**Preparation(sự chuẩn bị)**

**Installing the Required Software**

Each year millions of home routers được bán cho người tiêu dùng(consumers), a large majority(số đông) of them don’t even(thậm chí) know whats running on them. Today we’re going to take a look(xem xét). Before proceeding(tiến hành) we will need a few tools:

* Access to a Linux distribution(1 bản phân phối Linux) (Or WSL) with strings and binwalk on it.
* Linksys WRT1900ACS v2 Firmware found here: <https://github.com/Sq00ky/Dumping-Router-Firmware-Image/>
* Lastly, [ensure binwalk has JFFS2 support with the following command](https://github.com/ReFirmLabs/binwalk/blob/master/INSTALL.md):

sudo pip install cstruct

git clone <https://github.com/sviehb/jefferson>

cd jefferson && sudo python setup.py install

After you've got the tools, you're ready to setup your work space!

**Rebuilding the Firmware**

First, we're going to clone the repository that holds(chứa) the firmware:

git clone https://github.com/Sq00ky/Dumping-Router-Firmware-Image/ /opt/Dumping-Router-Firmware && cd /opt/Dumping-Router-Firmware/

Next, we're going to unzip the **multipart zip file**:

7z x ./FW\_WRT1900ACSV2\_2.0.3.201002\_prod.zip

running an lsyou should see the firmware image:

FW\_WRT1900ACSV2\_2.0.3.201002\_prod.img

Lastly, running a sha256sum on the firmware image you should be left(bạn nên để lại) with the value: **dbbc9e8673149e79b7fd39482ea95db78bdb585c3fa3613e4f84ca0abcea68a4**

Investigating Firmware (điều tra....)  
\*(firmware = software for hardware)

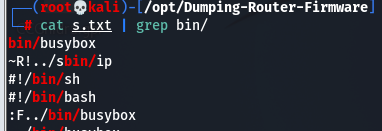
**One Inch Deep Analysis(phân tích sâu one inch)**

Trong phần này, chúng tôi sẽ taking a look at the firmware, checking for strings và dump the file system form the image. Phần tiếp theo sẽ bao gồm lắp đặt và exploring the file system.

**Answer the questions below**

Trong khi running a strings on the file, there is a lot of notable(đáng chú ý) clear text. Điều này là do để chắc chắn các khía cạnh của hình ảnh firmware không được mã hóa. Điều này có nghĩa như với Binwalk, chúng ta sẽ có thể dump the firmware from the image.

* Running strings on the file, what does the first clear text line say? “**Linksys WRT1900ACS Router**”
* Also using strings, what operating system is the device running? ”**Linux**”

Scrolling(cuộn) through with strings, you may notice(lưu ý) some other interesting lines like **/bin/busybox** and various(nhiều) other lua files. It really makes you wonder(ngạc nhiên) whats going on inside there...  


Next we will be dumping the file system from the image file. To do so, we will be using a tool called binwalk.

Binwalk is a tool that checks for wellknown file signatures within(ở trong) a given(đã được cho) file. This can be useful for many things, it **even**(thậm chí) has its uses in Steganography. A file could be hidden within the photo and Binwalk would reveal(tiết lộ) that, and help us extract it. We will be using it to extract the file system of the router in this instance(ví dụ)

* What option within Binwalk will allow us to extract files from the firmware image? **‘-e’**
* Now that we know how to extract the contents of the firmware image, what was the first item(mục) extracted? “**uImage header**”
* What was the creation date? “**2020-04-22 11:07:26**”

The **Cyclical Redundancy Check** is used in a similar way that file hasing is, to ensure(đảm bảo) that the file contents were not corrupted(bị hỏng) and or modified(sửa đổi) in transit(vận chuyển).

* What is the CRC of the **image**? “**0xABEBC439**”
* What is the image size? “**4229755 bytes**”
* What architecture does the divice run(Thiết bị nào đang chạy architecture)? “**ARM**”
* Researching the results to(kết quả) question 10, is that true? “**Yes**”  
  **Hint: “Yes, it is true. Having the architecture can be very important to reverse engineering.”**

You will notice two file got extracted; one being(một là) the jffs2 file system and another that Binwalk believes it to be gzip compressed(được nén) data

You can attempt(nỗ lực) to extract the data, however, you won’t get anywhere. Binwalk misinterpreted(giải thích sai) the data, however, we can still do some analysis of it.

* Running strings on 6870, we notice a large chunk(khối) of clear text. We can actually(thực ra) run binwalk again on this file to receive even more files to investigate(để nhận được nhiều tệp hơn nữa để điều tra). Interestingly enough, a copy(một bản sao) of the Linux kernel is included. What version is it for? “**3.10.39**”

If you run extract the contents of 6870 with Binwalk and run strings on 799E38.cpio, you may see a lot of hex **towards the bottom of the file**(phía dưới cùng của tệp). Some of it can be translated into **human readable text**(văn bản có thể đọc của con người). Some of it is interesting to see and really makes you wonder(tự hỏi) what its purpose was for(mục đích của nó là gì). Có lẽ một số cuộc điều tra bổ sung sẽ tiết lộ mục đích của nó. Mặc dù vậy tôi sẽ để bạn tự mình khám phá điều đó :)

Tiếp tục với the analysis, chúng tôi có một jffs2 file system, cái mà chúng tôi có thể kiểm tra nội dung. Đầu tiên, chúng ta cần phải gắn kết nó mặc dù, điều này đưa chúng ta vào phần tiếp theo

Mounting and Analysis of the Router’s Filesystem

**Mounting the Filesystem**

In this section, we will begin to go over how to mount the file system. Note, if you are doing this with any other file system and it is not in the **Little Endian** format, you will need to convert it from **Big Endian** to **little Endian** using a tool called jffs2dump. But here là hướng dẫn khá ngắn gọn của tôi to mounting the filesystem:

Step 1. If /dev/mtdblock0 exists, remove file/directory and re-create the block device

rm -rf /dev/mtdblock0

mknod /dev/mtdblock0 b 31 0

Step 2. Create a location for the jffs2 filesysystem to live  
mkdir /mnt/jffs2\_file/

Step 3. Load required kernel modules

modprobe jffs2

modprobe mtdram

modprobe mtdblock

Step 4. Write image to /dev/mtdblock0

dd if=/opt/Dumping-Router-Firmware /\_FW\_WRT1900ACSV2\_2.0.3.201002\_prod.img.extracted/600000.jffs2 of=/dev/mtdblock0

Step 5. Mount file system to folder location

mount -t jffs2 /dev/mtdblock0 /mnt/jffs2\_file/

Step 6. Lastly, move into the mounted filesystem.

cd /mnt/jffs2\_file/

Để giải thích một chút về những gì command làm, we're creating a block device (mtdblock (<https://en.wikipedia.org/wiki/Memory_Technology_Device> )) that will allow us to dump the flash memory. We're first removing it if it exists(tồn tại), and then re-creating it.

Next, we're creating a location for our [jffs2](https://en.wikipedia.org/wiki/JFFS2) file (<https://en.wikipedia.org/wiki/JFFS2> ) to be mounted to.

After that we're loading some kernel modules that will allow us to interact(tương tác) with the jffs2 file system and dump the flash memory.

Next, we are writing the file system to(vào) the block device, and after that we are mounting the mtdblock device, which now contains the flash memory of the file system.

Lastly, executing  cd /mnt/jffs2\_file/ we are now sitting inside the routers dumped firmware and can begin điều tra.

**Answer the questions below**

Run ls -la tiết lộ rất nhiều thông tin thú vị. Đầu tiên chúng tôi nhận thấy rằng rất nhiều tệp được liên kết tượng trưng (tương tự như shortcut).

* Where does linuxrc link to? “**bin/busybox”**
* What parent folder does mnt, opt, and var link to? **“/tmp/”**
* What folder would store the routers HTTP server? **“/www/”**

Scannng through rất nhiều thư mục này, bạn có thể bắt đầu nhận thấy rằng chúng trống rỗng(empty). Điều này cực kỳ kỳ lạ, nhưng đó là bởi vì bộ router không hoạt động. Hãy nhớ rằng, chúng ta đang xem xét một mẫu của filesystem sẽ được nhấp nháy trên router, không phải firmware từ một router đã bị dump. Các thông tin khác về router có thể được chứa trong previous section trong 6870 block.

The first of the folders that isn't empty is /bin/, Đâu là phần lớn các tệp tin liên kết đến?

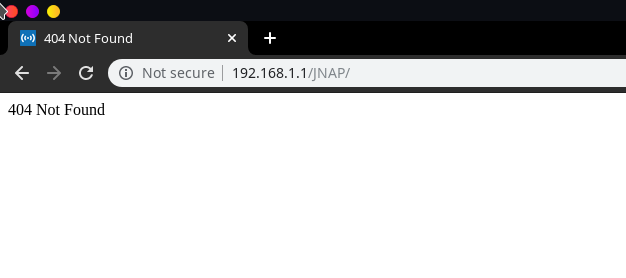
* Why is that? Well, [busybox is more or less a tool suite(bộ công cụ) of common(phổ biến) executable(thực thi) commands within the Unix environment](https://ubuntuforums.org/archive/index.php/t-846852.html). **“busybox”**
* Within the bin folder, interestingly enough, what database(cơ sở dữ liệu) would be running if the router was online? **“sqlite3”**

Thư mục quan tâm đáng chú ý tiếp theo là /etc/. This folder contains a lot of configuration(cấu hình) files for the router, such as **Access Point power levels** được quy định bởi một số quốc gia nhất định. One you might recognize is the FCC (Federal Communications Commission).Bạn có thể nhận ra là the FCC (Federal Communications Commission - ủy ban truyền thông liên bang).

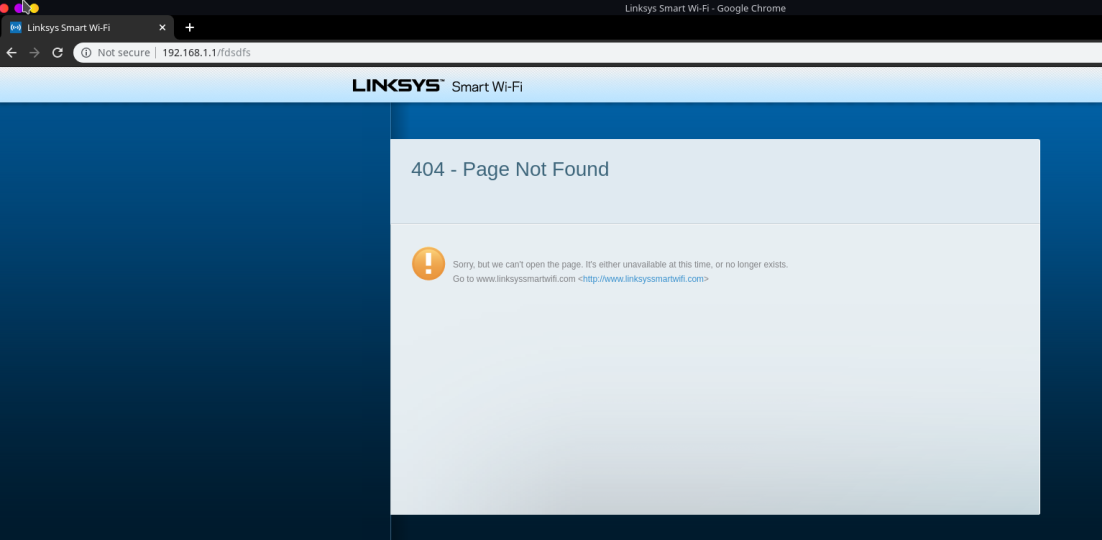
* We can even(thậm chí) see the build date of the device. What is the build date?   
  catr**“2020-04-22 11:44”**
* There are even files related(liên quan) to the SSH server on the device. What SSH server does the device run?   
  **“dropbear”**
* We can even see the file for the media server, which company developed it? **“Cisco”**
* This company use to own Linksys at one point in time, đó có thể là khả năng tại sao nó vẫn đang được sử dụng.
* Which file within /etc/ contains a list of common Network services and số cổng liên quan của chúng?
* Which file contains the default system settings? **“system\_defaults”**
* Within the /etc/ folder, what is the version specific firmware version? **“2.0.3.201002”**

Sao lưu vào the JNAP folder, the JNAP API (trước đây được biết đến như HNAP, the Home Network Administration Protocol) has been a **potential attack vector(vecto tấn công tiềm năng?)** and lỗ hổng trong quá khứ, which this article(bài báo) highlights [here](https://routersecurity.org/hnap.php). Interestingly enough, **reminense** of it is still here today on Linksys devices. Going to  http://<Default\_Gateway>/JNAP/ on a Linksys router reveals(tiết lộ) an interesting 404. Much different than the standard(tiêu chuẩn) 404.

**Accessing /JNAP/**



**Accessing any other invalid URI**



Điều này làm cho bạn tự hỏi nếu một cái gì đó vẫn còn thực sự ở đó. Nếu bạn điều tra trong /JNAP/modules folder trở lại the dumped filesystem, bạn sẽ thấy một số nội dung liên quan đến device và những services mà nó cung cấp, some of them are firewalls, http proxies, QoS, VPN servers, uPnP, SMB, MAC filtering, FTP, etc.

Side note: If you have a Linksys router and quan tâm đến việc khám phá, I found this [Github Repository](https://github.com/jakekara/jnap) for tools to interact with JNAP, Tôi đã chọn không bao gồm điều này trong phòng vì không phải ai cũng có quyền truy cập vào Linksys router. Tôi sẽ không đi xa hơn là khám phá the File System.

* What 3 networks have a folder within /JNAP/modules?

Sau thư mục JNAP, lib là thư mục duy nhất khác, nó có bất kì nội dung nào và những gì trong đó khá tiêu chuẩn về libraries. Phần còn lại của Filesystem là tương đối thoáng dẫn chúng ta đến cuối căn phòng này.

Tôi hy vọng tôi đã làm cho tất cả các bạn tò mò hơn một chút about whats going on in your device, most importantly, I hope you enjoyed. Tôi khuyến khích all of you to go out on your own and get your own router's Firmware and do some firmware dumping and take a look at whats going on inside your device.

A room about Cable Modems may come in the future, however, Cable Modems firmware images are tương đối difficult to get your hands on vì chúng chỉ được phân phối cho các CMO (Cable Modem Operators, like Charter, XFinity, Cox, etc.)