

Exhibit 3

UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF FLORIDA

CASE NO. 11-20427-JORDAN

DISNEY ENTERPRISES, INC.,
TWENTIETH CENTURY FOX FILM CORPORATION,
UNIVERSAL CITY STUDIOS PRODUCTIONS LLLP,
COLUMBIA PICTURES INDUSTRIES, INC., and
WARNER BROS. ENTERTAINMENT INC.,

Plaintiffs,

v.

HOTFILE CORP., ANTON TITOV, and
DOES 1-10,

Defendants.

**DECLARATION OF J. ALEX HALDERMAN IN SUPPORT OF DEFENDANTS'
OPPOSITION TO PLAINTIFFS' MOTION TO COMPEL RESPONSES TO
REQUESTS FOR PRODUCTION OF DOCUMENTS AND INTERROGATORIES**

I, J. ALEX HALDERMAN, declare as follows:

1. I am an assistant professor of electrical engineering and computer science at the University of Michigan, where I have been a faculty member since January 2009. In addition to my professorship, since 2009 I have held a research collaborator appointment at Princeton University's Center for Information Technology Policy. I received my Ph.D., M.A., and A.B. degrees in computer science from Princeton. My curriculum vitae, attached as Exhibit A, contains a list of my publications and a detailed account of my professional experience. I offer this Declaration in support of Defendants' opposition to Plaintiffs' Motion to Compel Responses to Requests for Production of Documents and Interrogatories.

2. This declaration has three parts. First, I discuss alternatives to source code analysis that could be used to resolve disputed issues about the operation of the Hotfile system. Second, I describe limitations of source code analysis that limit its relevance to issues raised by the Plaintiffs. Third, I offer my perspective on the security risks of source code production.

3. I have reviewed Plaintiffs' Motion to Compel Responses to Requests for Production of Documents and Interrogatories as well as Dr. Ian Foster's declaration in support of that motion. The observations and conclusions set forth below are based on my specialized knowledge, education, and expertise as applied to the facts and circumstances in this case.

Alternatives to Source Code Analysis

4. A computer program's source code functions like a detailed engineering blueprint. It conveys the software engineers' instructions to other computer programs, which interpret the source code to construct a software application or website.

5. A web-based service such as HotFile typically uses various kinds of source code. These include HTML, the code that conveys the text and structure of a web page to the user's web browser. Any user of the service can view its HTML source code by simply right-clicking in the web browser. In contrast, other kinds of source code specify the behavior of software that runs on the web server, and this source code is not typically accessible to users.

6. I understand that the Plaintiffs in this Action are requesting discovery of the complete source code for the Hotfile service. While analysis of the source code blueprints is sometimes the best tool for investigating low-level details of software's operation—as in some patent litigation—it is only one of several methods of inquiry that may be used to understand a website's design (such as in this Action). There are several other methodologies that would

resolve the disputed questions as well as or better than source code review. These include black-box testing, data analysis, and—at most—deposition of Hotfile developers.

7. Black-box testing is an established methodology in computer security and software testing. Engineers use it to probe the functionality and inner workings of a program *without* access to the source code. Black-box testing typically involves using the software and exercising the available functions while observing its behavior in a systematic way. By noting how the program behaves under a variety of inputs and operating conditions, engineers can form and test hypotheses about the software's internal workings. Black-box testing can be applied to a website from any Internet-connected location.

8. In this Action, the Plaintiffs and Dr. Foster appear to have applied a form of black-box testing in preparation for their filings. I understand that they have experimented with Hotfile by interacting with it using one or more user accounts over a period of several months. Through such testing, the Plaintiffs have such technical understanding of the Hotfile system as to offer detailed opinions about Hotfile's website.

9. Considering the asserted bases for Plaintiffs' demand for Hotfile's source code brings to mind the analogous situation of a lawsuit concerning the design of an automobile where the plaintiffs sought engineering blueprints to establish whether the transmission was manual or automatic. This could be ascertained in several other ways, such as by driving the car, by reviewing the purchase documents, or by asking the engineers. Similarly, technical questions about Hotfile can be resolved without access to source code through further black-box testing, analysis of Hotfile data, and questioning of Hotfile developers at deposition.

10. To give one example, Dr. Foster's declaration discusses a Hotfile feature that allows users to create multiple links referring to the same file. He states that he is uncertain

whether all the links are disabled in response to an infringement complaint concerning the file. There are several ways that the Plaintiffs could resolve this uncertainty without source code. For instance, with permission from Hotfile, they could conduct a test by having an agent (unknown to the Defendants) pose as a user, upload a file, and request multiple links. They could then have another agent send Hotfile a copyright complaint concerning one of the links and check whether the other links continued to be accessible.

11. A second way that the Plaintiffs could resolve this uncertainty is by analyzing data from the Hotfile system. For the sake of argument, I assume that Hotfile records instances where the multiple links feature is used, instances where these links are used to download a file, and instances where the links are taken down in response to infringement complaints. If Hotfile does not disable the multiple links in response takedown requests, I would expect these data to show many occasions when one link was disabled and then another was subsequently used to retrieve the file.

12. A third way that the Plaintiffs could resolve such questions is by taking testimony from Hotfile developers. If there is actual reason to doubt the veracity of their testimony, it can be corroborated or refuted by the other modes of inquiry I have discussed.

13. I understand that Plaintiffs' allegations of copyright infringement concern the overall design of the Hotfile website, the features it provides or fails to provide, and the intentions of its developers. In other software I have examined of similar complexity to the Hotfile system, the overwhelming majority of the source code is dedicated to technical minutiae of implementation that would have no relevance to such questions. Large portions of the Hotfile source code likely deal with mundane tasks such as generating the web pages shown to users, coordinating tasks among multiple servers, and responding to errors. To continue the automobile

analogy, the blueprints for the car's electrical system would be irrelevant in determining whether the transmission was automatic or manual, as would the precise metallurgical specifications, machining tolerances, and gearing ratios found in the blueprints to the transmission itself.

Limitations of Source Code Analysis

14. Source code analysis provides a particular kind of view into the design of a computer program, and it has important limitations. Since source code acts as a blueprint, it is most useful for understanding the low-level (*i.e.*, detailed) mechanics of the program's operation. Line-by-line analysis of source code is often relevant to patent cases, since a patent claims may be infringed by combinations of instructions that are dispersed through the program. However, while source code documents precisely *how* a program is implemented, it typically provides little or no information about why the designers decided to incorporate certain features or about their goals and intentions. Furthermore, source code, like blueprints, provides only a snapshot of the program's design; it is not a record of how the software has been used in operation.

15. Due to these limitations, source code is not relevant to many of the technical questions that the Plaintiffs in this Action have raised. For instance, I understand that Plaintiffs argue that source code access would help them establish that Defendants could have identified and terminated repeat infringers but failed to do so. The Hotfile source code is neither necessary nor relevant to this question. Source code specifies the mechanical function of a computer program, not whether the program's operators used or failed to use any particular feature in the past. If that information is recorded, it appears in logs and other data files, not the source code.

16. I understand that the Plaintiffs further argue that source code access will help them establish whether "there are readily available technological steps that defendants could have implemented" to remove infringing files. Counterfactual questions about how a website

could have been designed typically do not turn on the actual precise workings of the system described in its source code. They are more likely to depend on the technical feasibility of the supposed changes—that is, whether they are possible in general—which is (of course) independent of the current implementation.

17. A program's source code specifies precisely how it operates, but source code typically does not describe why a program was designed to provide particular features. While engineers sometimes explain technical decisions in annotations within the source code known as "comments", these are typically intended to assist other engineers in extending and maintaining the software and concern low-level engineering considerations rather than the program's overall design goals. The designers' goals and intentions, if they are written down, are typically recorded in design documents that are separate from the source code. On questions of intent, analysis of the Hotfile source code is more likely to lead to opinion or conjecture than facts.

Security Risks of Source Code Production

18. I understand that the Plaintiffs and Defendants have agreed to a Stipulated Protective Order that contains provisions that are intended to safeguard any source code produced in the course of this litigation. However, even with these protections in place, production of the complete Hotfile source code would create substantial security risks for Hotfile and its users.

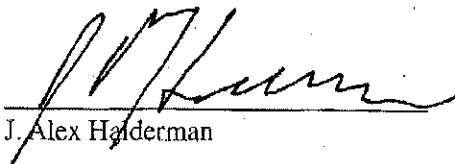
19. Popular web-based services such as Hotfile are products of rapid software development, which typically leads to the existence of security vulnerabilities due to errors or oversights by the programmers. While these vulnerabilities may not be apparent to users of the websites, they can typically be found and exploited rapidly with access to the source code. If the Hotfile source code were accidentally leaked, it would likely allow attackers to penetrate the

Hotfile system, potentially exposing users to theft of private data, compromising the forensic integrity of data collected by the system, and necessitating costly and time-consuming repairs that might force the service offline for an extended period. Parties that receive the Hotfile source code as part of this Action could be targeted by hackers seeking to use it to attack Hotfile. There is a substantial risk that a targeted hack would succeed, even if the receiving parties meticulously adhere to the provisions of the Stipulated Protection Order.

20. Such an attack would have several historical precedents. In 2009, agents of the Chinese government compromised Google's software development systems in an apparent effort to gain access to source code for Google websites and use it to launch further attacks. More recently, in April 2011, anonymous hackers penetrated servers used by Sony to operate the PlayStation Network gaming service. Reports indicated that these hackers stole data containing personally identifiable information for all of the system's 77 million users and forced the system offline for repairs for more than three weeks. Hotfile and its users could be exposed to similar damage in a successful attack.

I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

Executed on this 17th day of June 2011, at Ann Arbor, MI.


J. Alex Halderman