How to block and analyze LAN traffic with Packet Squirrel and Wireshark

If you spot an unsupervised Ethernet connection and wonder what you can do with all the information passing through that connection, there's an easy way to learn this.

Suppose there is a router for which you know the password and have physical access to an Ethernet connection where a Packet Squirrel can be attached - a pocket-sized tool inserted in the middle of the network. If the router is not using HTTPS, you can log all traffic over Ethernet. Once someone accesses the router, you'll have access to information to log in and do whatever you want.

Packet Squirrel and Wireshark record all traffic over Ethernet

Request

You will need a Packet Squirrel, created by Hak5 from its website, Amazon, or possibly another online store.

The device doesn't come with a power adapter, so you'll also need a Micro-USB cable and power supply. We'll plug the device into your computer's USB port, but if you want to deploy it in the real world, you'll need a power adapter or a small battery instead.

Packet Squirrel only requires 5V, 150mA of power, so you can use something as simple as a smartphone charger. If used for a short time, a small battery pack can be used (this battery pack usually comes with its own small USB cable). For extended sessions, you'll need a larger battery pack with a Micro-USB cable.

To record monitoring traffic, you'll also need a USB flash drive. You can choose any USB you like. The USB will need to be formatted in NTFS (this will be resolved later).

Ultimately, you need access to the Ethernet connection you want to monitor, as well as Wireshark to unravel the mysteries of the data you will collect.

Step 1: Select payload

Packet Squirrel comes with three types of payloads: TCPdump packet blocking, man-in-the-middle attack that spoofs DNS, and OpenVPN tunnels in or out. Since I don't want to do anything too difficult like setting up a server or creating complex configurations, the example will use a simple TCPdump.

Step 2: Get started with Packet Squirrel

To get started, you'll need to set Packet Squirrel to Arming Mode, which can be done using the same switch on the device, allowing you to choose between payloads. To do so, push the switch fully towards the USB port. Next, plug the Packet Squirrel into the power source.

Next, plug the Ethernet cable from the computer into the Ethernet In port on the Packet Squirrel, right next to the power port. If you want to access the Internet while checking everything, you can connect it via another Ethernet cable, with the Ethernet Out port next to the USB port. Nothing bad will happen. You will simply have access to the device via SSH.

Step 3: Prepare for Packet Squirrel

Now let's see if you can access the device without a problem. In a terminal window, use SSH to access the device with its root IP address. The IP address is **172.16.32.1** for all Packet Squirrel. Then, enter hak5quirrel's default password to login.

```
~$ ssh root@172.16.32.1 root@172.16.32.1's password: hak5quirrel BusyBox v1.23.2
```

Now take a quick list (**ls**) to see what you have.

```
root@squirrel:~# ls VERSION payloads
```

Payload is what you want, so change into that directory, then do another quick listing to see what's there.

```
root@squirrel:~# cd payloads root@squirrel:~/payloads# ls switch1 switch2 switch3
```

There should be 3 switches there, one for each payload for which the Packet Squirrel is loaded. If you want, you can customize one or two of these switches with your own payload, and for example switch to switch1, for the TCPdump payload, the tool you want to use. Then do another quick listing to see what's inside.

```
root@squirrel:~/payloads# cd switch1 root@squirrel:~/payloads/switch1# ls payload
```

Let's see how payload works now. Use **cat** to open the file.

```
root@squirrel:~/payloads/switch1# cat payload.sh #!/bin/bash # TCPDump payload v1
```

The entire script is visible above. Basically what it does is route the traffic, and also dump all the traffic it routes, into a TCPdump file. As you can see near the bottom of the script, the payload only works if there is a USB storage device attached to the Packet Squirrel.

Step 4: Format the USB

Now format the USB. In a terminal in Kali Linux or macOS use the following command to locate the USB drive, to make sure the correct device is selected.

```
~$ df -h Filesystem Size Used Avail Use% Mounted on udev 2.8G 0 2.8G 0% /dev tmp
```

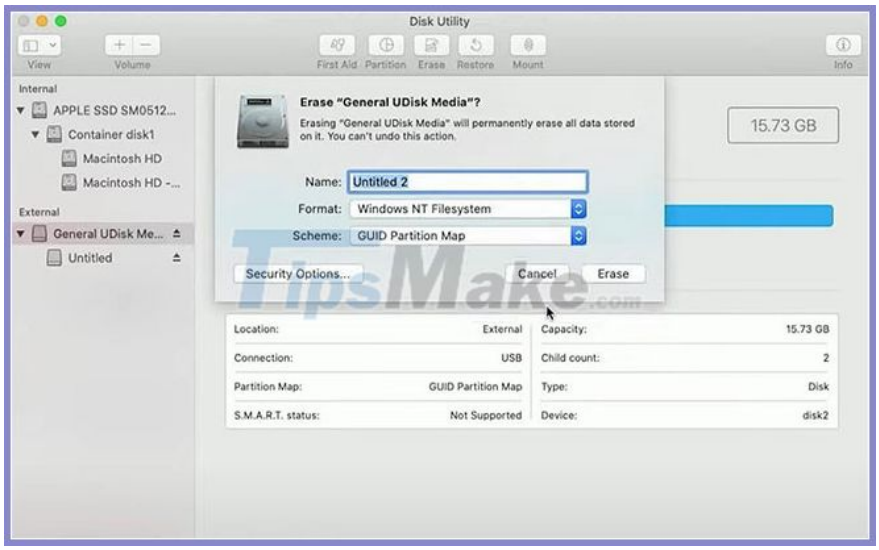
In our example, the 128GB USB drive of the article author is located in **/ dev / sdc1** , which is what is being used. Since this USB flash drive is already mounted, it is necessary to disconnect it before formatting using Windows NT Filesystem or NTFS.

```
~$ sudo umount /dev/sdc1
```

Now, to format it using NTFS, use:

```
~$ sudo mkfs.ntfs /dev/sdc1
```

On a Mac, you can also use Disk Utility to format the drive. Just make sure to use Windows NT Filesystem when you delete it.



When done, take the USB out, then plug it into the Packet Squirrel.

Step 5: Select the appropriate payload switch

Now, on Packet Squirrel, you need to switch from Arming Mode to TCPdump payload. So push the switch towards the power port. Next, restart the Packet Squirrel by unplugging it from the power source, then plugging it in again.

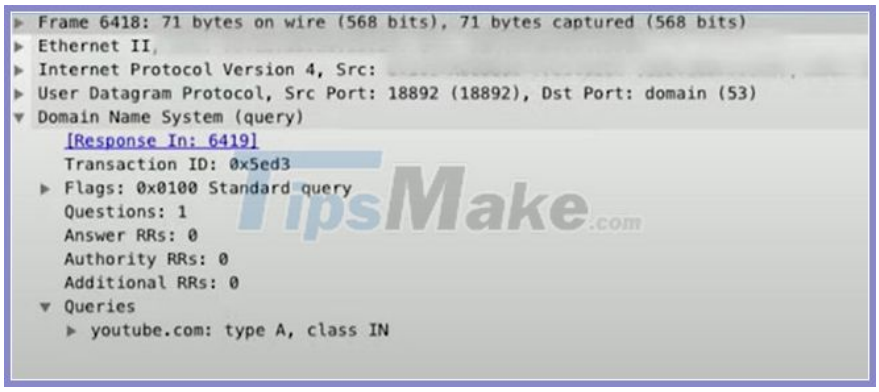
Step 6: Start crawling!

Now, Packet Squirrel is ready for you to plug into the network you want to monitor and start capturing the Internet activity of anyone using Ethernet. You can see all kinds of interesting content, depending on the type of website the person is visiting and the services they are running. After capturing enough information using Packet Squirrel, you can take the USB out and plug it into your computer to check what's found.

Step 7: Analyze the data with Wireshark

Navigate to the **/ loot / tcpdump / directory** to find all the .pcap files that Packet Squirrel has collected. These files can be viewed with Wireshark, so please open one of them. A lot of unencrypted data will be recorded. The unencrypted data are mainly HTTP and DNS requests. By looking at that unencrypted information, you can find the desired target.

In the Wireshark search bar, type **dns** to see all DNS requests sent. Click on any of them, you can open **Domain Name System (response)> Queries** and see if it's an automated or user-sent query. Next, go through the rest of the section, looking at the **Queries** section of each section, to see where they come from.



See if the query is automatic or user-sent
In the Wireshark search bar, type **http** to view all sent HTTP requests. Left click on any recorded item, choose **Follow> HTTP Stream** . That will open up a new window, monitoring what happened, can display both sides of the conversation between the computer and the server (send and receive). You can see things like information exchanged, user agent (software acting on the user's behalf), referrer (information sent by the user's browser as they move from page to page). , etc.

You can also search for something like **http contains "vistamiddle"** to see all HTTP requests interacting with **vistamiddle.org.** Double-click a request to view its details. Here's a way to find out the router's password or something more personal.

Some limitations

While the Hak5 Packet Squirrel is a great tool for taking advantage of any unattended Ethernet connection, there are a few limitations that you need to keep in mind.

Firstly, you will need to go back to the USB host location if you want to get data. So if you only have one-time access, then this might not be the best tool to use.

Second, this tool will not be able to capture any traffic sent by VPN or over SSL, so there will be some things you won't be able to see. However, the HTTP request and DNS request are both ready for you to fetch.