x61a00000

```
=> bootz 0x61000000 - 0x61a00000
Kernel image @ 0x61000000 [ 0x000000 - 0x3fcd88 ]
## Flattened Device Tree blob at 61a00000
   Booting using the fdt blob at 0x61a00000
   Loading Device Tree to 7fe83000, end 7fe8985d ... OK

Starting kernel ...

Booting Linux on physical CPU 0x0
```

The machine will start and after going through boot sequence, eventually present console login to the user. The user has to use the following credentials:

Username: root
Password: <none>

```
Welcome to Buildroot
buildroot login: root
#
```

After logging into the machine, the user can run common Linux commands.

Command: ps

```
# ps
PID   USER     COMMAND
    1 root     init
    2 root     [kthreadd]
    3 root     [rcu_gp]
    4 root     [rcu_par_gp]
    5 root     [kworker/0:0-eve]
    6 root     [kworker/0:0H]
```

In this manner, a machine can be booted from u-boot and SD card.

References:

- U-boot source: https://www.denx.de/wiki/U-Boot/SourceCode