BURSOR & FISHER
P.A.

888 Seventh Avenue
New York, NY 10019
www.bursor.com

Alec Leslie
Tel: 646.837.7150
Fax: 212.989.9163
aleslie@bursor.com

July 30, 2021

Via ECF

The Honorable Rozella A. Oliver
United States District Court
Central District of California
Roybal Federal Building and United States Courthouse
255 E. Temple St., Courtroom 590, 5th Floor
Los Angeles, CA, 90012
RAO_Chambers@cacd.uscourts.gov

Re: *Saleh v. Nike, Inc., et al.*, Case No. 2:20-cv-09581-FLA-RAO (C.D. Cal.)
Plaintiff's Statement Pursuant To The Court's July 23, 2021 Order (ECF No. 62)

Dear Judge Oliver:

Pursuant to the Court's July 23, 2021 Order (ECF No. 62), Plaintiff submits this letter brief asking the Court to compel discovery of Defendant FullStory, Inc.'s ("FullStory") source code used during the relevant period on nike.com, as requested in Plaintiff's RFP No. 15.[1] As detailed below, the source code is the single most important evidence in this case, and the protective order signed by Your Honor in this matter includes significant measures for the safe disclosure of source code *that FullStory requested*. FullStory's other arguments against withholding the code are boilerplate and lack merit. FullStory's proposed alternatives to disclosure of its source code are prejudicial, invite further discovery disputes, and have been rejected by other courts.

I. Fullstory's Code Is Relevant

Courts routinely grant motions to compel disclosure of source code where, as here, it is relevant to the parties' claims or defenses. *Forterra Systems, Inc. v. Avatar Factory*, 2006 WL 2458804, at *1 (N.D. Cal. Aug. 22, 2006) ("[b]ecause the source code is at the heart of the dispute, [Plaintiff's] expert must have access to the entire source code."); *see also Keithley v. Homestore.com, Inc.*, 629 F. Supp. 2d 972, 975 (N.D. Cal. 2008) (finding the "source code would be relevant to plaintiffs' infringement claims because source code, along with

[1] RFP No. 15 requests "All versions of the source code, including version numbers and deployment dates, for Session Replay as utilized on nike.com. This request includes any 'front end' code deployed on nike.com [i.e, such as JavaScript], and any "back end' code residing on any server, and any other proprietary code necessary for end-to-end functionality of Session Replay."

functionality, implementation and design documents, helps establish how the accused websites operate."); *Burnett v. Ford Motor Co.*, 2015 WL 1527875, at *4 (S.D. W. Va. Apr. 3, 2015) ("The source code is the foundation of the ETC's operating system; accordingly, it is relevant."); *Quest Integrity USA, LLC v. A.Hak Industrial Services US, LLC*, 2016 WL 4533062, at *3 (W.D. Wash. Mar. 23, 2016) ("Relevance has been found where a party's device operation theory implicates its source code, or where the source code bears on the operation of a product."); *id.* ("Because inspection of A.Hak's source code and executable code could lead to Quest's discovery of evidence to support its claims against A.Hak, the Court finds that the source code and executable code are relevant").

Here, there are two separate reasons FullStory's code is relevant. First, this is a wiretapping case, and FullStory's "Session Replay" software is the alleged wiretapping device. Plaintiffs allege Defendant Nike, Inc. ("Nike") hired Defendant FullStory to secretly make recordings of everything anyone does on Nike's website, nike.com. Using its "Session Replay" software, FullStory "watch[ed] and record[ed] a visitor's every move on a website, in real time," with the software providing "pixel-perfect playback." FAC ¶¶ 17, 18-19, 24. FullStory also can monitor website visitors live where (in FullStory's own words) "you'll essentially be riding along in near real time." *Id.* ¶¶ 28.

One of the elements of Plaintiff's claim under Cal. Penal Code § 631(a) is whether the software "reads or attempts to read or learn the contents or meaning of any … communication while the same is in transit or passing over any wire, line or cable or is being sent from or received at any place within [California]." Plaintiff also must prove that the software is a "device" within the meaning of Cal. Penal Code § 635. The FullStory software that worked on nike.com is directly relevant to these elements.

Second, a party cannot raise a defense and then refuse to produce information necessary to test that defense. *See*, *e.g.*, *MLC Intellectual Prop. LLC v. Micron Tech, Inc.*, 2018 WL 6175982, at *3 (N.D. Cal. Nov. 26, 2018) ("The Court agrees that Micron has put those sales at issue by asserting that allegedly infringing sales by subsidiaries and affiliates were permitted by licenses").

Both Defendants have put FullStory's software at issue by raising a host of factual arguments about how that software works. They argue FullStory's software "did not collect the 'contents' of Plaintiff's purported communications." MTD, at 15:9-11 (Dkt. No. 30). They deny that FullStory captures and records sensitive information like payment card information. *Id.* at 16:15-26. They admit researchers found that the security of FullStory's software is "imperfect," but deny that there was a problem on nike.com. *Id.* at 16:25-26. They argue that whether FullStory may be deemed to be a party to the communication (and therefore not liable), turns on the "technical context" of how the software is used, and that "Nike uses FullStory's software the same way" that other courts have "made clear" are lawful. *Id.* at 14:14-22. They suggested liability here would "criminalize" the internet because FullStory's software is no different than any other analytical software used by website operators. *Id.* at 14:23-15:5. They argue FullStory's software cannot determine the state where anonymous website visitors are located, despite FullStory's marketing materials saying that its software can provide that information "fairly accurately."[2] Other defendants in similar cases have argued their session recording software does not capture information while it is "in transit or passing over any wire," and Defendants presumably will make the same argument here.

Plaintiff has offered to drop any discovery of FullStory's code if both Defendants stipulate that FullStory's software satisfies the technical requirements of §§ 631(a) and 635, and that they will not raise arguments or defenses about how the software works. Defendants declined and will continue to make whatever factual arguments they want about the software. Accordingly, taking discovery on how the source code operates on nike.com is necessary to the determination of this action.

II. The Protective Order Specifically Contemplates Disclosure Of Source Code Information

In its discovery responses, FullStory objected to the "burden and risks" of disclosing confidential source code information. However, "[t]here is no absolute privilege for trade secrets and similar confidential information." *DIRECTV, Inc. v. Trone*, 209 F.R.D. 455, 459 (C.D. Cal. 2002). The existence of a protective order is sufficient to overrule objections concerning the proprietary or confidential nature of source code. *See, e.g.*, *Indacon, Inc. v. Facebook, Inc.*, 2012 WL 12862665, at *2-3 (W.D. Tex. May 7, 2012) ("Finally, although Facebook argues the confidentiality of these documents will be compromised if produced to Indacon, the PO would appear to more than adequately address Facebook's concerns."); *WeRide Corp. v. Kun Huang*, 2019 WL 5722620, at *7 (N.D. Cal. Nov. 5, 2019) (requiring production of the "complete source code repositories" and noting that protective order provided adequate protections).

What is more, *at FullStory's request*, Plaintiff agreed to a protective order ("PO") that contemplates the production of the source code data, and this Court signed that PO on June 23, 2021 (Dkt. No. 55). The PO is based on the Northern District of California's model protective order for litigation involving patents, highly sensitive confidential information, and/or trade secrets. Among other measures to protect confidentiality, source code is to be made available via onsite inspection. *Id.* at § 8. To examine the source code, Plaintiff must also, among other procedures, (i) provide early disclosure of his software experts, including the expert's employees and any case the expert has appeared in in the last five years, and (ii) allow FullStory to object to those experts. *Id.* at § 7.4(a). These are optional and onerous provisions under the model protective order that Plaintiff was not required to agree to. That Plaintiff agreed to such terms demonstrates the necessity of examining the source code in this case.

This Court has also correctly held that boilerplate and unsupported objections of undue burden are "improper." *Esebag v. Whaley*, 2019 WL 6604869, at *1 (C.D. Cal. Aug. 9, 2019) (Oliver, J.). Nearly three months have passed since FullStory served its written objections, and during this time, FullStory never produced evidence of any burden it will face by allowing the inspection procedures that *FullStory asked to be included in the PO*. *eDirect Publ'g, Inc. v. LiveCareer, Ltd.*, 2014 WL 2527401, at *2 (N.D. Cal. June 4, 2014) (overruling objections to

[2] *See* https://help.fullstory.com/hc/en-us/articles/360045926353-IP-Address-Geolocation.

request to inspect source code where party presented "[n]o specific or concrete information" backing up their suggestion of an "undue burden," such as the costs involved); *In re Google Litig.*, 2011 WL 286173, at *6 (N.D. Cal. Jan. 27, 2011) (rejecting burden argument because defendant provided "no estimate of the number of hours, business distraction, or cost of production"); *accord Pom Wonderful LLC v. Coca-Cola Co.*, 2009 WL 10655335, at *3-4 (C.D. Cal. Nov. 30, 2009).

III. Other Objections Raised By FullStory Lack Merit

FullStory's other objections are easily dispensed with. FullStory objects to Plaintiff's request for the source code as "vague and ambiguous in the use of the phrases 'version numbers,' 'deployment dates,' 'front end,' 'back end,' 'other proprietary code,' and 'necessary for end-to-end functionality.'" But "[t]he party objecting to discovery as vague or ambiguous has the burden to show such vagueness or ambiguity by demonstrating that more tools beyond mere reason and common sense are necessary to attribute ordinary definitions to terms and phrases." *Thomas v. Cate*, 715 F. Supp. 2d 1012, 1030 (E.D. Cal. 2010); *see also Castle v. Lugo*, 2020 WL 5356935, at *2 (C.D. Cal. June 19, 2020) ("A party responding to discovery should use common sense and attribute ordinary definitions to terms in discovery requests."). As a technology company, it is implausible that FullStory would not know the meaning of these terms, and it never pressed its vagueness arguments during the meet and confer efforts. The objection also is improper boilerplate. *Cf. Esebag*, 2019 WL 6604869 at *1.

FullStory also objected that "this is not a patent or other intellectual property dispute in which plaintiff asserts some ownership or proprietary interest in FullStory's code." However, after three months, FullStory has provided no authority that those circumstances are the *only* circumstances where software code can ever be deemed relevant. The objection also ignores the many ways that Defendants have put the code at issue, as shown in Section I above.

IV. FullStory's Alternative Proposals Are Prejudicial

During the meet and confer efforts, FullStory proposed two alternatives to disclosing its source code pursuant to the procedures in the PO. Both are prejudicial because they allow Defendants to say whatever they want about FullStory's code, while dictating which evidence Plaintiff's experts can use to test or challenge those statements.

FullStory first offered to provide "high level" architecture documents and manuals demonstrating how FullStory's software generally works. But FullStory is already required to produce this information in connection with Plaintiff's RFP No. 16.[3] The *whole purpose* of RFP 16 is to help Plaintiffs' expert understand the code requested in RFP No. 15 by providing a general, macro-level overview of the code. Standing alone, high-level information about how FullStory's code generally operates does not disclose all relevant details of how FullStory

[3] RFP No. 16 seeks "[a]ll documents describing high-level software architecture that illustrates communications or data transfer between Nike, FullStory, and all other third-party entities used to facilitate the end-to-end functionality of Session Replay."

operated on nike.com during the relevant period. At least one court has correctly rejected similar proposals advanced by FullStory here:

> Ford's argument that Plaintiffs are not entitled to the source code because they already have "the design, modification, and testing documents related to the ETC system that are relevant to and necessary for the analysis and development of their defect theory" is not persuasive. In Ford's view, because Plaintiffs have these materials, they do not need the source code, thereby rendering the source code irrelevant. However, Plaintiffs should not be forced to rely on Ford's determination as to what is the "most relevant" evidence in its possession. The source code is not duplicative or cumulative. Rather, it reflects an important step in the production of the ETC system between the system's design and the testing phases.

Burnett, 2015 WL 1527875, at *5.

Second, FullStory offered to disclose portions of the code that it deems relevant to Plaintiff's claims or to any arguments it may make about its code. But letting FullStory unilaterally dictate which portions of the code are relevant is like letting the proverbial fox guard the henhouse. Source code consists of software programming instructions, and it is common for the source code to comprise many separate programing files, many of which are often individually small in size (perhaps a few thousand characters or less per file). A program is rarely comprised of a single file of source code. There are typically a great many files of code, and the program's instructions require multiple files to be processed for the instructions to be properly executed. For example, it is normal for a variable to be declared in one file, an interface[4] declared in a second file, and a method using the variable and the interface together in a third file. The only way the behavior of the code can be understood is for all three of these files to be present so the execution can be traced. If any of the files are missing, the code cannot be analyzed fully, and the program's behavior will remain opaque to the Court.

Thus, by withholding portions of source code, a producing party not only dictates the contents of evidence to the Court, but also can impair the analysis of any portions that are produced. This invites protracted discovery disputes and requests to extend the discovery period, because the producing party often has incentive to obstruct or constrain the production process. The onsite inspection procedure that FullStory demanded also increases the cost of any such disputes, because it would require Plaintiff's experts to travel multiple times if there is a dispute over the portions FullStory discloses. These problems can be avoided simply by permitting Plaintiff's experts direct access to the code, as contemplated by the terms of the protective order that FullStory requested here.

Respectfully Submitted,

Alec M. Leslie

[4] An "interface" in programing is similar to a template.