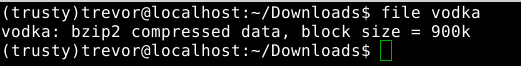
**Vodka Writeup**

**Introduction:** Vodka was a medium level 400 point forensics challenge given during the 2014 NCN CTF. This writeup was made possible using resources from the [CTF Wiki](https://github.com/ctfs/write-ups-2014/tree/master/ncn-ctf-2014/Vodka) and an external write-up from [BalalaikaCr3w](https://ctfcrew.org/writeup/90). After reading this write up you should have a decent idea of the steps required to find a flag hidden within a firmware file.

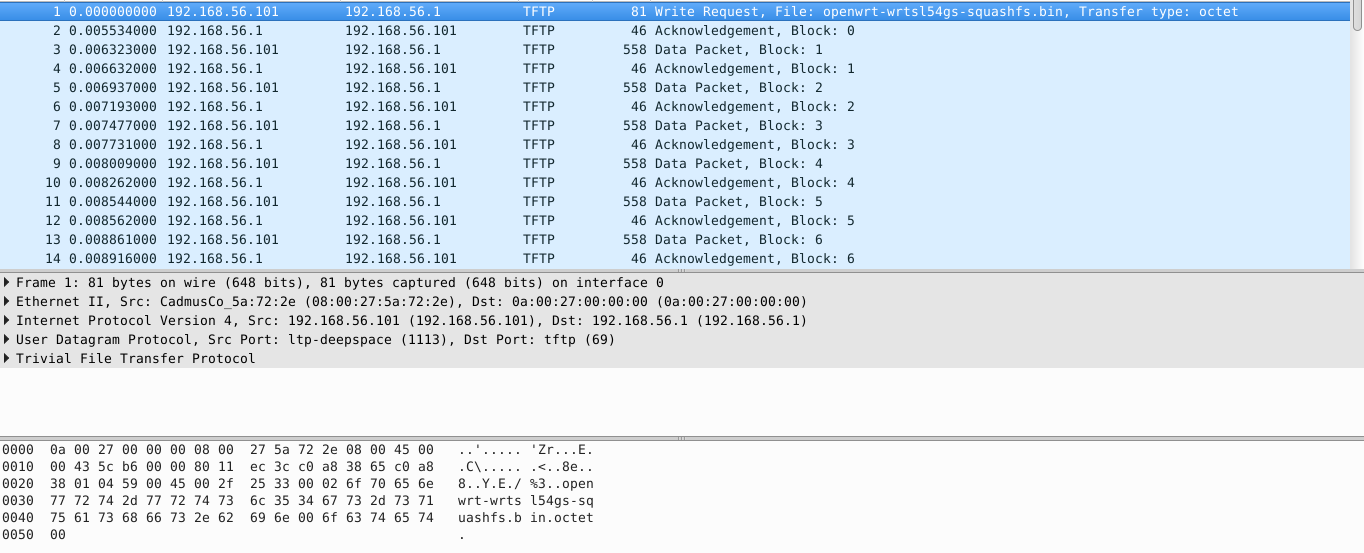
**Task:** No hints :( just get the flag.

**Tools:** This challenge requires a few tools to solve. You’ll need [Wireshark](http://wireshark.org), [Tshark](https://www.wireshark.org/docs/man-pages/tshark.html), and a copy of [Firmware Mod Kit](https://code.google.com/archive/p/firmware-mod-kit/). Wireshark and Tshark can both be installed via the command line by typing “sudo apt-get install wireshark” . “sudo apt-get install tshark”. Firmware Mod Kit however requires a few dependencies to build correctly, you can install the required dependencies with “sudo apt-get install git build-essential zlib1g-dev liblzma-dev python-magic”. Once you’ve installed the required dependencies for Firmware Mod Kit you’ll need to extract the .zip file anywhere you choose. Once the Firmware Mod Kit directory has been extracted you need to build the tool. Open a terminal in the ‘src’ folder and type “./configure”, then “make” to build Firmware Mod Kit.

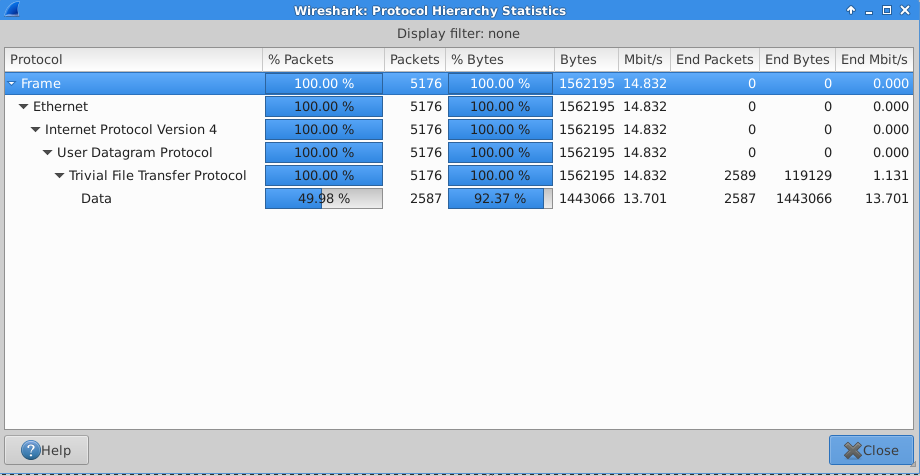
**Solving:** Download the provided file and let’s get to work. Start by running ‘file’ to see what we’re working with. Go ahead and extract the bzip2 archive using ‘bunzip2’ and you should get a .pcap file out.



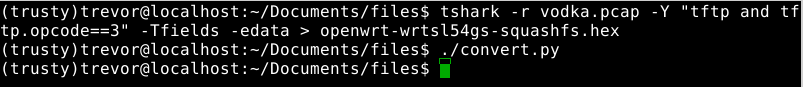
Let’s analyze the given pcap file using Wireshark. The very first packet in the capture is a TFTP request for a file named ‘openwrt-wrtsl54gs-squashfs.bin’ it would seem someone was trying to download router firmware.



We can continue our analysis by examining the protocol hierarchy of this capture.

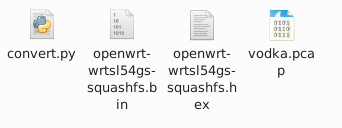


100% of the captured packets are TFTP protocol but only 50% of the packets contain data. Using Tshark we should be able to extract the data that was downloaded using the command ‘tshark -r vodka.pcap -Y “tftp and tftp.opcode==3” -Tfields -edata > openwrt.hex’. Once we’ve extracted the data using Tshark it must be converted from hex to binary using the provided python script.

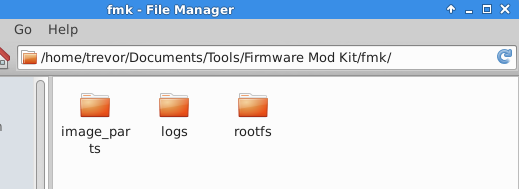


The provided script is incredibly simple. The script simply opens the .hex file specified, reads the number of lines in the file. Truncates the ‘\n’ character from each line then converts the hex into binary. Note: You may have to chmod the script to run it ‘chmod 755 scriptname’ should do the trick.

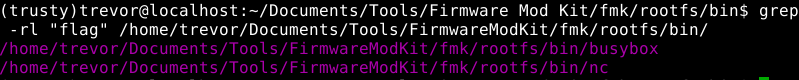
All said and done you should have 4 files like so:



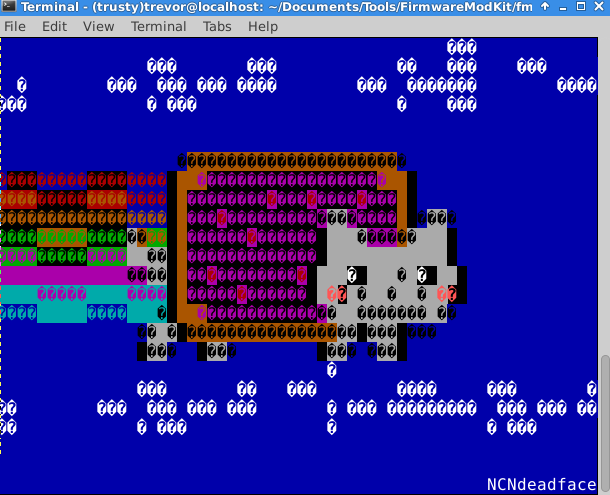
Now that we’ve converted the file from hex to binary let’s try and extract the firmware using the Firmware Mod Kit. Firmware Mod Kit is an incredibly powerful tool allowing users to make modifications to vendor firmware files that ship with routers. For this write up we’ll simply be using the ‘extract-firmware.sh’ script. To make your life easier I would recommend copying the converted binary file into the root of the Firmware Mod Kit directory. Once you’ve done so run the command ‘./extract-firmware.sh openwrt-wrtsl54gs-squashfs.bin’ to extract the firmware from the image. Unless otherwise specified Firmware Mod Kit will dump the firmware contents into another folder called ‘fmk’ inside of the Firmware Mod Kit folder.



Navigate inside the ‘rootfs’ folder and and you should notice a familiar file system here, time to begin looking for our flag using grep. Using Grep search the file system folder by folder for the keyword ‘flag’ using the following command ‘grep -rl “flag” pathToFile’.



Running our search returns two files containing the keyword “flag”. I’m going to go out on a limb and say that our flag isn’t contained in the busybox binary, let’s take a look at the nc binary. Remember to chmod the binary so you can execute it. After executing the binary we’re given:



Our flag is NCNdeadface