Emulating the Firmware using FAT Script

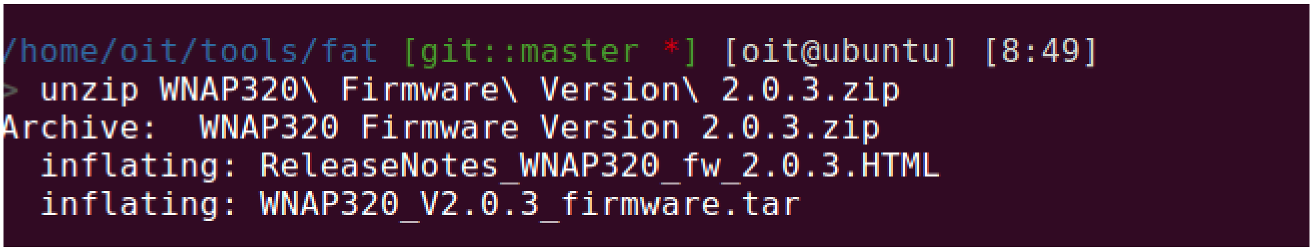
- Why emulate a complete firmware?
- Emulating using FAT
- Bring the device to the network as a real physical device
- Run additional scanning tools and scripts to identify more vulnerabilities

Lab File : WNAP320 Firmware Version 2.0.3.zip

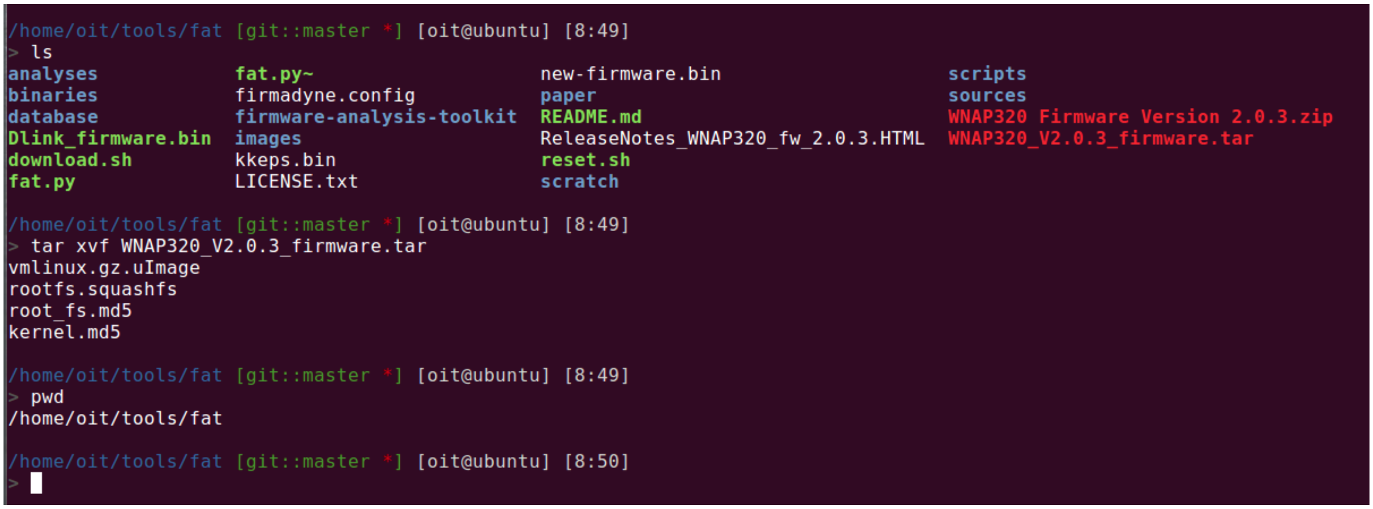
Navigate to **/home/oit/tools/fat**

#ls

#unzip WNAP320\ Firmware\ Version\ 2.0.3.zip



```
/home/oit/tools/fat [git::master *] [oit@ubuntu] [8:49]
> unzip WNAP320\ Firmware\ Version\ 2.0.3.zip
Archive:  WNAP320 Firmware Version 2.0.3.zip
  inflating: ReleaseNotes_WNAP320_fw_2.0.3.HTML
  inflating: WNAP320_V2.0.3_firmware.tar
```

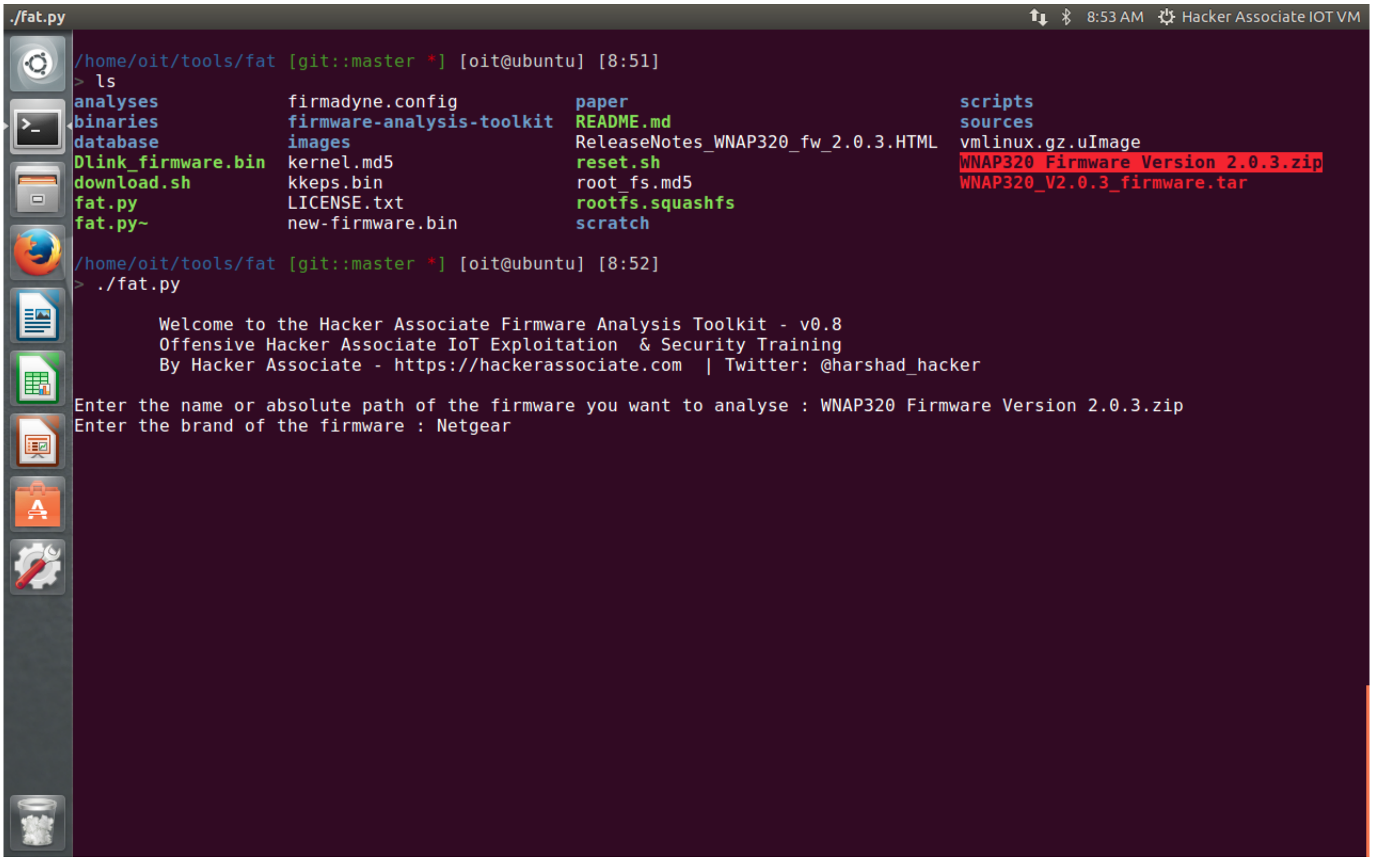


```
/home/oit/tools/fat [git::master *] [oit@ubuntu] [8:49]
> ls
analyses            fat.py~                   new-firmware.bin                    scripts
binaries            firmadyne.config          paper                               sources
database            firmware-analysis-toolkit README.md                           WNAP320 Firmware Version 2.0.3.zip
Dlink_firmware.bin  images                    ReleaseNotes_WNAP320_fw_2.0.3.HTML  WNAP320_V2.0.3_firmware.tar
download.sh         kkeps.bin                 reset.sh
fat.py              LICENSE.txt               scratch

/home/oit/tools/fat [git::master *] [oit@ubuntu] [8:49]
> tar xvf WNAP320_V2.0.3_firmware.tar
vmlinux.gz.uImage
rootfs.squashfs
root_fs.md5
kernel.md5

/home/oit/tools/fat [git::master *] [oit@ubuntu] [8:49]
> pwd
/home/oit/tools/fat

/home/oit/tools/fat [git::master *] [oit@ubuntu] [8:50]
> 
```

Emulating Firmware using FAT Script

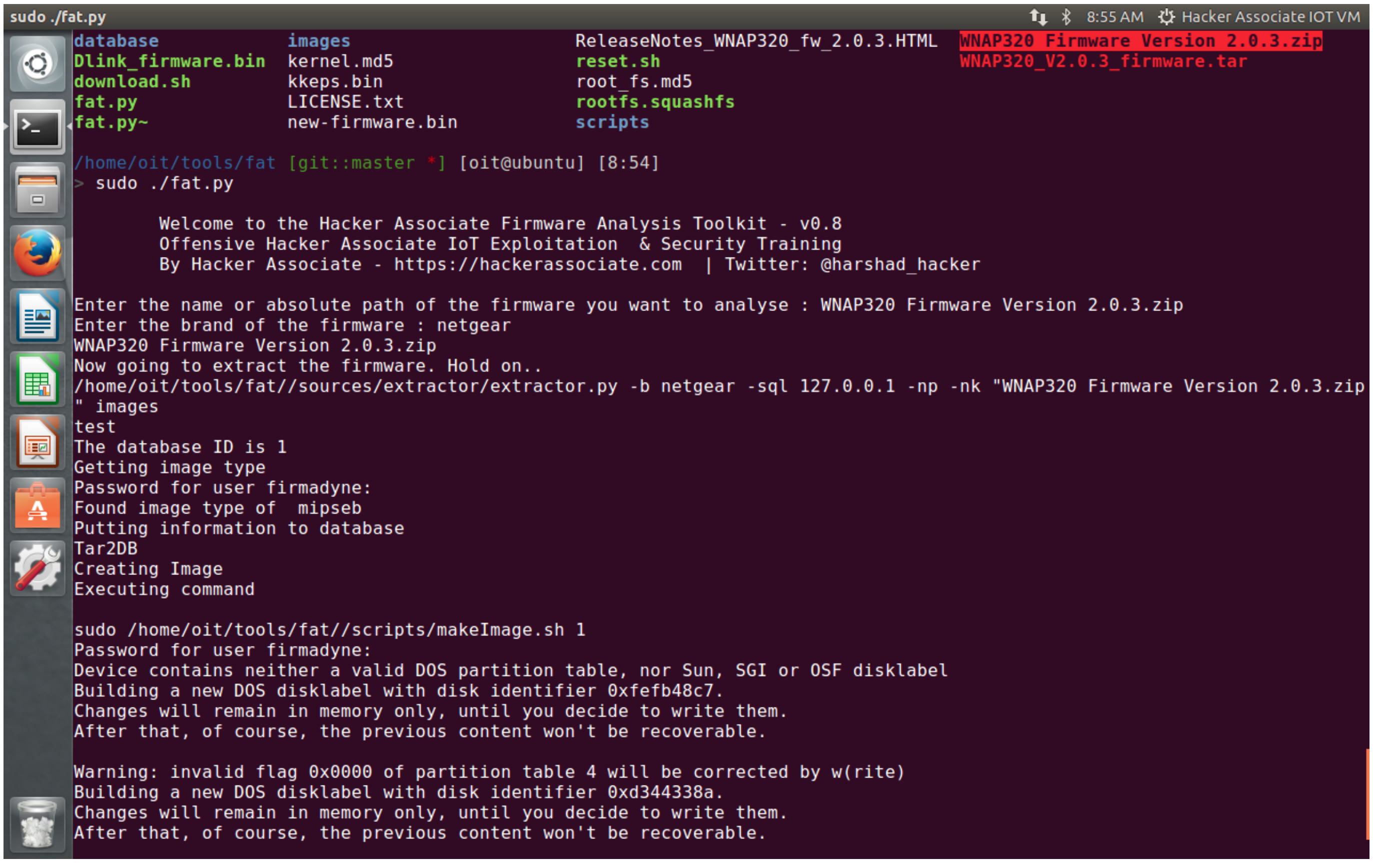
/home/oit/tools/fat [git::master *] [oit@ubuntu] [8:52]

```
./fat.py
```



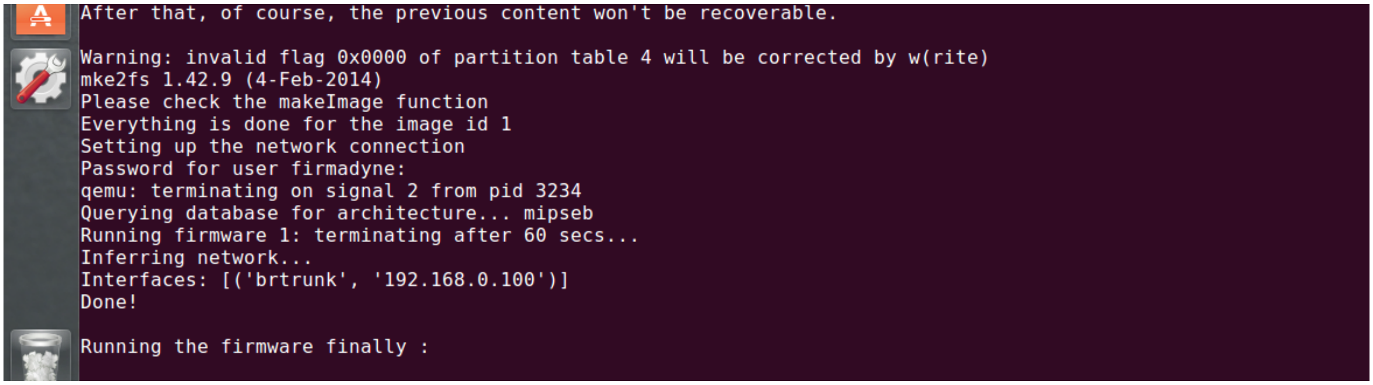
```
/home/oit/tools/fat [git::master *] [oit@ubuntu] [8:51]
> ls
analyses            firmadyne.config            paper                             scripts
binaries            firmware-analysis-toolkit   README.md                         sources
database            images                      ReleaseNotes_WNAP320_fw_2.0.3.HTML vmlinux.gz.uImage
Dlink_firmware.bin  kernel.md5                  reset.sh                          WNAP320 Firmware Version 2.0.3.zip
download.sh         kkeps.bin                   root_fs.md5                       WNAP320_V2.0.3_firmware.tar
fat.py              LICENSE.txt                 rootfs.squashfs
fat.py~             new-firmware.bin            scratch

/home/oit/tools/fat [git::master *] [oit@ubuntu] [8:52]
> ./fat.py

        Welcome to the Hacker Associate Firmware Analysis Toolkit - v0.8
        Offensive Hacker Associate IoT Exploitation  & Security Training
        By Hacker Associate - https://hackerassociate.com  | Twitter: @harshad_hacker

Enter the name or absolute path of the firmware you want to analyse : WNAP320 Firmware Version 2.0.3.zip
Enter the brand of the firmware : Netgear
```

Getting IP



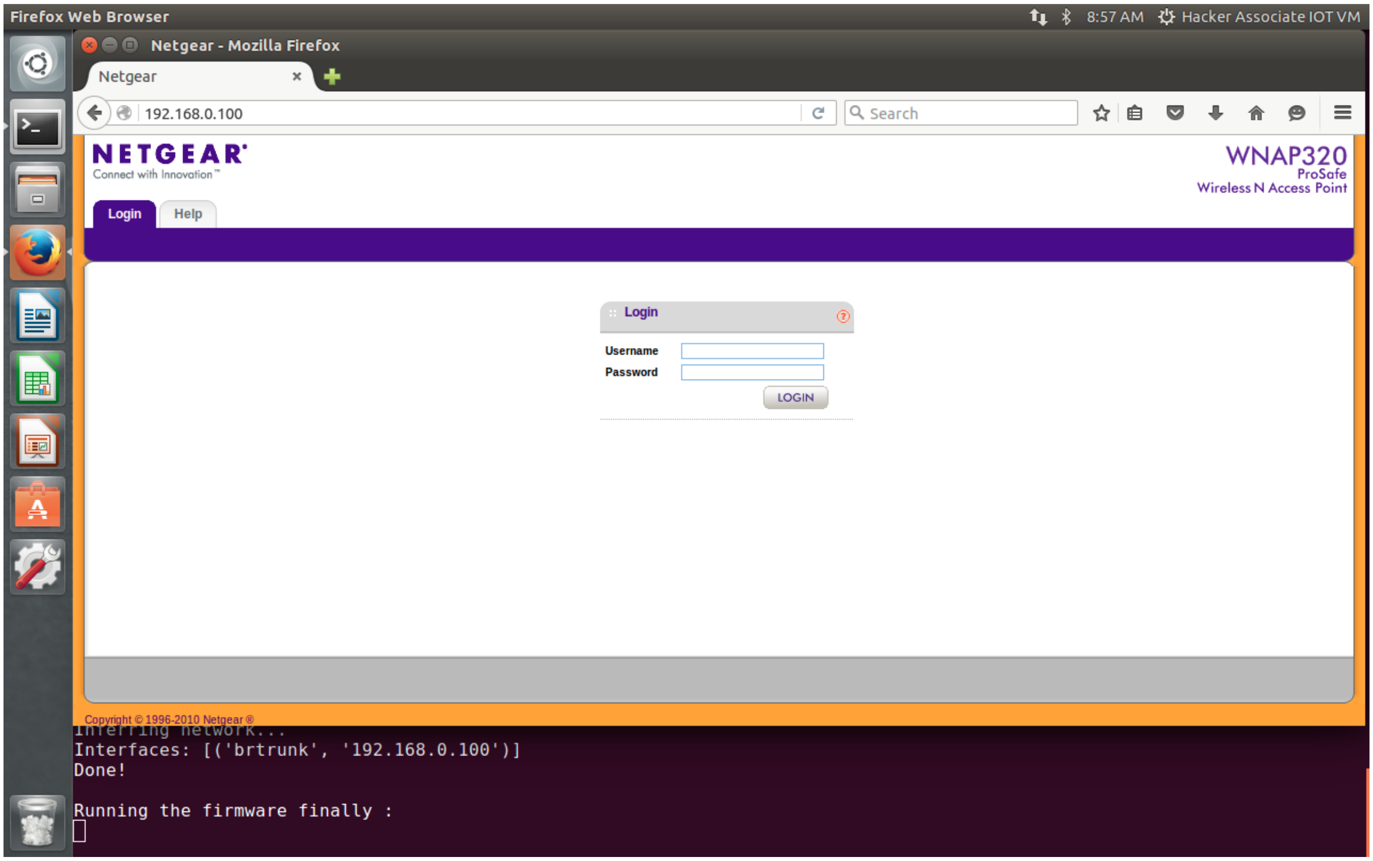
```
database            images                      ReleaseNotes_WNAP320_fw_2.0.3.HTML WNAP320 Firmware Version 2.0.3.zip
Dlink_firmware.bin  kernel.md5                  reset.sh                          WNAP320_V2.0.3_firmware.tar
download.sh         kkeps.bin                   root_fs.md5
fat.py              LICENSE.txt                 rootfs.squashfs
fat.py~             new-firmware.bin            scripts

/home/oit/tools/fat [git::master *] [oit@ubuntu] [8:54]
> sudo ./fat.py

        Welcome to the Hacker Associate Firmware Analysis Toolkit - v0.8
        Offensive Hacker Associate IoT Exploitation  & Security Training
        By Hacker Associate - https://hackerassociate.com  | Twitter: @harshad_hacker

Enter the name or absolute path of the firmware you want to analyse : WNAP320 Firmware Version 2.0.3.zip
Enter the brand of the firmware : netgear
WNAP320 Firmware Version 2.0.3.zip
Now going to extract the firmware. Hold on..
/home/oit/tools/fat//sources/extractor/extractor.py -b netgear -sql 127.0.0.1 -np -nk "WNAP320 Firmware Version 2.0.3.zip
" images
test
The database ID is 1
Getting image type
Password for user firmadyne:
Found image type of  mipseb
Putting information to database
Tar2DB
Creating Image
Executing command

sudo /home/oit/tools/fat//scripts/makeImage.sh 1
Password for user firmadyne:
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0xfefb48c7.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)
Building a new DOS disklabel with disk identifier 0xd344338a.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.
```

Got IP = 192.168.0.100



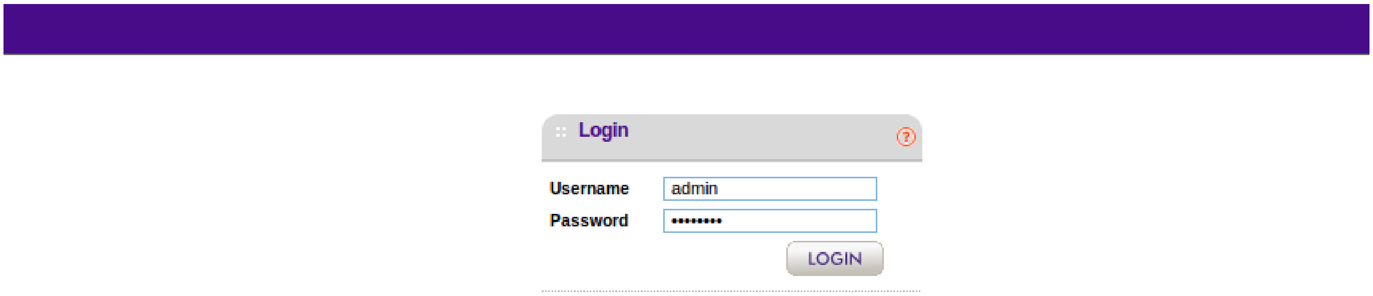
```
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)
mke2fs 1.42.9 (4-Feb-2014)
Please check the makeImage function
Everything is done for the image id 1
Setting up the network connection
Password for user firmadyne:
qemu: terminating on signal 2 from pid 3234
Querying database for architecture... mipseb
Running firmware 1: terminating after 60 secs...
Inferring network...
Interfaces: [('brtrunk', '192.168.0.100')]
Done!

Running the firmware finally :
```

Open in Browser and get the web console



```
Inferring network...
Interfaces: [('brtrunk', '192.168.0.100')]
Done!

Running the firmware finally :
```

```
Note: We can brute force username and password but default user is admin and
pass is password

User = admin
Pas  = password
```

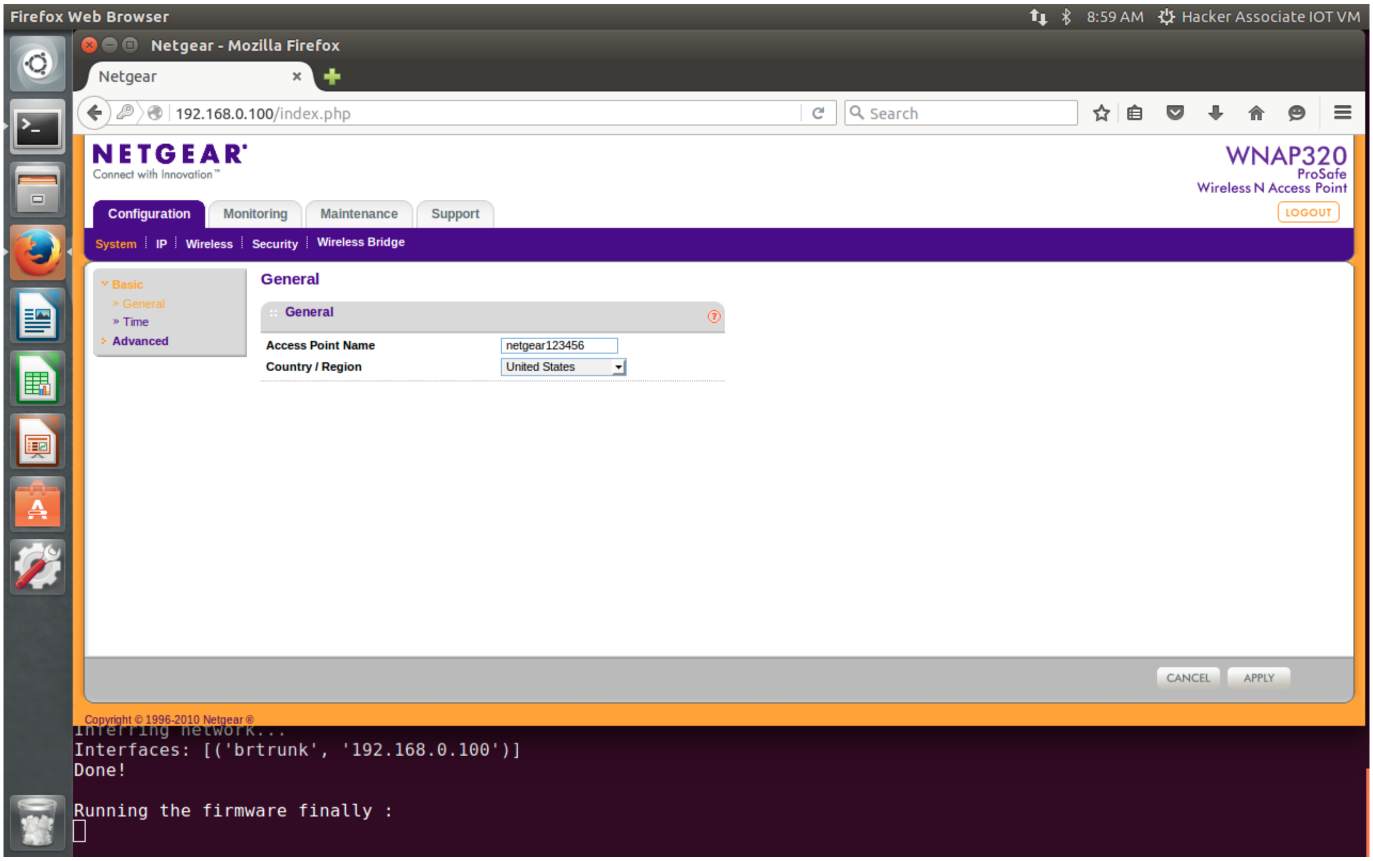


Login

Username: admin

Password: ••••••••

Examine Web Console



General

Access Point Name: netgear123456

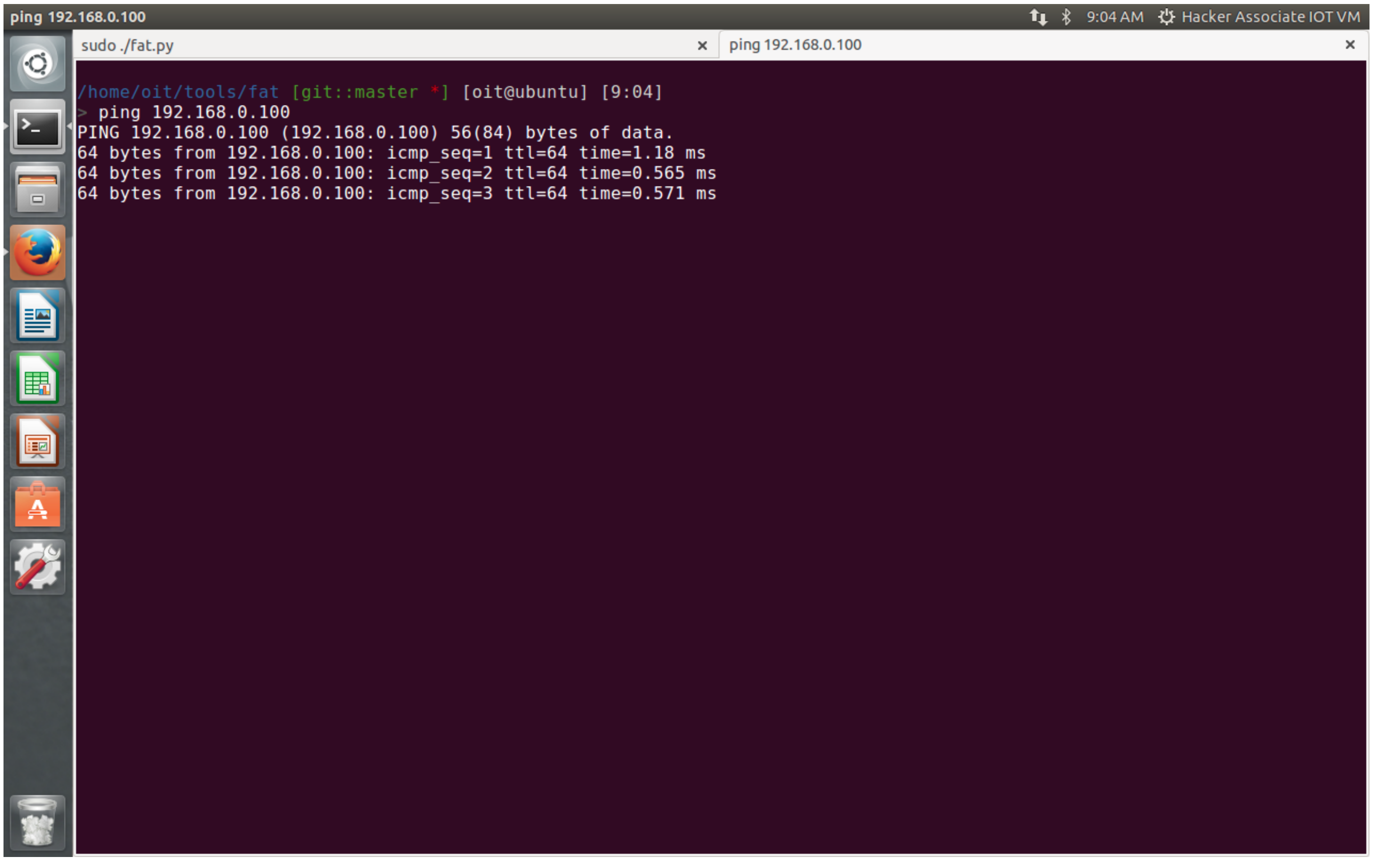
Country / Region: United States

```
Inferring network...
Interfaces: [('brtrunk', '192.168.0.100')]
Done!

Running the firmware finally :
```

Bring the device to the network as a real physical device

#ping 192.168.0.100

Now we can apply all hacking phases & able to perform all sorts of attacks.



```
/home/oit/tools/fat [git::master *] [oit@ubuntu] [9:04]
> ping 192.168.0.100
PING 192.168.0.100 (192.168.0.100) 56(84) bytes of data.
64 bytes from 192.168.0.100: icmp_seq=1 ttl=64 time=1.18 ms
64 bytes from 192.168.0.100: icmp_seq=2 ttl=64 time=0.565 ms
64 bytes from 192.168.0.100: icmp_seq=3 ttl=64 time=0.571 ms
```

Congratulations!, you have successfully finished ***"Hacker Associate Emulating Firmware Lab"***

```
Thanks and Regards
Harshad Shah
Founder & CEO, Hacker Associate
```