This is a Word document that allows users to type into the spaces below. The comment may be single-spaced, but should be in at least 12-point type. The italicized instructions on this template may be deleted.

UNITED STATES COPYRIGHT OFFICE

Long Comment Regarding a Proposed Exemption Under 17 U.S.C. § 1201

[ ] Check here if multimedia evidence is being provided in connection with this comment

ITEM A. COMMENTER INFORMATION

The Petition submitter is Software Freedom Conservancy (Conservancy), a not-for-profit organization that helps to promote, improve, develop, and defend Free and Open Source Software (FOSS)—software developed by volunteer communities and licensed for the benefit of everyone. Conservancy is the nonprofit home for dozens of FOSS projects representing over 5,000 volunteer contributors. Our communities maintain some of the most fundamental utilities in computing today, and introduce innovations that will shape how software will be created in the future.

Among the projects for which Conservancy provides logistical, administrative, and legal support are OpenWrt and BusyBox. OpenWrt produces an embedded operating system for routers that can be installed in place of the stock firmware on commercially available routers. BusyBox provides a number of key system utilities that enable such devices to run applications, interact with files, access network services, and more. Conservancy also represents the interests of a coalition of contributors to the Linux kernel. Both BusyBox and Linux are core components of the operating system of OpenWrt and most consumer routers.

Conservancy may be contacted as follows:

Karen Sandler, Executive Director
Software Freedom Conservancy, Inc.
137 Montague St., Ste.
380 Brooklyn, NY 11201-3548
dmca-exemption@sfconservancy.org
+1-212-461-3245

ITEM B. PROPOSED CLASS ADDRESSED

Proposed Class 11: Computer Programs—Jailbreaking

To enable the installation of alternative firmware in routers and other networking devices.

ITEM C. OVERVIEW

The purpose of the proposed exemption is to enable owners of wireless routers and other networking devices to improve the reliability, functionality, and security of their devices by installing alternative operating system software. Wireless routers can be found in nearly every home or business with an internet connection and they provide a critical link between end-user computing devices and the internet at large. Every wireless router is a general purpose computer with an embedded operating system, and on most routers, that operating system is built primarily from FOSS components, including the Linux operating system kernel, BusyBox, and other utilities.[1]

As with other computers, the software on wireless routers—referred to collectively as "firmware" because it is installed in the router's semi-permanent memory—can be upgraded, extended, and replaced. Router manufacturers sometimes release new versions of their own firmware to fix bugs and security issues, or to add new functionality, such as support for a new wireless standard. However, manufacturer-supplied firmware is a "take it or leave it" proposition, leaving consumers with limited options for deciding what tools to install or how the router is configured. When manufacturers stop providing updates, their routers become increasingly vulnerable to attack by malicious parties, and may become functionally obsolete prematurely as networking protocols advance and the devices become incompatible with current standards.

For over a decade, OpenWrt and similar projects[2] have made it possible for consumers to replace the default firmware on their routers[3] with a FOSS operating system that can improve the

[1] FOSS software licensed under the GNU General Public License, which requires that recipients of licensed software be provided a copy of the software source code, is so prevalent in consumer networking devices that many manufacturers host websites to provide consumers access to the GPL-licensed code on their devices. *See, e.g.*, TP-Link, *GPL Code Center*, https://www.tp-link.com/us/support/gpl-code/; D-Link, *GPL Source Code Support*, https://tsd.dlink.com.tw/downloads2008list.asp?SourceType=download&OS=GPL; Netgear, *Netgear Open Source Code for Programmers (GPL)*, https://kb.netgear.com/2649/NETGEAR-Open-Source-Code-for-Programmers-GPL.

[2] While our comment focuses on OpenWrt, a member project of Conservancy, similar arguments would apply to a number of other FOSS projects that produce operating systems for networking devices, including DD-WRT (https://dd-wrt.com/) and Tomato (https://advancedtomato.com/).

[3] Our comments focus primarily on routers, because they are by far the most common platform for the installation of alternative operating systems such as OpenWrt, given their ubiquity. However, the proposed exemption extends to other networking devices because these operating systems can also be used in switches, network-attached storage devices, WiFi range extenders, modems, WiFi cameras, and other devices. For a full list of devices supported by OpenWrt, *see*

router's performance, reliability, and security, expand its capabilities, and extend its useful life until long after the manufacturer stops supporting it. The OpenWrt operating system is composed of software components licensed under FOSS terms that permit users to copy, modify, and redistribute their code. OpenWrt completely replaces a router's stock firmware, and does not require any of the manufacturer's original software to operate.

Many routers employ technological protection measures (TPMs) to prevent the installation of firmware files except those produced by the router manufacturer. Techniques vary, but common examples include encryption and cryptographic signing of firmware files, and access control measures that limit user access to certain functionality. To install an alternative firmware such as OpenWrt, a user must often circumvent these measures. The proposed exemption would permit the owner of a device to circumvent TPMs that control the installation of firmware on a router or other commercially available networking device for the purpose of installing licensed software.

ITEM D. TECHNOLOGICAL PROTECTION MEASURE(S) AND METHOD(S) OF CIRCUMVENTION

A variety of technological protection measures are used to control the installation of firmware on networking devices such as routers. The following examples of TPMs used on specific devices illustrate common TPMs, as well as a range of techniques used to bypass them:

- The TP-Link TD-W9980 router uses a proprietary encryption scheme to prevent the installation of unauthorized firmware. To circumvent this, a user must login to the router via a telnet connection and replace a configuration file with a new one that is encrypted according to the encryption scheme recognized by the router's firmware. The new configuration enables "root" or "shell" access to the router, and the user can then install an OpenWrt firmware onto the router.[4]

- The TP-Link TD-W8970 router uses a proprietary encryption scheme to prevent the installation of unauthorized firmware. To circumvent this, a user must retrieve a configuration file via the router's web-based user interface and decrypt it. A line in the configuration file is then modified to include a command that, via a security vulnerability in the router, will be inserted into the router's startup routine, enabling root access to the router. The user can then login as the root user and install OpenWrt from the command line.[5]

- A wide range of D-Link routers use encryption and cryptographic signatures to prevent the installation of unauthorized firmware. To circumvent this, a user must reboot the

OpenWrt, OpenWrt supported hardware database, *available at* https://openwrt.org/_media/toh_dump_tab_separated.zip.

[4] OpenWrt, *TP-Link TD-W9980 / TD-W9980B,* https://openwrt.org/toh/tp-link/td-w9980.

[5] OpenWrt, *TP-Link TD-W8980 v1,* https://openwrt.org/toh/tp-link/td-w8970_v1.

router into a special emergency recovery mode, which will permit the installation of unencrypted firmware files.[6]

- The AXIMCom MR-102N mobile 3G/4G router uses encryption to prevent the installation of unauthorized firmware. The router's encryption scheme has been reverse-engineered, and scripts for encrypting and decrypting firmware updates according to the scheme are available online. A user may circumvent the TPM by encrypting OpenWrt firmware files according to the required encryption scheme and installing the new firmware using the router's web interface.[7]

- The Meraki MR18 router uses encryption to prevent the installation of unauthorized firmware. To circumvent this, users must open the router's physical enclosure and connect a computer directly to the router's circuit board using a Universal Asynchronous Receiver Transmitter (UART) or Joint Test Action Group (JTAG) interface device. Using this interface, the user can gain root access to the device and flash an OpenWrt firmware file to the router's memory.[8]

ITEM E. ASSERTED ADVERSE EFFECTS ON NONINFRINGING USES

1. *The class of works affected and non-infringing uses*

The proposed exemption would permit owners of networking devices to install licensed software onto their devices. OpenWrt alone makes available over 3,000 different add-on software packages providing a wide range of functionality unavailable in the stock firmware of commercial networking devices.[9] All of the works in question are software applications protected by copyright and, in OpenWrt's case, are licensed under FOSS license terms that permit anyone to copy, modify, and redistribute them. For this reason, the installation and use of OpenWrt on a router is non-infringing, because it is a licensed use of the software.

2. *Adverse effects on non-infringing uses*

The technological protection measures limiting access to firmware on networking devices preclude non-infringing uses of copyrighted works in two ways. First, the TPMs restrict access to FOSS components pre-installed on the devices by their manufacturers, preventing device owners from exercising their corresponding rights under FOSS licenses. Second, the TPMs prevent device owners from installing other licensed software on their devices.

a. *Access to pre-installed FOSS components*

The stock firmware on the vast majority of consumer networking devices is built primarily from FOSS components. These typically include, for example, the Linux operating

[6] OpenWrt, *D-Link DIR-878 A1*, https://openwrt.org/toh/d-link/d-link_dir-878_a1; OpenWrt, *D-Link Recovery GUI*, https://openwrt.org/docs/guide-user/installation/installation_methods/d-link_recovery_gui.

[7] OpenWrt, *AXIMCom MR-102N*, https://openwrt.org/toh/aximcom/mr-102n.

[8] OpenWrt, *Meraki MR18*, https://openwrt.org/toh/meraki/mr18#flashing_method_c.

[9] OpenWrt, *Welcome to the OpenWrt Project*, https://openwrt.org/.

system kernel and BusyBox, a set of utilities to provide basic operating system functionality on embedded hardware devices. Both of these components are licensed under the GNU General Public License version 2 (GPLv2),[10] a license that permits every recipient of the software to exercise the "four freedoms" common to all FOSS: "the freedom to run, copy, distribute, study, change and improve the software."[11]

By distributing this software to customers who purchase their routers, manufacturers are obligated by the GPLv2 license to extend these freedoms to those customers, as well as to provide them with a copy of the FOSS components' source code and the means to install modified versions of the software.[12] These rights are pyrrhic in the face of TPMs that prevent their beneficiaries from exercising them, for example by installing their licensed modifications back onto their device.

a. Availability of other licensed software

The TPMs protecting the firmware on these devices also prevent users from installing new software on the devices, limiting them to the functionality pre-configured by the manufacturers. By circumventing the TPMs and installing a FOSS operating system like OpenWrt, devices can draw on an almost unlimited supply of FOSS applications to customize, improve, and extend their device's functionality.

Packages available via OpenWrt allow users to enable a host of features not typically available on commercial routers, including: router-level ad-blocking, secure report access via virtual private network (VPN), DNS encryption to secure outbound network traffic, parental controls and internet use time limits, network volume rate limits, traffic-shaping and quality-of-service prioritization to improve network performance and home automation support.[13] Each of these features performs a fundamentally non-infringing purpose and is provided by FOSS that cannot be installed on a TPM-controlled networking device without circumventing the TPM.

OpenWrt also ensures that software updates remain available for many networking devices after they cease to receive updates from its manufacturer. According to a database of supported devices updated daily by the OpenWrt project, the latest OpenWrt release, version 19.07.5, supports 418 networking devices that have been discontinued by their manufacturers.[14] The oldest of these devices was first supported by version 0.9 of the project, released in 2007.[15]

[10] *See* GNU, *GNU General Public License, version 2*, https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html.

[11] GNU, *What is free software?*, https://www.gnu.org/philosophy/free-sw.html.

[12] *See* GNU General Public License, version 2 § 3, https://www.gnu.org/licenses/old-licenses/gpl-2.0.en.html (requiring licensees to provide recipients with "the scripts used to control compilation and installation of the executable").

[13] *See* OpenWrt, Reasons to use OpenWrt, https://openwrt.org/reasons_to_use_openwrt#extensibility.

[14] *See* OpenWrt, OpenWrt supported hardware database, available at https://openwrt.org/_media/toh_dump_tab_separated.zip.

[15] *See* "Index of (root)/whiterussian/0.9," https://archive.openwrt.org/whiterussian/0.9/.

Over 80 of the currently-supported devices were first supported by versions of OpenWrt released before 2014.

1. *Evaluation of statutory factors*

a. *Availability for use of copyrighted works*

As demonstrated above, the proposed exemption is primarily focused on increasing device owners' access to copyrighted works for use on their devices.[16] We have primarily focused on OpenWrt and the over 3,000 FOSS packages that the project makes available to users who install it on their networking devices. However, our arguments apply with equal force to several other projects that produce FOSS firmware for use on networking devices, including DD-WRT, Tomato, OPNsense, PFSense, and VyOS.[17]

b. *The impact that the prohibition on the circumvention of technological measures applied to copyrighted works has on scholarship and research*

The proposed exemption would also promote research into the improvement of networking technology.[18] OpenWrt has been fundamental to a number of such research efforts. Each of these efforts depends upon researchers' ability to install an OpenWrt-based operating system onto consumer networking devices for the purpose of research and testing. TPMs that prevent the installation of alternative firmware impede these efforts.

The Bufferbloat Project researches and produces solutions to latency (network lag) issues resulting when networking devices overzealously buffer data (i.e. store network data temporarily in memory—a technique for improving performance that can hamper it when misused).[19] The project produced its own operating system based on OpenWrt, CeroWrt, "as a platform for further research into algorithms for solving state of the art problems in networking."[20] Through this effort, the project produced several improvements to networking technology that have been incorporated into OpenWrt and taken up by several networking and computing hardware manufacturers, including Qualcomm and Netgear.[21] This research has also served as the basis for proposed networking technology standards.[22]

The Homenet Project is also built on top of the OpenWrt codebase.[23] It is the testbed for the research of the IETF Home Networking Working Group, a standards working group

[16] 17 U.S.C. § 1701(a)(1)(C)(i).
[17] *See* Serdar Yegulalp, *Review: 6 slick open source routers*, InfoWorld, Feb. 1, 2018, https://www.infoworld.com/article/3106865/review-6-slick-open-source-routers.html.
[18] *See* 17 U.S.C. § 1201(a)(1)(C)(iii).
[19] *See* Bufferbloat, https://www.bufferbloat.net/projects/.
[20] *See* Bufferbloat, *Overview of the CeroWrt Project*, https://www.bufferbloat.net/projects/cerowrt/wiki/.
[21] *See* Bufferbloat, *CoDel Overview*, https://www.bufferbloat.net/projects/codel/wiki/.
[22] *See* IETF, *Controlled Delay Active Queue Management*, https://tools.ietf.org/html/draft-ietf-aqm-codel-10.
[23] *See* Homenet, *Homenet Technical Overview*, https://www.homewrt.org/about/overview.

chartered to develop networking protocol improvements to support "the evolving networking technology within and among relatively small residential home networks."[24]

c. Effect on the market for or value of copyrighted works

The proposed exemption supports the market for alternative FOSS firmwares for networking devices, which has developed over the last two decades in spite of the prohibition on circumvention. By affirming the legitimacy of circumvention by device owners for the purpose of installing their choice of licensed software on their devices, the exemption would promote the use of these works on a broader range of devices. Likewise, the value of manufacturer-supplied firmware will increase, as the research supported by FOSS firmware projects produces improvements to wireless technology generally.[25]

The exemption would have no negative impact on the market for the copyrighted works protected by the TPMs, i.e. the stock operating systems of networking devices. These works have no market independent from the devices they're sold on, and permitting their replacement with alternative software will not cause fewer devices to be sold. On the contrary, the evidence shows the opposite. The Linksys WRT54G router, the first model OpenWrt supported, gained a reputation for being easy to customize with FOSS firmware. Consumer demand kept it on the market longer than nearly any other consumer router.[26] Linksys even marketed its replacement, the WRT1900AC, as being "open source ready" and capable of supporting OpenWrt.[27]

d. Other factors

As on all computers, the software on networking devices must be updated regularly to fix security vulnerabilities. These updates are particularly critical on routers, which are connected to the Internet at all times, and serve as the entry point to home networks. But in a study released this year by the German communications research institute Fraunhofer found that the security practices of top home router manufacturers were "alarming."[28] In their review of the latest firmware from 127 commercially available router models, researchers found that, on average, devices had not received any software update in over a year.[29] The most-updated devices had at least 21 critical vulnerabilities, and at least 348 high-severity vulnerabilities.[30]

[24] *See* IETF, *Home Networking (homenet)*, https://datatracker.ietf.org/wg/homenet/about/.

[25] *See* Section 3(b), *supra*.

[26] Sebastian Anthony, *11 years on: Linksys cashes in on WRT54G popularity with overpriced WRT1900AC router*, ExtremeTech, Jan. 16, 2014, http://www.extremetech.com/computing/174875-11-years-later-linksys-cashes-in-on-wrt54gs-popularity-with-overpriced-wrt-1900ac-router.

[27] *Id*.

[28] *See* Perter Weidenbach, Johannes vom Dorp, *Home Router Security Report* (2020) at 1, *available at* https://www.fkie.fraunhofer.de/content/dam/fkie/de/documents/HomeRouter/HomeRouterSecurity_2020_Bericht.pdf.

[29] *Id* at 6.

[30] *Id* at 9.

When manufacturers fail to keep their devices secure, users can protect themselves and their networks by installing a FOSS firmware like OpenWrt. OpenWrt is regularly updated to fix known security vulnerabilities and to include the latest versions of the software packages it contains. In 2020 alone, the OpenWrt project issued six stable releases of the software, each correcting a number of security issues.[31] By installing OpenWrt, users gain timely security updates for not only late-model devices, but also for devices long-unsupported by their manufacturers, enhancing the security of their networks and extending the life of their hardware, potentially by many years.

DOCUMENTARY EVIDENCE

N/A

[31] *See OpenWrt 19.07*, OpenWrt.org, https://openwrt.org/releases/19.07/; *Security*, OpenWrt.org, https://openwrt.org/docs/guide-developer/security/.