GitHub
88 Colin P Kelly Jr Street
San Francisco, CA 94107

December 1, 2020

The Honorable Thom Tillis
Chairman, Senate Judiciary Subcommittee on Intellectual Property
113 Dirksen Senate Office Building
Washington, DC 20510

Dear Chairman Tillis,

GitHub offers these comments in response to your letter seeking public input on reform of the Digital Millennium Copyright Act (DMCA). We offer these comments on behalf of software developers who are building programs, websites, and applications that power industries all across America and the world.

GitHub is the largest software code hosting and software development platform in the world. Headquartered in San Francisco, GitHub enables more than 50 million developers, students, startups, small businesses, large companies, nonprofits, and governments around the world to host and collaborate on open source and proprietary software projects.

Our comments reflect three key themes, which will be expanded upon in direct answers to your questions.

First, DMCA reform should recognize that software developers and the platforms they use to distribute code have unique considerations. Software code hosting platforms are unique in that they are not designed for sharing popular consumer content like social media, but are instead professional or enterprise platforms designed for collaboration and technical innovation involving software code. Activity on GitHub is overwhelmingly legitimate, with 99.99% of all software projects not subject to copyright infringement complaints.[1] Computer programs are copyrightable. When the rightsholders of this content—software developers—choose to post their code publicly, including under an open source license, that means they want their code to be shared. Platforms like GitHub that host developers' content do not monetize the sharing of their content.

[1] GitHub, "2019 Transparency Report," https://github.blog/2020-02-20-2019-transparency-report/#projects-affected-by-dmca-takedown-requests.

When software code is taken down based on a claim of copyright infringement, removal of the code has significant and wide-ranging collateral impacts. Open source software hosted on platforms like GitHub is widely used, not just by individuals and companies for their own ends but also by developers who as a matter of course build on such software to create new products and solutions. Ninety-nine percent of all software projects are built using preexisting open source software components.[2] Such code is heavily relied upon and contributed to by industry: for example, 35 of the largest 50 companies in the world, by revenue, contributed code to open source projects on GitHub in 2019.[3] Thus, code hosted on GitHub often functions as pervasive digital infrastructure.

A seemingly narrow takedown that disrupts one project can severely disrupt millions of other software projects and the countless businesses and users that rely upon them. For context, the 50 most popular open source projects hosted on GitHub are relied upon by an average of 3.6 million other software projects.[4] Before a copyright claim takes down or otherwise grinds to a halt the development of a project utilized by industries, governments, and countless millions of developers, there should be a higher bar than a mere allegation. These particularities illustrate the importance of a flexible application of DMCA Section 512 so it does not impose a “one-size fits all online platforms” approach, as elaborated in our answer to Question 2.

Second, notice and staydown, which would effectively create obligations such as mandatory filtering, are a poor fit for platforms that host open source software projects and would undermine the interests of software developers, who are the rightsholders of software code. As noted above, *takedown* has significant collateral impacts for software code. Making that *staydown* would exacerbate those impacts, including to works that are not in fact infringing. This is a real likelihood given that notice and staydown would, in practice, require platforms to implement automated, pro-active filters, raising market and technical problems. Automated filters do not understand essential context for code hosting platforms—like open source license terms and fair use—meaning in practice developers’ code would be erroneously flagged without a

[2] Steven Vaughan-Nichols, “GitHub: All open-source developers anywhere are welcome,” *ZDNet,* https://www.zdnet.com/article/github-all-open-source-developers-anywhere-are-welcome/; Synopsys Open Source Security and Risk Analysis Report 2020, https://www.synopsys.com/blogs/software-security/2020-ossra-findings-infographic/.

[3] GitHub, “State of the Octoverse” 2019 report, section “At work,” https://web.archive.org/web/20201128070819/https://octoverse.github.com/.

[4] GitHub, “State of the Octoverse” 2019 report, section on “The interconnected community,” https://web.archive.org/web/20201128070819/https://octoverse.github.com/.

reasonable basis. These problems led the EU to expressly exclude "open source software developing and sharing platforms" from Article 17 of the 2019 Copyright Directive.

Third, any scrutiny of the DMCA's Section 512 needs to also include scrutiny of Section 1201. Activities of developers have changed dramatically since the statute was passed, and as it stands today, DMCA Section 1201 has a chilling effect on software innovation and many socially beneficial, non-infringing software use cases. For example, developers who pursue security research to identify vulnerabilities in software code face uncertainty on whether they will face liability. This uncertainty undermines the research and funding to pursue it in the first place. As a result, we are all left worse off as the digital infrastructure we rely upon is less secure than it would otherwise be. Thankfully, your questions acknowledge that this uncertainty in third-party assistance and the wording of exemptions should be clarified. What constitutes circumvention and a technological protection measure should also be more clearly defined. More generally, Section 512 was developed alongside Section 1201 as a balancing of interests. Changing Section 512 without a careful review of DMCA Section 1201 would greatly upset the careful balance of interests.

Responses to Specific Questions

Question 2

> *OSPs eligible for the safe harbor under section 512 are divided into four categories (conduits, caching services, hosting services, and web location tools) that can be both under-inclusive and over-inclusive. First, what types of OSPs should be covered to account for technological advances and business practice changes that have occurred during the past twenty-two years? Second, how should the categories be revised to better cover the types of OSPs that need—rather than just appreciate—the safe harbor's benefit? Among the possibilities would be to either increase the number of statutory categories to more explicitly cover specific types of service providers or to reduce the number of statutory categories, possibly to only one, and delegate authority to the Copyright Office to identify, by regulation, the covered types of service providers. If Congress were to take the latter approach, would this raise concerns about such authority being delegated to a non-presidentially-appointed Register?*

The different subsections of 512 describe different activities, and each of those activities should receive a safe harbor.[5] There may be subtypes of different service

[5] The reason for the division among 512(a) through 512(d) is not to address different categories of service providers, but to address the different conditions for safe harbor that should apply to different activities. For example, almost every service

providers within each of the statutory safe harbors; thus, a flexible approach that takes into account the circumstances is needed in applying 512. In its present form, Section 512 manages that task well.

For example, the 512(c) safe harbor relates to infringing content stored at the direction of a user. But not all services that may need the 512(c) safe harbor are identically situated. For example, services designed to host and share code—like GitHub—are uniquely different in users and content from other platforms, and the consequences of a particular removal or particular account termination raise issues very different from those of other platforms. The current statute demonstrates its suitability and flexibility, for example, in the requirement that removals be "expeditious" under the circumstances.

Question 3

> *Section 512 places the burden on copyright owners to identify infringing materials and affirmatively ask the OSP to remove the material or disable access to it. This burden appears to strike the correct balance, but the burden that the notice-and-takedown system itself places on copyright owners is too heavy; the system is also woefully inefficient for both copyright owners and service providers. I believe U.S. copyright law should move towards some type of a notice-and-staydown system—in other words, once a copyright owner notifies a service provider that a use of a copyrighted work is infringing, the service provider must, without further prompting, remove subsequent infringing uses absent a statement from the user (whether the copyright owner or not) that they believe the use is licensed or otherwise authorized by law (e.g., fair use). What are your thoughts on such a system, and how could it best be implemented?*

"Notice and staydown" is particularly ill-suited to code hosting platforms, because most copyright disputes arising on sites like GitHub are not about whether particular code may *ever* be posted publicly, but instead are about which contexts are permissible and which are not, applying the terms of the relevant licenses. For example, posting a piece of code licensed under the open source Apache license (which permits sharing but requires appropriate attribution) could be considered to constitute copyright infringement if it omits the required notice—but that is an easily remedied infraction that is ill-suited to a filtering requirement. Nor do developers of Apache-licensed code intend on removing such code as an automatic response—they would generally prefer to see the infraction remedied.

provider that provides storage of material at the direction of a user, and is thus protected by 512(c) with respect to a claim arising out of that storage activity, also acts as a conduit for that same material, and is thus protected by 512(a) with respect to a claim arising out of its role as a conduit. But the conditions for application of each of those safe harbors is different—with more requirements for the application of 512(c) to hosting activity than for the application of 512(a) to conduit activity.

A blunt “notice and staydown” system would sweep up content like copyrighted software code, but it is critical to understand that software developers have very different approaches to addressing how to remedy their infringement claims. Unlike other content owners, most open source software developers want their works to continue to be made available and not automatically blocked. Their goal is not “staydown” but rather remediation with little to no disruption of the availability of their code. As discussed in the opening comments, such disruptions can have significant collateral impacts.

Question 4

> *Starting from the place of the provisions that support the current notice-and-takedown system, a notice-and-staydown system would need to give more teeth to the knowledge standards and requirements for implementing a repeat infringer policy; to clarify that section 512(m)’s lack of a duty to monitor does not mean lack of a duty to investigate once notified and also that representative list and identifiable location do not require as much detail as courts have required; and to provide better mechanisms for users to contest a takedown as authorized by a license or by law. How would you revise or add to the existing provisions in section 512 to accomplish this or, if this could better be achieved by starting from scratch, what new legislative text do you think would best accomplish this?*

Section 512, in its current form, has proved flexible enough for platforms like GitHub to address concerns in an appropriate manner. We caution against imposing a duty to “investigate” beyond the duty to remove particular content identified at a particular location. This is both because, as we discuss above, machines can’t consider context, and because if such an investigation is to consider context, copyright holders are best-positioned to know it. Shifting this task to service providers would be exceptionally labor-intensive, and less reliable for them to perform.

Question 6

> *More generally, the notice- and counter-notice sending process have many shortcomings. These could be improved by clarifying when automation is appropriate and that OSPs cannot erect requirements beyond those in section 512(c)(3); by authorizing the Copyright Office to develop standardized web forms for notices and counter-notices and to set regulations for the communications that OSPs must deliver to a user when their content is taken down or had access disabled (including offering information about the fair use doctrine as codified in section 107 and as illustrated in the Copyright Office’s Fair Use Index); and by increasing privacy protections for notice and counter-notice senders by masking certain personally identifiable information, including address and phone number.*

> *How could this best be done? Please provide specific provisions for accomplishing these goals.*

With respect to masking of personally identifiable information, we think that requiring both notice and counter-notice senders to provide an electronic mail address is a good compromise between the need for notice and counter-notice senders to contact one another and the need for privacy, since masked electronic mail addresses are widely available for situations where sensitive privacy issues arise.

With respect to the implementation details of web forms, different services may justify different web forms and different practices. To the extent a standardized form is implemented, while some information would always be the same (such as an acknowledgement that the sender has considered fair use), the requirements would have to be flexible, since different service providers may need different information. For example, a code hosting service like GitHub needs to know the specific portion of code alleged to be infringing, rather than receiving a notice pointing at an entire software project, which may contain thousands or millions of lines of codes from different sources.

Question 11

> *Section 1201 currently allows for temporary exemptions to be granted from the circumvention prohibition, but those exemptions do not extend to third-party assistance. This means that when the Librarian of Congress grants an exemption for circumvention of technological protection measures (TPMs) over software for a tractor to allow for repair, the tractor owner must perform the software repair themselves. The Copyright Office has recommended amending the statute to grant the Librarian authority to adopt temporary exemptions permitting third-party assistance "at the direction of" an intended user, and this may be the right way to address this problem. Do you agree with the Copyright Office? If so, how should this provision be drafted to avoid unintended consequences, and to what extent is the Unlocking Consumer Choice and Wireless Competition Act a helpful model? If not, please explain why you do not agree and provide specific recommendations as to how you think this problem should be addressed?*

We agree with the Copyright Office that exemptions permitting third-party assistance would be a helpful step. For example, developers collaborating to create software that is compatible with other preexisting software would benefit from being able to give and receive third-party assistance with noninfringing acts of circumvention for otherwise lawful purposes.

Question 12

> *The Copyright Office has recommended revising some of the permanent exemptions so that they are better tailored to the types of uses sought today. In particular, the exemptions for security testing and encryption research should be revised to expand the types of activities permitted, ease the requirements to seek authorization from the owner of the relevant system or technology, and eliminate or clarify the multifactor tests for eligibility. What thoughts do you have about revising these existing permanent exemptions, and how would you recommend that be done?*

We agree with the Copyright Office that aspects of 1201 should be further scrutinized to ensure that 1201 is properly balanced, and does not discourage or chill important innovations including security research. Good faith security research encourages more secure software and responsible reporting of security vulnerabilities. For that reason, improvements to the permanent exemption for security research would serve the public interest.

Question 13

> *Congress should adopt new permanent exemptions for noninfringing activities that have repeatedly received exemptions in recent triennial rulemakings, or where there is a particularly broad-based need, including to enable blind or visually impaired persons to utilize assistive technologies and to allow diagnosis, repair, or maintenance of a computer program, including to circumvent obsolete access controls. What other temporary exemptions should be made permanent?*

We agree with the Copyright Office that these would be helpful steps. As discussed above, a broad and permanent exemption for all noninfringing security research would benefit developers and the public interest. Indeed, all of the current temporary exemptions should be made permanent, since there is an established record that those exemptions are justified.

Question 14

> *There are various ways that the triennial rulemaking process could be streamlined to be more efficient and so that section 1201 better accounts for user concerns. These include establishing presumptive renewal of exemptions adopted in the previous rulemaking cycle, shifting the burden to those who want to oppose an exemption from the previous rulemaking, and authorizing the Librarian, upon recommendation of the Register, to make permanent a temporary exemption that has been renewed twice without opposition and without modification. How ought section 1201 be revised to reflect the stakeholder desire for a less burdensome triennial rulemaking process and consumer interests, and what other means should be adopted to make the rulemaking process more efficient?*

For the reasons we stated in response to questions 12 and 13, we agree with the Copyright Office that these would be helpful steps.

Question 15

> *Though it did not receive as much attention during my hearings as sections 512 and 1201, section 1202 is another important part of copyright law added to title 17 by the DMCA, and it too is in need of modernizing. For example, Congress could amend section 1202 to drop the double-intent standard and only require a copyright owner to prove that a defendant removed or altered rights management information (knowingly or not) with the knowledge that it would encourage infringement. And Congress could adopt the Copyright Office's recommendation to enact a new section 1202A to provide the author of a copyrighted work—rather than just the copyright owner—with a right of action when someone removes or alters rights management information with the intent to conceal an author's attribution information. Do you think that the proposed legislative text that appears on page 98 of Authors, Attribution, and Integrity: Examining Moral Rights in the United States is the best way to add a right for the copyright owner, or would you recommend different text? And what are your thoughts on revising section 1202's double-intent standard?*

GitHub has concerns about the application of potential 1202A to open source software. Many common open-source software licenses, such as the MIT and BSD licenses, require that information about the copyright holder be preserved but do not require that information about each author be preserved. Enacting 1202A might well gum up the works of open source software development by creating situations where downstream developers who comply with all of the requirements of the relevant open-source license might still be the target of valid claims from individuals who wrote code but who are not identified by name in the copyright notice. Where popular open source projects on GitHub have thousands of contributing developers, this clearly becomes unmanageable for content like code.[6]

Congress should not lower the mental-state requirements of section 1202, either. Many of the disputes that arise in the context of open-source software are disputes about attribution or other metadata falling within the statutory definition of "copyright management information." Those disputes are virtually always worked out between the developers. Lowering the required mental state would put code hosting platforms at risk because they could be regarded as distributing material from which copyright management information was removed without

[6] GitHub, "State of the Octoverse" 2019 report, section on "Community trends," https://web.archive.org/web/20201128070819/https://octoverse.github.com/.

authorization, but would not have knowledge of that removal. That would leave “reasonable grounds to know that it will induce, enable, facilitate, or conceal” infringement as the only mental-state requirement, and that is much too lenient a standard standing alone.

* * *

GitHub thanks you for the opportunity to respond to your questions and discuss how proposed changes to the DMCA have a unique impact on software developers and code hosting platforms.

Respectfully submitted,

Mike Linksvayer
Head of Developer Policy
GitHub