GEORGETOWN UNIVERSITY

S$^2$ERC
SECURITY AND SOFTWARE
ENGINEERING RESEARCH CENTER

S$^2$ERC Project: Vendor Truth Serum
Authors: Dr. Gregory Klass and Dr. Eric Burger
Date: 22 September 2016

## Abstract

This report, a result of the *Vendor Truth Serum* project in the S$^2$ERC, examines approaches for enabling the sharing of software engineering test tool performance and coverage benchmark results in general and for the SWAMP in particular.

This material is based upon work supported by the National Science Foundation under Grant No. 1362046 and the industry affiliates of the Security and Software Engineering Research Center (S$^2$ERC). The views and analysis provided are entirely our own and not attributable to any other party. Support for this work includes funding from the Department of Homeland Security Science and Technology Directorate. Payments are made to Georgetown University and the funds are used to cover the expenses of the study and related academic and research activities of the institution.

## The SWAMP Project

The Software Assurance Market Place, or SWAMP, is an environment where developers can, for free, test their software code for security flaws. The SWAMP receives support from the Department of Homeland Security Science & Technology Directorate.[1] As well, the SWAMP is a venue where tool developers can test their tools against code samples with known flaws as well as allow tool vendors to make their services available to the development community.[2]

Software engineering tools are an important technology for building secure software and cyber-physical systems. There is a class of tools targeting software security, stability, and correctness. Experience is that any single tool covers a relatively small percentage of potential errors. A mix of tools should increase coverage. One needs to know what the tools cover in order to make a rational decision of which tools a software engineering project needs. Likewise, one needs to know where the vulnerabilities are that no tools can check. For example, not one tool on the market detected the Heartbleed OpenSSL bug.[3] As such, industry and government software developers need to test their code against multiple test tools.

With many tools to choose from, a real issue is deciding which tools a given developer needs to use to ensure satisfactory test coverage over their software artifact. What is needed is a way for developers to know which tools provide what coverage, so they can make informed choices and accomplish satisfactory testing in minimal time at minimal expense. Unfortunately, there are common industry contractual practices which inhibits making such knowledge generally available.

## DeWitt Clauses

### (a) The Standard DeWitt Clause

Many software vendors include so-called DeWitt clauses in their licenses. A DeWitt clause specifies that the licensee may not publish or otherwise share the results of its testing, benchmarking or other evaluations of the licensed software. Such clauses are named after David DeWitt. In the early 80s, as an assistant professor of computer science at the University of Wisconsin Madison, DeWitt performed and published benchmarking tests on Oracle software. Displeased with the results, Oracle subsequently included in its end user license agreement (EULA) a clause prohibiting such acts. The clauses are now common in the software industry.

[1] See https://www.dhs.gov/science-and-technology/swamp-key-resource-improving-software-assurance-activities

[2] See https://continuousassurance.org/about-us/backgrounder/

[3] Kupsch, J.A., & Miller, B.P., *Why Do Software Assurance Tools Have Problems Finding Bugs Like Heartbleed?*, Continuous Software Assurance Marketplace, 22 Apr. 2014, https://continuousassurance.org/swamp/SWAMP-Heartbleed.pdf

Examples of DeWitt from SWAMP licenses include the following:

> ***GrammaTech:*** Limitations on Software Use. You may not: . . . disclose Software output, including, but not limited to the results of any benchmark test of the Software, or Software documentation to any third party without GrammaTech's prior written approval.[4]
>
> ***ParaSoft:*** You shall not . . . perform, publish, or release to any third parties any benchmarks or other comparisons regarding the Software or User Documentation.[5]
>
> ***Red Lizard:*** You will not publish any findings regarding or resulting from use of the Software and you will not disclose to any third party any comparison of the results of operation of the Software with other services or products, without first obtaining the written consent of Red Lizard Software (unless expressly permitted by this Agreement).[6]

Although the wording of each example is somewhat different, all purport to prohibit the sharing of any benchmark studies. Of the three examples, ParaSoft appears to be the most restrictive, purporting to prohibit even the performance of benchmarking. Although GrammaTech's is the only clause that expressly permits benchmarking disclosure with the licensor's prior written approval, it is likely that any of the clauses could be modified by such written approval—whether expressly permitted or not.

### (b) The Microsoft Benchmarking Clause

The standard DeWitt clause found in most of the industry prohibits the disclosure of any benchmark testing. In July 2005, as part of the settlement agreement to an antitrust suit against it, Microsoft updated the license for its .NET Framework to replace the blanket prohibition with a more nuanced clause. The current .NET license reads as follows:

> You may disclose the results of any benchmark test of the .NET Component, provided that . . .:
>
> (1) you must disclose all the information necessary for replication of the tests, . . .;

[4] GrammaTech, Inc. CodeSonar EULA, version 2012.2.11.

[5] ParaSoft EULA, Rev. 20130910.

[6] Red Lizard Software EULA, licensed to the Software Assurance Marketplace ("SWAMP") operated by Morgridge Research Institute at University of Wisconsin.

> (2) you must disclose the date(s) that you conducted the benchmark tests, along with specific version information for all Microsoft software products tested . . .;
>
> (3) your benchmark testing was performed using all performance tuning and best practice guidance . . .;
>
> (4) it shall be sufficient if you make the disclosures provided for above at a publicly available location such as a Web site . . .[7]

Our research indicates that other software vendors have not followed Microsoft in adopting such clauses.

## The Costs of DeWitt Clauses

DeWitt clauses have their defenders. Vendors have argued that the clauses are necessary to protect both consumers and software producers from poorly executed benchmarking studies. But they also impose significant informational costs.[8] Nowhere is this truer than in the case of software assurance licenses.

There are broadly two sources of costs imposed by DeWitt clauses. The first is on the developers of software and the second is on society at large. Examining the first, a diligent software developer, not being certain which tools provide what coverage, will spend more time, effort, and money on different tools in order to have some level of assurance they can find most kinds of software bugs. This is wasteful and, more especially given the limited resources available to software developers, means the software developer will be underinvesting in useful features for their customers (impacting revenue) or underinvesting in providing more robust availability and integrity features (impacting software and system security). In reality, given the fact that no one tool provides total coverage, the more likely scenario for a rational software developer is to significantly underinvest in using software engineering tools to test for known flaw types, on the presumption that using multiple tools will not significantly improve test coverage and as such would be a waste of time and resources.

In a regime where no one know which tools cover what flaws may be present in source code, it would be hard to say that a software developer who does minimal if any testing would be negligent in their duty to provide software with a minimal level of integrity. This leads to the second source of costs imposed by DeWitt clauses. Namely, if a developer fails to achieve satisfactory test coverage over their software,

[7] Microsoft .NET Framework 1.1 Redistributable EULA, *available at:* https://msdn.microsoft.com/en-us/library/ms973265.aspx.

[8] See Genelle I Belmas & Brian N. Larson, *Clicking Away Your Speech Rights: The Enforceability of Gagwrap Licenses*, 12 Comm. L. & Pol'y 37, 40-45 (2007).

flaws will inevitably be discovered in the field. While such flaws may impact a software developer's reputation, they can be catastrophic to the users of that software. For example, a piece of software might be vulnerable to a stack overflow flaw. To the software developer, that may mean they have to generate a patch to fix the bug. To the enterprise using the software, their bank accounts may be cleaned out, customer personal information may be disclosed, proprietary intellectual property stolen, or worse.

Providing public benchmarking for software security integrity products would enable software developers to make rational choices in what tools to use. By being able to map the system vulnerabilities to the flaws that are known detected by various software engineering tools, developers can select the minimal set of tools that will provide sufficient coverage for the type of system, likelihood of attack or failure, and test resources available. Not only would this minimize the cost to the developer for properly testing their source code, but it also would provide a reasonableness benchmark for society at large to determine if the software developer is taking a due standard of care for producing secure and reliable products.

## Legal Framework

DeWitt clauses are contract terms like any other and are presumptively enforceable. That is, they are enforceable unless there exists some legal rule that renders them nonenforceable. Our research has found no authority for the existence of such a rule. This does not mean that a DeWitt clause could not be successfully challenged in court. There might be plausible arguments against a clause's enforcement in a specific case. But those arguments have not been tested in court.

### (a) Case Law

Our research found no case law that directly addresses the enforcement of DeWitt clauses generally.

Only one case has considered a challenge to a DeWitt clause. In *People v. Network Associates, Inc.*, 195 Misc. 2d 384, 758 N.Y.S.2d 466 (Sup. 2003), a New York trial court considered a challenge to Network Associates' license for several anti-virus programs. At issue was the following clause:

> Installing this software constitutes acceptance of the terms and conditions of the license agreement in the box. Please read the license agreement before installation. Other rules and regulations of installing the software are:
>
> A. The product cannot be rented, loaned, or leased—you are the sole owner of this product.
>
> B. The customer shall not disclose the result of any benchmark test to any third party without Network Associates' prior written approval.

> C. The customer will not publish reviews of this product without prior consent from Network Associates, Inc.

Network Associates had invoked the clause against a company that published a review of its software. That action prompted the Office of the New York State Attorney General (OAG) to investigate Network Associates' business practices. The OAG subsequently sued Network Associates, alleging that the clause violated New York State's Unfair and Deceptive Acts and Practices Statute, N.Y. G.B.L. § 349. Although the trial court held that the clause did violate the Act, its decision turned on the use of the words "rules and regulations," rather than the restriction on benchmarking generally. The OAG had argued that those words were misleading insofar as they suggested that the clause restated state or federal law, rather than being contractual in nature. Because the case's outcome turned on the wording of the clause, rather than its substance, it is of little import for DeWitt clauses more generally.

The use of DeWitt clauses was also an issue in *United States v. Microsoft*, the government's antitrust case against Microsoft Corporation. As part of its compliance agreement with the plaintiffs in the case, Microsoft agreed to alter the DeWitt clause that it was using in licenses for its .NET products.[9] Microsoft's current benchmarking clause is reprinted in Section 2(b) above. The plaintiffs in the case did not argue, however, that the DeWitt clause was unenforceable, but focused instead on its anticompetitive effects within the context of their broader claims about Microsoft's exercise of monopoly powers.

### (b) Statutes

There is also almost no statutory law that addresses DeWitt clauses.

The final version of the Uniform Computer Information Transactions Act (UCITA), developed by the National Conference of Commissioners on Uniform State Laws between 1999 and 2002, would have prevented the enforcement of DeWitt clauses. Section 105(c) provides:

> [Lawful public comment not prohibited.] In a transaction in which a copy of computer information in its final form is made generally available, a term of a contract is unenforceable to the extent that the term prohibits an end-user licensee from engaging in otherwise lawful public discussion relating to the computer information. However, this subsection does not preclude enforcement of a term that establishes or enforces rights under trade secret, trademark, defamation, commercial disparagement, or other laws.

[9] See Joint Status Report on Microsoft's Compliance with the Final Judgments, Civil Action No. 98-1232 (CKK) (Jan. 25, 2005), http://www.usdoj.gov/atr/cases/f207200/207283.htm, 20.

Only two states, however, Virginia[10] and Maryland,[11] enacted UCITA, and only Virginia's statute includes the above provision. At this point, enactment by additional states or Congress is highly unlikely.

Our research identified no other statutes or regulations that address the enforcement of DeWitt clauses.

### (c) General Legal Doctrines

Scholars writing in law reviews have suggested several legal grounds on which DeWitt clauses might be attacked: the copyright misuse doctrine, the copyright fair use doctrine, and the public policy doctrine.[12] None of these theories have been tested in court. And each faces the same significant challenge: the general enforceability of contract terms in software license agreements.

Although any or all of these arguments would be worth making in litigation, none provides a firm ground for saying at present that a standard DeWitt clause in a business-to-business contract is not enforceable.

## The Consumer Analog: Nondisparagement Clauses

DeWitt clauses commonly appear in software license agreements between businesses. Recently vendors have begun including a similar type of clause in contracts governing the sale of consumer goods or services. These so-called nondisparagement clauses commonly purport to prohibit the consumer from posting negative online reviews of the vendor's products or services.

An important example of such a clause can be found in Public Citizen's suit against Kleargear. Kleargear operated a website that sold novelty items. At the time of the lawsuit, Kleargear's online Terms of Sale and Use purported to forbid consumers "taking any action that negatively impacts KlearGear.com, its reputation, products, services, management or employees." After the spouse of a Kleargear consumer posted a negative review on RipoffReport, Kleargear demanded from the couple a $3,500 payment for violation of the nondisparagement clause. When the couple refused to pay, Kleargear reported to the credit reporting agencies that the couple had a $3,500 unpaid debt. Public Citizen, a consumer protection group, then sued Kleargear on the customer's behalf. Although in the end it was unclear whether the nondisparagement clause was even in Kleargear's Terms of Sale at the time of the transaction at issue, the case was important insofar as it focused public attention on the existence of nondisparagement clauses.

[10] Va. Code Ann. § 59.1-501.5.

[11] Md. Code Ann., Com. Law § 22-105.

[12] For the first two, see Anthony G. Read, Note: *DeWitt Clauses: Can We Protect Purchasers Without Hurting Microsoft?*, 25 Rev. Litig. 387,402-410 (2006). For the second, *see* Belmas & Larson, *supra* note 8, at 73-88.

One result of that attention was the California State Legislature passage of Assembly Bill No. 2335, approved by the California Governor on September 9, 2014. The California Office of Legislative Council has summarized the statute as follows:

> This bill would prohibit a contract or proposed contract for the sale or lease of consumer goods or services from including a provision waiving the consumer's right to make any statement regarding the seller or lessor or its employees or agents, or concerning the goods or services. The bill would make it unlawful to threaten or to seek to enforce, a provision made unlawful under the bill, or to otherwise penalize a consumer for making any statement protected under the bill. The bill would impose civil penalties upon any person who violates the provisions of the bill, of $2,500 for the initial violation and $5,000 for each subsequent violation, as well as an additional penalty of $10,000 if the violation was willful, intentional, or reckless. The bill would authorize the consumer, the Attorney General, or a district attorney or city attorney to bring a civil action for a violation of the provisions of the bill. The bill would provide that the penalty set forth in the bill is not an exclusive remedy, and does not affect any other relief or remedy provided by law. The bill would not prohibit or limit a person or business that hosts online consumer reviews or comments from removing a statement that is otherwise lawful to remove.

In short, consumer nondisparagement clauses in California are not only unenforceable, but the inclusion of such a clause can result in significant penalties.

Beginning in 2014, similar bills have been introduced in both the US House and Senate. The most recent Senate version, titled the Consumer Review Freedom Act of 2015,[13] would prohibit and void clauses in consumer contracts that attempt to prohibit, restrict or penalize public reviews of the product. The bill would also categorizes violations of the law as unfair and deceptive trade practices, rendering them liable for civil penalties of up to $10,000 for each violation. The bills in both chambers have had bipartisan support.

To be clear, these statutes aim at consumer contracts only. They do not, or would not, prohibit the enforcement of a DeWitt clause. They do, however, suggest a model for what legislative action on DeWitt clauses might look like.

## Possible Solutions

Given that there is no presently well-established legal basis for challenging the enforceability of a DeWitt clause, this leaves two options: working at the contract ne-

[13] S.2044, 114th Congress (2015-2016), *available at:* https://www.congress.gov/bill/114th-congress/senate-bill/2044.

gotiation stage to exclude such restrictions, and legislation addressing their enforceability.

### (a) Negotiated Terms

DeWitt clauses are contract terms like any other, which is to say that they are the terms to which the parties have agreed. If a license contains a DeWitt clause, the licensee's agreement to the term is the best argument for enforcement of it. By the same token, at the negotiation stage a licensee with bargaining power might insist that the DeWitt clause be removed or modified. And after a license has been entered into, a licensee might request that a DeWitt clause be modified.

The Microsoft .NET Framework license provides a good template for alternative terms a government licensee might request. As noted above, the Microsoft clause permits the disclosure of benchmarking studies so long as the licensee (a) uses best practices in performing the benchmarking studies and (b) discloses how it performed those studies. Such a clause strikes a balance between the vendor's legitimate interest in preventing poorly executed benchmarking and the government's interest in advancing cybersecurity.

Speculating a bit, the primary hurdle to this solution is likely to be the fact that software licenses are generally not negotiated. Instead, they are drafted by the vendor and offered to the licensee on a take-it-or-leave-it basis. They are what lawyers call "contracts of adhesion." A significant advantage of a contract of adhesion is that it saves on negotiation and drafting costs. It is expensive to get lawyers involved in every sale. Vendors might therefore be reluctant to negotiate or renegotiate terms with individual licensees.

### (b) Legislative Action

Congress could act to address the costs of DeWitt clauses. In fact, state and congressional action on nondisparagement clauses in consumer contracts means that some legislators are already familiar with the general types of problems that DeWitt clauses cause. Legislation might target DeWitt clauses generally, DeWitt clauses in government contracts only, or DeWitt clauses in licenses for cybersecurity software only.

A generic law might be pattered after UCITA section 105(c), which provides that "a term of a contract is unenforceable to the extent that the term prohibits an end-user licensee from engaging in otherwise lawful public discussion relating to the computer information."

Attempts to pass such a law at the federal level could encounter industry opposition. A more modest legislative fix might limit itself to software licenses with the federal government or to licenses for cybersecurity software. The former approach would permit DHS and other federal agencies to share benchmarking and other test results among themselves. The latter would have even broader benefits. Cybersecurity (a) is an area of special national concern, (b) provides considerable positive externalities, and (c) benefits especially from the sharing, by both public and private actors, of benchmarking and other testing information. Taken together, these factors support Congressional action to limit the effectiveness of DeWitt clauses in licenses for cybersecurity software.

Here again Microsoft's license for the .NET Framework might be used to address industry concerns. Legislation might prohibit blanket DeWitt clauses currently common in the industry, but permit contract terms that require best practices and publication of methods.

## Further Research

Our research has, we believe, studied all of the legal authorities relevant to the enforcement of DeWitt clauses. Next steps would therefore focus on solutions. One avenue of further research would be to draft model legislation or regulations addressing the enforcement of DeWitt clauses in EULAs for cybersecurity products.

## Acknowledgements

The authors would like to express our thanks to Dr. Barton Miller, Professor of Computer Science at the University of Wisconsin-Madison, who was present at the birth of the DeWitt clause and gave us invaluable advice as well as Dr. James Hill, Associate Professor of Computer Science at the Indiana University-Purdue University Indianapolis. We would also like to express our thanks to Mr. Kevin Greene of the Department of Homeland Security Science & Technology Directorate for the genesis of the project and for connecting us with the numerous people who shared contract samples with us.