CEN 3078 Computer Security
Spring 2025
Lab # 5 – OpenWrt

Chengyi Qu (cqu@fgcu.edu)

1. Purpose of the Lab

In this lab, you will understand the OpenWrt project and learn how to use an OpenWrt based router. You will be able to use a storage device (USB or sata or SDcard or any other removable storage drive) to expand your OpenWrt/LEDE (Linux Embedded Development Environment) device's space in the root filesystem, to install freely all the packages you need and to collect data. You will also be able to capture, filter and inspect packets using tcpdump and Wireshark tools on OpenWrt/LEDE router.

2. References to guide Lab work

[1] OpenWrt project main page: https://OpenWrt.org/
[2] Quick Start for Adding a USB drive on OpenWrt: https://OpenWrt.org/docs/guide-user/storage/usb-drives-quickstart
[3] OpenWrt extroot configuration: https://OpenWrt.org/docs/guide-user/additional-software/extroot_configuration
[4] Tcpdump or Wireshark tools configuration and OpenWrt configuration: https://OpenWrt.org/docs/guide-user/firewall/misc/tcpdump_wireshark
[5] Getting start with OpenWrt: https://www.ayomaonline.com/security/getting-started-with-OpenWrt-linuxfying-routers/
[6] Tcpdump usage examples: https://www.rationallyparanoid.com/articles/tcpdump.html
[7] Extroot configuration: https://openwrt.org/docs/guide-user/additional-software/extroot_configuration
[8] DNS spoofing with Dnsmasq: https://www.linux.com/learn/intro-to-linux/2017/7/dns-spoofing-dnsmasq

3. Lab Steps and output collection guidelines

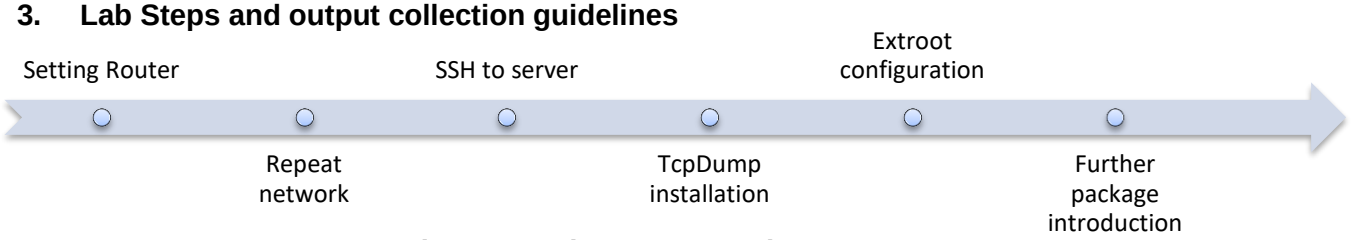

Figure 1: Lab Steps Overview

Figure 1 shows the required steps to be followed to have a brief understanding of OpenWrt project and some settings on extroot configuration and network sniffing. You will need an erased USB drive device to finish this lab. After this lab, you will be able to turn your router in a fully capable GNU/Linux computer that is always powered on and even do some hacking operations through the router.

Let's get started!

3.1 OpenWrt

The OpenWrt Project is a Linux operating system targeting embedded devices. Instead of trying to create a single, static firmware, OpenWrt provides a fully writable filesystem with package management. This frees you from the application selection and configuration provided by the vendor and allows you to customize the device using packages to suit any application. For developers, OpenWrt is the framework to build an application without having to build a complete firmware around it; for users, this means the ability for full customization, to use the device in ways never envisioned.

OpenWrt is a highly extensible GNU/Linux distribution for embedded devices (typically wireless routers). Unlike many other distributions for these routers, OpenWrt is built from the ground up to be a full-featured, easily modifiable operating system for your router. In practice, this means that you can have all the features you need with none of the bloat, powered by a Linux kernel that's more recent than most other distributions.

In this Lab, we will use a router called GL-MT300N-V2 which has OpenWrt pre-installed. This router is small, light and easy to use. Some hacker may use this kind of router on the public environment such as a coffee store, shopping mall, airport, etc to mark a free wi-fi network and achieve user's information and password.

3.2. Getting started with OpenWrt

Power up the router and wait for few seconds. Once orange color light up you will have the default WiFi network up and running.

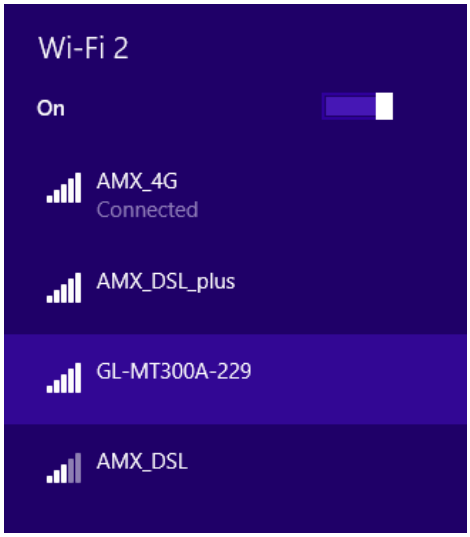

Figure 1: Examples on connecting with router Wi-Fi network

Default WiFi password is written on the back of the mini router. Once you are connected to the network, use the default IP address written on the back of the mini router to access the management console.

After language, country selection, and setting admin password, you will get the simplified ad min console for GL.iNet.

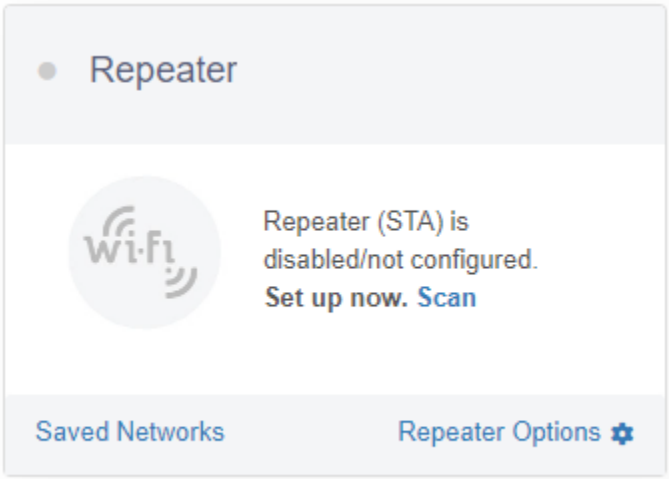

Figure 2: Simplified admin console for GL.iNet

In this section, we will use the repeat network on the network connection. Select internet settings on simplified admin console shown as figure 2. Find “Repeat” on main page, select Scan, then you will find some available Wi-Fi network. If not, you can use your mobile phone to start an access point. Then use your phone as a hotspot and connected with the network. If you are using your mobile phone as a personal hotspot, you may disconnect with any Wi-Fi network use LTE/4G as the network connection.

Notice: You may not successfully connect to the internet as a repeater through Eagle-WiFi since it needs to be accessed through username and password.

When you successfully connected with one Wi-Fi, refresh the website and you will see the ‘internet connection’ bar has changed from blank to an IP address setting on it and if you click on that, you will find the detailed information on repeater connection.

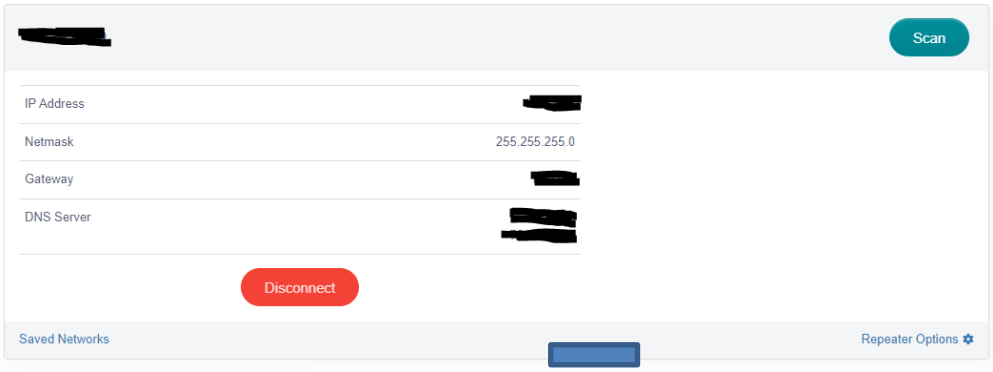

Figure 3: Example on successfully connect to the internet through repeater mode

Notice: To make sure you are successfully connected to the internet, you may ping 8.8.8.8 and check the connection. If you are using mobile devices on repeater connection, be aware of mobile data use.

Sometimes the operating system will automatically switch to an Internet-accessible Wi-Fi network. So make sure your repeatable Wi-Fi has Internet access all the time and you are always connecting on the Wi-Fi by the router.

3.3. Analyzing network traffic with OpenWrt

In this section, we will SSH to OpenWrt installed router and use software such as tcpdump and Wireshark to analyze network traffic.

Step 1: SSH to the router.

In this step, you can use whether windows PuTTY or Linux/Unix based system to SSH to the router i.e. Kali system. If you are using windows system, download PuTTY from the website (Choose MSI ('Windows Installer') -> 64-bit or 32-bit) and install. Open PuTTY, on Host Name(or IP address) part, type in the router IP address and click open. Then, on command window, log in as root and type in the password you set before. You will then successfully SSH to the router. If you are using Kali, make sure to configure the network interface to Bridge.

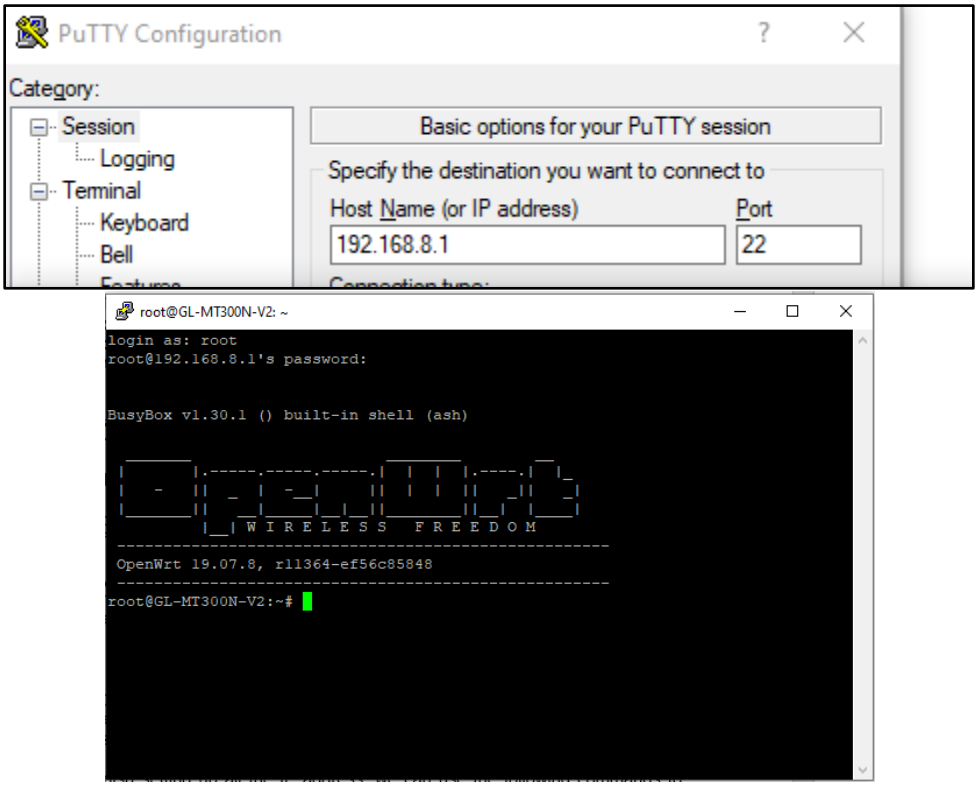

Figure 4: Example on successfully using PuTTY on Windows SSH to the router

Warning: These approaches might specially come in handy when you need to analyze packets generated by mobile devices. It could be a great setup to analyze how an Android device infected with a malware, behaves and communicates with external components.

In addition, a similar setup can also be used to perform a passive man in the middle attack. This is why it is generally advised not to use public WiFi networks unless you are protected with a VPN (or at least trusted SSL/TSL connections).

Step 2: Analyzing network traffic with remote Wireshark listener

In this section, you will use the Kali VM system and Wireshark in Kali to analysis the network traffic in your laptop. Following three steps are setups before analyzing.

First, open your Kali system and make sure the network adapter is in "Bridge" mode.
Second, run "dhclient" on the command line to refresh the network settings and make sure your Kali system has an updated IP address in the same network of your laptop and your router.
Third, mark down all the IP address of these three devices. Following is an example:

- OpenWrt Router: 192.168.8.1
- My laptop device: 192.168.8.226 (Victim, check your ipconfig/ifconfig under Wi-fi)
- Kali system with Wireshark: 192.168.8.181 (attacker, check this as well)

Once you finish setting up all the IP address, we can use the following commands to analysis on Wireshark.

1. SSH into OpenWrt installed router (usually port 22, as we did on step 1) and install "iptables-mod-tee" to update the router with below command:

```
$ opkg update
$ opkg install iptables-mod-tee
```

2. Run following iptables command to "forward a copy of each packet with source-IP (-s) on out interface (-o) to gateway-IP (–gateway)"

```
$ iptables -A POSTROUTING -t mangle -o br-lan ! -s 192.168.8.226
-j TEE --gateway 192.168.8.181
```

3. Run following iptables command to "forward a copy of each packet with destination-IP (-d) on in interface (-i) to gateway-IP (–gateway)"

```
$ iptables -A PREROUTING -t mangle -i br-lan ! -d 192.168.8.226 -
j TEE --gateway 192.168.8.181
```

4. Start capturing traffic on Wireshark over eth0 with below filter applied :

```
(ip.src == 192.168.8.226) || (ip.dst == 192.168.8.226)
```

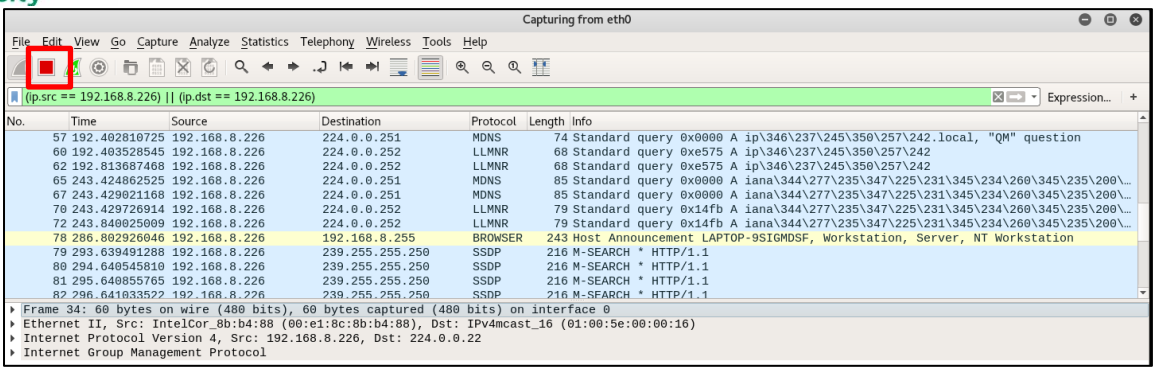

Figure 5: Wireshark capturing from Laptop

From victim send 10 packets to 8.8.8.8, after that stop Wireshark.

Use Wireshark to display the captured traffic.

Step 3: Capturing communication with tcpdump

Tcpdump can be installed on OpenWrt router itself. Therefore, this approach eliminates the need of having a remote Wireshark or similar listener to analyze the traffic in real-time.

SSH into OpenWrt installed the router and install “tcpdump” with below command:

```
opkg update
opkg install tcpdump
```

Execute below command to listen on the interface (-i) and store captured information to a file (-w) and be verbose while doing so (-v).

```
tcpdump -i any -v -w pcap.cap
```

In the exercises, you will need to retrieve and open the `pcap.cap` file with Wireshark for further analysis.

From victim send 10 packets to 8.8.4.4

Use `tcpdump -r pcap.cap` command to display the content of the file

3.4. Extroot configuration

Notice: In this section, you will need an empty USB storage device since all the data inside the device will be erased.

Many useful LEDE utilities and packages rely on external storage to hold data files. This section is used on how to use a storage device (USB or sata or sdcard or whatever) to expand your LEDE device's space in root filesystem, to install freely all the packages you need.

Background Info:

In most supported devices, the LEDE firmware splits the internal storage into two partitions

- “root filesystem” (/), a highly-compressed read-only partition
- “overlay” (/overlay), a second partition that is writable

The overlay partition is merged with the root filesystem using the overlayfs (a feature of linux kernel), showing a single “whole” read-write filesystem to applications. This way LEDE fits even in tiny amounts of internal storage (as low as 4 MiB) but still allows to write settings and install some packages in the writable partition without changing all Linux programs used.

Extroot, also known as rootfs on external storage, works by setting another overlay partition in the external storage device, and during boot, this new overlay partition will be mounted over the internal storage's overlay partition. This approach allows easy fallback in case the external storage device is removed, as your LEDE device will still have its own overlay partition and thus will load all configuration from there. Which means that it will behave exactly the same as just before you set up extroot.

Step 1: From the command line interface write (on a single line):

```
$ opkg update && opkg install block-mount kmod-fs-ext4 kmod-usb-
storage e2fsprogs kmod-usb-ohci kmod-usb-uhci fdisk
```

This installs packages needed for a partition with the ext4 filesystem (and doesn't install packages for the f2fs filesystem).

Step 2: Run command “fdisk -l” to check which part is the major storage section on your USB disk. Your USB device may only have one partition, so make sure to use your own sda to mount. For example, **sda2** is the major storage section in this USB disk:

```
Disk /dev/sda: 7.5 GiB, 8086618112 bytes, 15794176 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x426b5e8b

Device     Boot  Start      End  Sectors  Size Id Type
/dev/sda1           1   409639   409639  200M  e W95 FAT16 (LBA)
/dev/sda2      411648 15792127 15380480  7.3G  c W95 FAT32 (LBA)
```

Figure 6: USB device size check information.

Step 3: We now first format the external drive as f2fs or ext4.

For f2fs:

```
$ mkfs.f2fs /dev/sda2
```

For ext4:

```
$ mkfs.ext4 /dev/sda2
```

Then we transfer the content of the current overlay inside the external drive

```
$ mount /dev/sda2 /mnt ; tar -C /overlay -cvf - . | tar -C /mnt -
xf - ; umount /mnt
```

Step 4: Now we create automatically the fstab uci subsystem and fill it with the right configuration to have /dev/sda2 as new overlay

```
$ block detect > /etc/config/fstab; sed -i
s/option$'\t'enabled$'\t'\'0\'/option$'\t'enabled$'\t'\'1\'/
/etc/config/fstab; sed -i s#/mnt/sda2#/overlay#
/etc/config/fstab; cat /etc/config/fstab;
```

Now we are successfully generated fstab on /dev/sda2.

Step 5: Now as you refresh the website of the router, you can see a device appears on the main page.

```
root@GL-MT300N-V2:~# block detect > /etc/config/fstab; sed -i s/option$'\t'enabled$'
tc/config/fstab; cat /etc/config/fstab;
config 'global'
        option  anon_swap       '0'
        option  anon_mount      '0'
        option  auto_swap       '1'
        option  auto_mount      '1'
        option  delay_root      '5'
        option  check_fs        '0'

config 'mount'
        option  target  '/overlay'
        option  uuid    '4dccb4c5-41e3-44fa-a9fc-b67albb35b2f'
        option  enabled '1'
```

Figure 7: USB device successfully mounted.

If you did not find the device, you may reboot the router. If you have any questions on extroot configuration, you may first check this website [7]

4. What to turn in for Grading?

(Report with answers to below questions should be turned on the due date; do not forget to write your name and title on the reports)

1. Provide a screenshot (like Figure 5) of your running results for Section 3.3. Highlight the IP addresses of the Router, victim and attacker. (30 points)

2. Based on packets captured in Section 3.3, use the command tcpdump to display the ten packets sent to **8.8.4.4** on your screen. Include a screenshot on input the .cap file in Wireshark (use scp to achieve that: https://www.freecodecamp.org/news/scp-linux-command-example-how-to-ssh-file-transfer-from-remote-to-local/) and the tcpdump query you used. (30 points)

3. Dnsmasq is a lightweight DNS, TFTP, PXE, router advertisement and DHCP server. It is intended to provide coupled DNS and DHCP service to a LAN. Dnsmasq accepts DNS queries and either answers them from a small, local, cache or forwards them to a real, recursive, DNS server. It loads the contents of /etc/hosts so that local hostnames which do not appear in the global DNS can be resolved and also answers DNS queries for DHCP configured hosts. It can also act as the authoritative DNS server for one or more domains, allowing local names to appear in the global DNS. Read the article: “DNS spoofing with Dnsmasq (https://openwrt.org/docs/guide-user/base-system/dhcp.dnsmasq)” and try to run Dnsmasq on your router. Provide a screenshot on successfully assign Dnsmasq on your router. (40 points)

4. (Extra Points) Open-ended question: In this lab, we introduce two basic packages which helps you start learning OpenWrt system. Search online and the official website, provide which you think are interesting such as security tools, guest hotspot and so on. Explain what these packages are mainly do and install at least one package which interests you the most in your router. Provide some screenshots. (10 points each package explore, 30 extra points at most to be given)