**IN THE UNITED STATES DISTRICT COURT**
**FOR THE SOUTHERN DISTRICT OF WEST VIRGINIA**

**HUNTINGTON DIVISION**

**CHARLES JOHNSON, et al.,**

**Plaintiffs,**

**v.** **Case No.: 3:13-cv-06529**

**FORD MOTOR COMPANY,**

**Defendant.**

## MEMORANDUM OPINION and ORDER

On April 3, 2015, the Court denied Ford Motor Company's ("Ford") motion for a protective order that would have prohibited Plaintiffs from discovering Ford's Electronic Throttle Control ("ETC") system source code for various makes and models of Ford vehicles.[1] (ECF No. 421 at 1). However, the Court granted "Ford's Motion to the extent that it request[ed] a protective order, separate from the universal protective order already entered in this case, (ECF No. 169), which is tailored to fit Ford's particular concerns related to sharing its highly proprietary commercial information with Plaintiffs." (*Id.* at 2). Accordingly, the Court ordered the parties to meet and confer on the subject of a special protective order concerning the production of Ford's ETC source code. (*Id.* at 1-2). The parties were required to submit an Agreed Order Governing

[1] "Computer programs are made up of lines of text written in a computer language, called the 'source code' of that program." *SAS Institute Inc. v. World Programming Ltd.*, --- F.Supp.3d ----, 2014 WL 6978300, at *2 (E.D.N.C. Sept. 29, 2014). Although this opinion references a singular "ETC source code," Ford has represented to the Court that different versions of the ETC source code exist for each specific model line and possibly each specific type of vehicle within a model line (e.g. different engine types). (ECF No. 531 at 10). Ford possesses these different versions, including past versions, of the ETC source code. (*Id.* at 12).

Disclosure of Ford's Source Code to the Court by April 13, or in the event that no agreement could be reached, to submit their respective versions to the Court by the same date. (*Id.*)

Since the Court's April 3, 2015 order, the parties have been unable to agree on a protective order governing the production of Ford's ETC source code, and instead, have submitted differing proposed protective orders. During a telephonic discovery conference on April 29, 2015, the parties informed the Court that they principally disagreed over the format in which Ford was to produce the ETC source code. Plaintiffs requested the code in "write-access"[2] format, while Ford argued that a "read-only" format was more appropriate. The undersigned permitted the parties to submit briefing on the issue, (ECF Nos. 502 & 504), and held a hearing related to ETC source code discovery on May 28, 2015, (ECF No. 531). After the hearing, Plaintiffs supplied the undersigned with a list of computer software programs (or as Plaintiffs refer to them, "tools") that they assert are necessary to effectively and efficiently analyze the ETC source code, and Plaintiffs designated on that list which format the source code would have to be produced in for the specific tool to work. Ford responded to Plaintiffs' list by providing the undersigned with a document either agreeing or disagreeing with the use of each tool and asserting its own opinion as to which format the ETC source code must be in to utilize each tool.[3]

[2] Throughout this opinion, "write access" means that the source code is in a format that provides the recipient of the code with the ability to change or edit the source code, including adding lines of code to, the source code.

[3] Subsequent to Ford's submission, Plaintiffs filed a Motion for Leave to Reply to Ford Motor Company's June 3, 2015 Source Code Letter, (ECF No. 533), to which Plaintiffs attached a copy of their proposed reply brief. The Court **GRANTS** Plaintiffs' Motion for Leave to Reply and will consider Plaintiffs' reply brief.

The Court has thoroughly considered the arguments of the parties and **ORDERS** that Ford must produce the ETC source code in read-only format and need **not** provide Plaintiffs with write access to the ETC source code. Consequently, and keeping in mind the guidance in this opinion as to the security protocols necessary for source code review, the parties are **ORDERED** to meet and confer regarding the remaining terms of the special protective order and shall tender an Agreed Order Governing Disclosure of Ford's Source Code to the undersigned no later than **June 22, 2015**. If the parties are unable to agree on the terms not resolved by this opinion, they shall submit their respective versions to the undersigned by electronic mail no later than 12:00 noon on June 22, 2015, and the disputed issues shall be addressed at the regularly scheduled telephone conference on June 24, 2015.

## I. Relevant Facts

These cases involve alleged events of sudden unintended acceleration in certain Ford vehicles manufactured between 2002 and 2010. In particular, Plaintiffs claim that their vehicles were equipped with defective ETC systems which were not fault tolerant, resulting in open throttle events during which the drivers of the vehicles lacked the ability to control the throttles. Plaintiffs assert that the mechanisms causing the throttles to open unexpectedly were numerous, including electromagnetic interference, resistive shorts, and other voltage and resistance fluctuations, and that these issues were known to Ford. Despite having knowledge of the potential for sudden unexpected acceleration, Ford nonetheless failed to properly design the ETC system to correct the events when they occurred, and further neglected to install fail-safes, such as a Brake Over Accelerator system, that would allow the drivers to physically prevent or mitigate sudden acceleration.

In the course of discovery, Plaintiffs requested that Ford produce for inspection and review the source code for the ETC system. Ford objected, and the parties were unable to resolve the issue in informal discussions. Ford then filed a motion for protective order, asking the Court to prohibit disclosure of the source code in its entirety. The Court denied Ford's motion and ordered the parties to meet and confer regarding the terms for production of the ETC source code under a special protective order. The parties were not able to agree on the terms of a special protective order, and instead submitted proposed protective orders and briefs outlining their respective positions. After discussing the issue no less than twice during telephonic discovery conferences, the Court held a hearing on the issue where testimony was provided by two expert witnesses—Nigel Jones for Plaintiffs and Dr. John Kelly on behalf of Ford.

Mr. Jones earned a Bachelor of Science degree in Electrical and Mechanical Engineering from Brunel University in London, and he is currently employed at the Barr Group as Chief Engineer and at R.M.B. Consulting, Inc., as its president. (ECF No. 496-1 at 1-3). In these positions, Mr. Jones has designed both hardware and firmware for various devices.[4] (*Id*. at 1-2). Mr. Jones has also performed "expert witness work," including reverse engineering of hardware and software, as well as code review. (*Id*.) At the hearing, Mr. Jones performed a demonstration of the typical steps taken in reviewing source code. He utilized integrated development environment software,[5]

[4] Firmware is defined as "computer programs contained permanently in a hardware device (as a read-only memory)." Merriam-Webster Online Dictionary, http://www.merriam-webster.com/dictionary/firmware (last visited June 10, 2015).

[5] An integrated development environment is "a programming environment that has been packaged as an application program, typically consisting of a code editor, a compiler, a debugger, and a graphical user interface (GUI) builder." Search Software Equality, http://searchsoftwarequality.techtarget.com/definition/integrated-development-environment (last visited June 10, 2015).

which permitted him to easily search, compile,[6] and debug[7] source code. (ECF No. 531 at 47-51). In relation to reading and searching the code, Mr. Jones stressed that some type of integrated development environment must be used; otherwise, attempting to read the code would be extremely time consuming given that (1) a skilled engineer can only read and understand 100 lines of code each day, and (2) Ford's ETC source code presumably contains around one million lines of code. (*Id*. at 53). With regard to the use of a compiler, Mr. Jones displayed how a compiler can detect errors in the code during the compiling process and provide warnings at the completion of compiling. (*Id*. at 56). Because many different types of compilers exist, Mr. Jones testified that he would need to obtain and use the same compiler that Ford uses when analyzing source code. (*Id*. at 60). Mr. Jones also emphasized the importance of using a debugger to test "timing relationships" within the code, "race conditions," and "stack usage."[8] (*Id*. at 50-51).

After explaining the usefulness of a debugger, Mr. Jones demonstrated how code may be analyzed using a software simulator and the process of fault injection.[9] (*Id*. at

---

[6] A compiler is "[s]oftware that translates a program written in a high-level programming language ... into machine language." PC Magazine, http://www.pcmag.com/encyclopedia/term/40105/compiler (last visited June 10, 2015); *see also* American Heritage Dictionary Online, https://ahdictionary.com/word/search.html?q=compiler (last visited June 10, 2015) (defining compiler as "[a] program that translates another program written in a high-level language into machine language so that it can be executed."); (ECF No. 531 at 48). In other words, a compiler translates source code into binary code or object code that can be understood by a computer.

[7] "Debugging software means locating the errors in the source code." PC Magazine, http://www.pcmag.com/encyclopedia/term/41022/debug (last visited June 10, 2015).

[8] A stack is "[a] set of hardware registers or a reserved amount of memory used for arithmetic calculations, local variables or to keep track of internal operations (the sequence of routines called in a program). For example, one routine calls another, which calls another and so on. As each routine is completed, the computer returns control to the calling routine all the way back to the first one that started the sequence." PC Magazine, http://www.pcmag.com/encyclopedia/term/51954/stack (last visited June 10, 2015).

[9] Mr. Jones described what he meant when using the term simulator: "I'm just talking about a simulator that will execute software, recognizing that's [*sic*] not interacting with any real hardware. Because it allows me to do things like measure how long a task takes to execute, measure how much stack size they use, and so on." (ECF No. 531 at 61).

59-60). By using a software simulator, Mr. Jones explained that he could "change[] [Ford's] code to force it to accept two inputs that then get processed," so that he could see the affect that those changes have on the output. (*Id*. at 60). Mr. Jones acknowledged that by forcing the code to accept certain values, the resulting code was not the same code that is contained in Ford's vehicles; however, he asserted that he would perform such testing if he were developing code for Ford and that fault injection using simulation software was necessary as some faults are "very difficult to create in physical hardware." (*Id*. at 59-60). When questioned by the Court whether the inputs that Mr. Jones required the code to accept could ever occur in "real life," Mr. Jones responded: "[T]ypically, if, for example, you have a frayed wire, a loose connection, or you've got a lot of interference from a cell tower, signals can take on just about any value. And so if I was designing something like [ETC source code], I would design the software to accommodate any value." (*Id*. at 63). Mr. Jones then further explained his method of testing code using a simulator: "I would examine the code. I would develop some hypothesis about maybe this code doesn't work the way it should. And then I would potentially construct an experiment to test that hypothesis. Now, sometimes I can run the test purely as a thought experiment. Other times I do need to go into the simulation environment because it is so complicated and say, I am going to put these numbers in and see what comes out the other end." (*Id*. at 64). Mr. Jones insisted that any changes made to source code can "always" be identified and presented one program that tracks source code changes. (*Id*. at 57). With regard to source code format limitations, Mr. Jones testified that he could use search and "comparison" tools on read-only source code, but could not use a debugger on read-only code because "you have to be able to create [] files in order to run the debugger." (*Id*. at 65). It is also apparent that

the fault injection in the way described by Mr. Jones could not occur with read-only code.[10]

Dr. Kelly testified that he is a software engineer and that he has experience working with nuclear reactor safety shutdown systems, flight control systems, medical devices, radiation devices, train control systems, and diesel engine control systems. (*Id*. at 81, 92). He stated that he had been hired as an expert in a number of cases "to assess and analyze" source code. (*Id*. at 81). Dr. Kelly asserted that he had "signed onto many protective orders" in some of those cases and that Ford's proposed protective order for producing its ETC source code was "entirely consistent" with those past protective orders. (*Id*. at 82). According to Dr. Kelly, Plaintiffs are "looking for conditions that would cause the software to fail," which can be accomplished by examining the software, developing a hypothesis, and then "actually testing the real binary on the real PCM [Powertrain Control Module] to see if that can occur." (*Id*. at 83). Dr. Kelly explained that such testing could be performed on the hardware module, which is connected to a computer, by forcing inputs into various input pins through the application of voltages to specific places on the hardware module. (*Id*. at 84). He insisted that write access to the source code was unnecessary to perform the testing that Plaintiffs desired and that there is no need "to change the code in order to test the binary." (*Id*. at 83-84). In response to Mr. Jones's demonstration where he changed certain portions of the code,

[10] At the conclusion of Mr. Jones's testimony, the Court inquired of Plaintiffs' counsel whether Plaintiffs' experts had performed black-box testing on any of the purportedly defective vehicles. (ECF No. 531 at 71). Black-box testing is the "[t]esting [of] software based on output requirements and without any knowledge of the internal structure or coding in the program." PC Magazine, http://www.pcmag.com/encyclopedia/term/38733/black-box-testing (last visited June 10, 2015). According to Dr. Kelly, black-box testing may be utilized to test usability problems, concurrency errors, initialization errors, termination errors, ordering errors, and timing errors. (ECF No. 531 at 87-88). Plaintiffs' counsel responded that testing "akin" to black-box testing had been performed on two vehicle models and that Plaintiffs' experts were able to recreate sudden unexpected acceleration events during that testing. (*Id*. at 72, 75).

Dr. Kelly testified that the "numbers set in the [binary] code cannot change" in the real world. (*Id*. at 84).

In relation to whether a read-only format would obstruct Plaintiffs' experts' ability to analyze the source code, Dr. Kelly asserted that the integrated development environment utilized by Mr. Jones could still be used with read-only access to the source code, as could a compiler and a debugger. (*Id*. at 86, 98, 100). Dr. Kelly also testified that read-only source code could be "put out on a simulator, as an emulator," and be tested.[11] (*Id*. at 87). However, Dr. Kelly acknowledged that read-only source code could not be rewritten to cause it to fail, and could not be changed to inject faults into the code. (*Id*. at 89). On cross-examination, Dr. Kelly opined that whether the source code was produced in read-only format or write-access format would "make no difference" to the type of code analysis that he thought should be performed in this case. (*Id*. at 94). Dr. Kelly explained that "if [the source code] is writable, then you run into the problem with changing the code and tracking those changes and determining what code exactly it is that you are testing. … The problem is, in a complex system … tracking all of the changes to the source code and how they are propagated down to the binary is very difficult." (*Id*. at 94-95). Dr. Kelly acknowledged that it was possible to track any changes made to the source code, but insisted that changing the code "adds an incredible complexity and it becomes extremely difficult to find out what files are actually on a binary. There are potentially thousands of files that can be changed in many different ways, and it becomes next to impossible to determine accurately which file made up a particular binary. …[I]t becomes extremely difficult to figure out what it

[11] An emulator is "[h]ardware, software or a combination of the two that enables a computer to run programs for another platform. …For example, Apple's iOS 'simulator' and Google's Android 'emulator' are both software utilities that run their respective mobile apps in the computer for testing purposes." PC Magazine, http://www.pcmag.com/encyclopedia/term/42579/emulator (last visited June 10, 2015).

is you did at all." (*Id*. at 96-97).

On the subject of efficiency in analyzing the source code, Dr. Kelly opined that having the source code in write-access format would not decrease the amount of time necessary to perform most of the tasks that Mr. Jones demonstrated. (*Id*. at 99). Dr. Kelly remarked that the only process made easier with write-access source code would be testing the code by "forc[ing] it to have certain inputs," which could also be done with black-box testing or white-box testing.[12] (*Id*. at 99). With regard to fault injection, Dr. Kelly testified that "injecting faults into the code is changing the code," and that he was unaware whether Ford injects fault into its source code during its testing process. (*Id*. at 101-02). Dr. Kelly also stated that, in his experience, deliberately modifying source code so that it fails in order to determine how the code operates is not done as part of the source code development and testing process. (*Id*. at 102). In addition, Dr. Kelly explained that he did not believe that there was any benefit to be gained from introducing errors into the code. (*Id*. at 103). He added that faults could be injected "into the as-built system" by "set[ting] various ends to certain values" or corrupting certain memory locations. (*Id*. at 106).

Before the hearing concluded, the parties discussed how the source code would be viewed by Plaintiffs' experts. Plaintiffs' counsel suggested that a physically secure room containing Ford's server be created by Ford wherever it likes and that a virtual private network ("VPN") be established between Ford's server and another location near Plaintiffs' experts. (ECF No. 531 at 112). The location near Plaintiffs' experts would contain computers for Plaintiffs' experts' use that would access Ford's source code and

[12] White-box testing is "[t]esting software with the knowledge of the internal structure and coding inside the program." PC Magazine, http://www.pcmag.com/encyclopedia/term/54432/white-box-testing (last visited June 10, 2015).

any tools used to analyze the source code over the VPN. (*Id*. at 113). To access the computers at the Plaintiffs' experts' location, Plaintiffs' counsel suggested that the computers be equipped with biometrics scanners (e.g. fingerprint or retina scanners). (*Id*.) Plaintiffs' counsel averred that any increased risk that Ford's source code might be intercepted by a hacker who accesses the VPN was "not severe enough" to justify the cost and additional time that would be necessary if all of the source code review were to take place at a facility designated by Ford. (*Id*. at 114). In response, Ford's counsel objected to Plaintiffs' counsel's VPN proposal and asserted that Ford's source code has never been placed anywhere outside of a secured location at a Ford facility and that the code did not "get carried around or put up on other networks." (*Id*. at 115, 119); *see also* (ECF No. 502 at 2) (Ford's brief on the issue stating that Ford has never produced its ETC source code in the manner proposed by Plaintiffs). Yet, Ford's counsel acknowledged that pieces of the source code during the design stage might be transported out of Ford's facility. (ECF No. 531 at 119-20). After hearing the parties' positions, the Court made the parties aware that it shared Ford's concerns as to Plaintiffs' counsel's VPN proposal.

## II. Positions of the Parties

In its brief filed before the hearing, Ford contends that Plaintiffs should only be permitted to inspect and search a read-only version of the ETC source code, and that during Plaintiffs' review of the source code, they should not be allowed to use "additional tools or applications," including compilers and simulators.[13] (ECF No. 502 at 1, 10). Ford argues that Plaintiffs desire the production of the ETC source code in write-

[13] At the hearing, Ford's counsel indicated that he would have to speak with his client as to whether it objects to Plaintiffs' experts' use of a compiler or debugger on the ETC source code in read-only format. (ECF No. 531 at 109).

access format so that they can manipulate it, which Ford believes contradicts the Court's prior ruling that ordered the production of the ETC source code *for review*. (*Id*. at 7). According to Ford's brief, source code may be manipulated "without leaving physical evidence or a trail of changes." (*Id*. at 8). Ford argues that source code altered by Plaintiffs' experts "would not be representative of the ETC system in vehicles sold by Ford," because the binary image flashed into the PCM ROM (read only memory) cannot be changed without a special tool to erase the ROM and flash a different binary image into the ROM. (*Id*.) In addition, Ford insists that providing write access to the source code increases a wrongdoer's ability to create a counterfeit Ford ETC system binary image. (*Id*. at 8-9). Furthermore, Ford asserts that the ETC source code has never been provided to any non-Ford personnel, including suppliers, for anything other than "eyes on" review on "an as needed basis at a Ford facility on a Ford supplied computer with a Ford employee present and operating the computer." (*Id*. at 9).

In their brief submitted before the hearing, Plaintiffs contend that they must have write access to the source code in order to perform an adequate and efficient analysis of the code. Plaintiffs claim that they must be able to test the source code by injecting faults "into Ford's system to see how it handles or addresses those faults." (ECF No. 504 at 2). They assert that "[t]his is a common and critical testing mechanism." (*Id*.) Plaintiffs also maintain that they must be able to analyze the "interactions between code sections," and that this cannot occur just by reading the code. (*Id*. at 4). In addition, Plaintiffs insist that any changes that they make to the source code during their testing of the code can be revealed by using a file comparison tool. (*Id*.) Finally, Plaintiffs argue that they must be permitted to use a compiler and any related tools in order to "gain a full understanding of how the system actually communicates," and that if they do not

have access to a compiler, then the source code review process will be protracted and more expensive. (*Id*. at 7-8).

After the hearing, Plaintiffs submitted to the Court a spreadsheet containing the names and descriptions of approximately fifty software tools that they hope to use in analyzing the ETC source code. Plaintiffs also indicated whether those tools were compatible with a read-only version of the source code. Plaintiffs' tool list includes virtual machines for operating systems; compilers, linkers, assemblers, simulators, debuggers, integrated development environments, static analysis tools, scripts, and batch files used by Ford; custom tool development programs; script development software; searching and matching programs; word processing programs; Adobe Acrobat; file compression programs; an encryption program; Linux editors; Windows editors; version control software; static analysis tools other than those used by Ford; file comparison software; a metrics tool; and source code "formatters and beautifiers." According to Plaintiffs, a majority of these programs can be used with a read-only version of source code, including the simulators and debuggers used by Ford. In contrast, Plaintiffs assert that the compilers used by Ford "probably" require source code in a write-access format. Nearly all of the tools listed by Plaintiffs would require write access permission for the ***file system***. In other words, those tools would require the ability to create files or data on the computer's hard drive or on the computer's network drive, but would not require the ability to change or edit the source code itself.

In response, Ford's counsel sent a letter informing the Court that Ford objected to Plaintiffs' experts' use of any compiling tools on Ford's ETC source code. In the event that the Court allows Plaintiffs' experts to utilize a compiler, Ford has requested that Plaintiffs' experts' compiler be similar to that which Ford uses (the Green Hills Software

Integrated Development Environment). Ford also objects to a multitude of Plaintiffs' proposed tools on the grounds that they pose "a significant safety or security threat to Ford," they require the source code in write-access format in order to run, they "implicate[] the production by Ford of source code for systems far beyond the ETC system," or they are duplicative of other tools that Ford has agreed Plaintiffs may use. Ford disagrees with Plaintiffs that many of their listed tools may be used with source code in a read-only format. Importantly, Ford asserts that its simulators require source code to be in write-access format; however, Ford's compilers, assemblers, linkers, and debuggers do not require write-access format. Ultimately, Ford has agreed to the use of six tools or types of tools proposed by Plaintiffs, including the static analysis tools used by Ford, Windows PowerShell version 5 (a script developing tool), Veracrypt (an encryption tool), PC-Lint (a static analysis tool), RSM (a metrics tool), and Beyond Compare (a file comparison tool).

## III. Relevant Law

Federal Rule of Civil Procedure 26(b)(1) provides that:

> Parties may obtain discovery regarding any matter, not privileged, that is relevant to the claim or defense of any party, including the existence, description, nature, custody, condition, and location of any books, documents, or other tangible things and the identity and location of persons having knowledge of any discoverable matter ... Relevant information need not be admissible at the trial if the discovery appears reasonably calculated to lead to the discovery of admissible evidence.

While the claims and defenses raised in the case should be the focus of discovery, broader discovery is permitted when justified by the particular needs of the case. Fed. R. Civ. P. 26(b)(1), advisory committee notes (2000). In general, information is relevant, and thus discoverable, if it "'bears on, or ... reasonably could lead to other matter[s] that could bear on, any issue that is or may be in the case. Although 'the pleadings are the

starting point from which relevancy and discovery are determined ... [r]elevancy is not limited by the exact issues identified in the pleadings, the merits of the case, or the admissibility of discovered information.'" *Kidwiler v. Progressive Paloverde Ins. Co.*, 192 F.R.D. 193, 199 (N.D.W.Va. 2000) (internal citations omitted). In many cases, "the general subject matter of the litigation governs the scope of relevant information for discovery purposes." *Id*. The party resisting discovery, not the party seeking discovery, bears the burden of persuasion. *See Kinetic Concepts, Inc. v. ConvaTec Inc.*, 268 F.R.D. 226, 243-44 (M.D.N.C. 2010)(citing *Wagner v. St. Paul Fire & Marine Ins. Co.*, 238 F.R.D. 418, 424-25 (N.D.W.Va. 2006)).

Simply because information is discoverable under Rule 26, however, "does not mean that discovery must be had." *Schaaf v. SmithKline Beecham Corp.*, 233 F.R.D. 451, 453 (E.D.N.C. 2005) (citing *Nicholas v. Wyndham Int'l, Inc.*, 373 F.3d 537, 543 (4th Cir. 2004)). For good cause shown under Rule 26(c), the court may restrict or prohibit discovery when necessary to protect a person or party from annoyance, embarrassment, oppression, or undue burden or expense. Fed. R. Civ. P. 26(c). In addition, Rule 26(b)(2)(C) requires the court, on motion or on its own, to limit the frequency and extent of discovery, when (1) "the discovery sought is unreasonably cumulative or duplicative;" (2) the discovery "can be obtained from some other source that is more convenient, less burdensome, or less expensive;" (3) "the party seeking the discovery has already had ample opportunity to collect the requested information by discovery in the action;" or (4) "the burden or expense of the proposed discovery outweighs its likely benefit, considering the needs of the case, the amount in controversy, the parties' resources, the importance of the issues at stake in the action, and the importance of the discovery in resolving the issues." Fed. R. Civ. P.

26(b)(2)(C)(i)-(iii). This rule "'cautions that all permissible discovery must be measured against the yardstick of proportionality.'" *Lynn v. Monarch Recovery Mgmt., Inc.*, 285 F.R.D. 350, 355 (D.Md. 2012) (quoting *Victor Stanley, Inc. v. Creative Pipe, Inc.*, 269 F.R.D. 497, 523 (D.Md. 2010)). To insure that discovery is sufficient, yet reasonable, district courts have "substantial latitude to fashion protective orders." *Seattle Times Co. v. Rhinehart*, 467 U.S. 20, 36, 104 S.Ct. 2199, 81 L.Ed.2d 17 (1984).

Federal Rule of Civil Procedure 26(c)(1)(G) allows the court, for good cause, to issue an order "requiring that a trade secret or other confidential research, development, or commercial information not be revealed or be revealed only in a specified way." In order for the court to apply the rule, two criteria must exist. First, the material sought to be protected must be "a trade secret or other confidential research, development, or commercial information." Second, there must be a "good cause" basis for granting the restriction. The party seeking protection bears the burden of establishing both the confidentiality of the material and the harm associated with its disclosure. *Deford v. Schmid Prods. Co.*, 120 F.R.D. 648, 653 (D.Md. 1987) (citing *Cipollone v. Liggett Group, Inc.*, 785 F.2d 1108, 1121 (3d Cir. 1986)). Once these elements are demonstrated, the burden shifts to the party seeking disclosure to show that the material is relevant and necessary to its case. *Empire of Carolina, Inc. v. Mackle,* 108 F.R.D. 323, 326 (D.C. Fla. 1985). The court "must balance the requesting party's need for information against the injury that might result if uncontrolled disclosure is compelled." *Pansy v. Borough of Stroudsburg,* 23 F.3d 772, 787 (3d Cir. 1994) (quoting Arthur R. Miller, *Confidentiality, Protective Orders, and Public Access to the Courts,* 105 Harv. L. Rev. 427, 432-33 (1991)).

If the court determines that disclosure is required, the issue becomes whether the materials should be "revealed only in a specified way." Fed. R. Civ. P. 26(c)(1)(G). "Whether this disclosure will be limited depends on a judicial balancing of the harm to the party seeking protection (or third persons) and the importance of disclosure to the public." *Id*. Factors to consider when deciding if and how to limit disclosure include:

> (1) whether disclosure will violate any privacy interests; (2) whether the information is being sought for a legitimate purpose or for an improper purpose; (3) whether disclosure of the information will cause a party embarrassment; (4) whether confidentiality is being sought over information important to public health and safety; (5) whether the sharing of information among litigants will promote fairness and efficiency; (6) whether a party benefiting from the order of confidentiality is a public entity or official; and (7) whether the case involves issues important to the public.

*Pansy*, 23 F.3d at 787-91. The court exercises broad discretion in deciding "'when a protective order is appropriate and what degree of protection is required.'" *Furlow v. United States*, 55 F. Supp. 2d 360, 366 (D.Md. 1999) (quoting *Seattle Times Co.*, 467 U.S. at 36).

## IV. Discussion

As discussed above, the parties disagree over the format in which the ETC source code must be produced. Beyond that, the parties also differ on what software tools Plaintiffs should be permitted to use in viewing and analyzing the ETC source code. Having considered the parties' submissions and the testimony adduced at the hearing, the Court finds that Plaintiffs are entitled to a read-only version of the ETC source code because the benefits that Plaintiffs seek from gaining access to the source code can be realized through a read-only format. As Dr. Kelly testified, most of what Mr. Jones demonstrated at the hearing can be performed on source code in a read-only format. (ECF No. 531 at 99). The only possible limitations of a read-only format are that the

source code cannot be changed and faults cannot be injected directly into the source code by editing the code. (*Id*. at 89). However, as Ford and Dr. Kelly have pointed out, changing the source code would make the resulting code unrepresentative of the code contained in Plaintiffs' vehicles. (*Id*. at 32-33, 104). Moreover, Dr. Kelly asserted that fault injection can be performed in a way that does not implicate editing or adding lines to the source code. (*Id*. at 106). Under a proportionality analysis, at this juncture, any additional benefit of providing Plaintiffs with the code in a write-access format does not outweigh Ford's concerns about the security of its highly proprietary ETC source code.[14] Furthermore, it would be premature, and possibly inefficient, to grant Plaintiffs write access to the ETC source code when they have yet to even review or analyze the code in a read-only format. Accordingly, the Court **ORDERS** that Ford must produce the ETC source code in a read-only format.

As for the specific programs that Plaintiffs' experts may use, Ford informed the Court that its compiler and debugger could be used on read-only code, but objected to allowing Plaintiffs to use a compiler. The Court finds that Plaintiffs should be permitted to use a compiler in analyzing the source code as the compiler may alert Plaintiffs to certain errors in the code. Moreover, the debugger requires compiled code. (ECF No. 531 at 49, 55-56, 86, 100). The Court agrees with both parties that Plaintiffs should be

[14] Under the protective order related to source code in the Toyota unintended acceleration litigation, the plaintiffs' expert, Michael Barr, was able to use the Green Hills Compiler and Simulator as well as a number of static analysis tools. Plaintiff's Opposition to Toyota's *Daubert* Motion to Exclude the Expert Testimony of Michael Barr at 25, *In Re: Toyota Motor Corp. Unintended Acceleration Marketing, Sales Practices, & Products Liability Litigation*, No. 8:10-cv-01460-JVS-FMO (C.D. Cal. Sept. 9, 2013), ECF No. 100. It is unclear whether the plaintiffs were given write access to Toyota's source code. Many of the provisions related to analysis of the source code have been redacted from the version of the protective order available to this Court. *See* Redacted Stipulated Protective Order Governing the Exchange and Handling of Source Code and Source Code Related Material at 13-16, *In Re: Toyota Motor Corp. Unintended Acceleration Marketing, Sales Practices, & Products Liability Litigation*, No. 8:10-ML-02151-JVS-FMO (C.D. Cal. Mar. 31, 2011), ECF No. 1344.

required to use a compiler that is similar to the compiler that Ford used in developing the ETC source code, so that there is no discrepancy as to how the code was compiled when it was analyzed by Plaintiffs' experts.

Ford offers multiple arguments against the remainder of Plaintiffs' proposed tools. First, Ford argues that certain programs may allow modification of the ETC source code in RAM (random access memory) without being able to trace any edits, or permit Plaintiffs' experts to create additional tools for source code review without Ford's knowledge. However, Plaintiffs and their experts are instructed by this opinion that they are not to change or edit the ETC source code in *any way* when performing their analysis of the code. Furthermore, Plaintiffs shall be required to disclose all of the software tools that they plan to use in analyzing the source code **ten business days before they plan to use each tool**. Plaintiffs' obligation will be continuous so that any new tools may not be used to analyze the source code without first notifying Ford and providing Ford the opportunity to voice any concerns to Plaintiffs and lodge an objection to the use of the tool with this Court within ***five days of Plaintiffs' proposed use of the tool***. If Ford objects to the use of the tool, then the tool will not be used until the parties reach a resolution on the issue or the Court resolves the issue.[15] With these parameters in mind, the parties are advised that Plaintiffs will likely be permitted to use *standard* read-only compatible tools on the ETC source code so long as repeated testing does not unnecessarily delay the discovery process.

In addition, Ford contends that some of Plaintiffs' proposed tools will require the production of source code beyond the ETC source code. The Court need not address that

[15] This is similar to the tool provision in the Toyota unintended acceleration litigation. Redacted Stipulated Protective Order Governing the Exchange and Handling of Source Code and Source Code Related Material at 14, *In Re: Toyota Motor Corp. Unintended Acceleration Marketing, Sales Practices, & Products Liability Litigation*, No. 8:10-ML-02151-JVS-FMO (C.D. Cal. Mar. 31, 2011), ECF No. 1344.

concern at this time given that Plaintiffs have formally requested the ETC source code only; therefore, they have been granted access to the ETC source code only. If after reviewing the ETC source code, Plaintiffs believe that other portions of source code are indispensable to completing a proper and efficient analysis of the ETC source code as it is relevant to the claims and defenses in this litigation, then Plaintiffs may return to the Court and demonstrate good cause for the production of additional portions of source code.

Ford also objects to the use of certain proposed tools on the grounds that they are duplicative of other proposed tools, or that Ford does not use those tools in developing its source code. For example, Ford suggests that Plaintiffs' experts should not be permitted to use multiple static analysis tools. Yet, the NASA Engineering and Safety Center recognized the value of utilizing different static analysis tools in conducting its review of Toyota's source code:

> The team's experience is that there is no single analysis technique today that can reliably intercept all vulnerabilities, but that it is strongly recommended to deploy a range of different leading tools. Each tool used can excel at a different aspect of static analysis, which results in remarkably little overlap in the set of warnings that is produced. The combination of different analyzers achieves the highest value in analysis results.

NASA Engineering and Safety Center, National Highway Traffic Safety Administration Toyota Unintended Acceleration Investigation – Appendix A, 9 (2011), *available at* www.nhtsa.gov/staticfiles/nvs/pdf/NASA_FR_Appendix_A_Software.pdf.[16] When the potentially differing strengths and weaknesses of the tools are taken into account, the tools about which Ford complains may not be duplicative at all. Moreover, there is no

[16] Plaintiffs' proposed tool list includes some of the tools used by NASA's engineers in testing Toyota's source code, such as Coverity (a static analysis tool) and Simulink (a program used to design, simulate, and test systems). National Highway Traffic Safety Administration Toyota Unintended Acceleration Investigation – Appendix A at 10, 12.

evidence that deploying similar tools to analyze the ETC source code will unreasonably delay the analysis process. Furthermore, there is some value in allowing Plaintiffs' experts to use certain analytical tools that Ford does not typically use in the development process. *See id*. at 26 (stating that NASA engineers "intentionally applied analysis tools" not used by Toyota).

In sum, Ford's objections to the read-only compatible tools proposed by Plaintiffs are unconvincing. The Court **ORDERS** that Plaintiffs' experts are entitled to use those programs listed in their spreadsheet that are compatible with a read-only format so long as the use of multiple tools does not unnecessarily delay the discovery process. For any additional tools that Plaintiffs' experts wish to use, Plaintiffs must follow the process described above.

Having concluded that Plaintiffs are entitled to the ETC source code in a read-only format and that Plaintiffs may use the listed tools that are compatible with such a format, the Court finds that Plaintiffs should start their analysis of the ETC source code with the vehicle models that they have already tested. As noted above, Plaintiffs' counsel has represented that testing was performed on two models and that Plaintiffs' experts were able to recreate sudden acceleration events during that testing. (ECF No. 531 at 72, 75). Plaintiffs' experts should begin their source code analysis on the source code used in those two models (the 2005 Explorer and the 2005 Mustang). (*Id*. at 76). After an initial review of the ETC source code for those models, Plaintiffs shall report back to the Court with an estimate of the time that it will take to complete the ETC source code analysis for those two models. Plaintiffs shall also provide the Court with an estimate of time that it will take to analyze the ETC source code for the additional models at issue in this litigation, as well as a proposed schedule setting forth in order of priority the vehicle

models and years from which the ETC source code shall be analyzed.

Finally, the parties are **ORDERED** to meet and confer regarding the remaining terms of the special protective order and shall tender an Agreed Order Governing Disclosure of Ford's Source Code to the undersigned no later than June 22, 2015. As previously stated, if the parties cannot agree on the terms not resolved by this opinion, they shall submit their respective versions to the undersigned by electronic mail no later than 12:00 noon on June 22, 2015 and the disputed issues shall be addressed at the regularly scheduled telephone conference on June 24, 2015. The Court urges both parties to review the redacted source code protective order filed in the Toyota sudden unintended acceleration litigation[17] to ascertain its applicability in the instant matter. As the Court stated at the hearing on this issue, the Court is not inclined to adopt Plaintiffs' VPN proposal. The source code review should occur in a secure room in a secure facility where access to the room may be strictly controlled.[18] Additionally, the Court suggests that any person who reviews the source code be required to sign a confidentiality agreement that includes a penalty clause for failure to comply with the terms of the agreement. Furthermore, no copies of the source code or any portion thereof (including electronic copies) shall be made, unless the parties otherwise agree, and no outside electronic devices shall be permitted in the secure room. To the extent that the parties have not already reached agreement on the following issues, the parties should also consider: the number of computers that will be placed in the secure room; what specifications those computers will have; how Plaintiffs' experts' tools (software) will be

[17] ECF No. 1344 in Case No. 8:10-ML-02151-JVS-FMO (C.D. Cal. Mar. 31, 2011).

[18] By electronic mail sent on June 11, 2015, Ford suggests that the review occur at "a secure location at a Ford Building with Ford IT infrastructure" to offer the best protection, and access to IT support if necessary. Ford indicates that it is in the process of identifying a location that will accommodate Plaintiffs' desire to analyze the source code after regular business hours.

loaded onto the computers; how access to the secure room will be monitored (e.g. by a security guard, with cameras, etc.); if the inside of the secure room will be monitored; whether a neutral system administrator will be required; the days and hours that Plaintiffs' authorized experts or representatives may access the secure room; how Plaintiffs will notify Ford of the identity of those who will be accessing the secure room; whether note taking will be allowed, and if so, how those notes will be stored; the process for returning the notes to Ford at the conclusion of the litigation; where expert reports will be written; and if there will be a limitation on citing to the source code in any documents filed with the Court.

The Clerk is directed to provide a copy of this Order to counsel of record and any unrepresented party.

**ENTERED:** June 12, 2015

Cheryl A. Eifert
United States Magistrate Judge