Los Angeles
Daily Journal

September 2, 2010

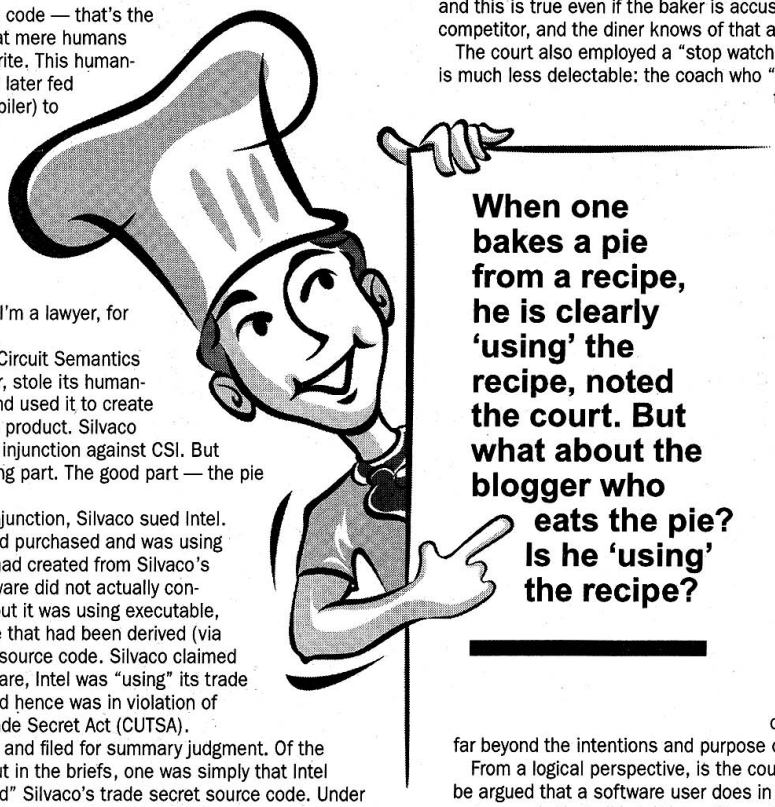
Eating Pie and Trade Secret Theft, What's the 'Use?'

By Michael Young

It is not often a case hits upon two of our favorite topics: Trade secrets and pie. But when it does, mmmmmmmm, it's good to be an attorney.

And for this reason, we are grateful to Justice Conrad Rushing who provided us with a little of both in *Silvaco Data Systems v. Intel Corp.*, 2010 DJDAR 7942, 6th App. Dist. (April 29, 2010).

Silvaco is a relatively standard trade secret case in the computer software world...with a twist. It starts with the alleged theft of source code — that's the computer language that mere humans can understand and write. This human-drafted source code is later fed into a machine (a compiler) to produce "object code" (or "target code"), which is the nearly indecipherable gob-bledygook that can be executed by the computer itself. (Yes, I know this is a slight oversimplification, but I'm a lawyer, for cryin' out loud.)



When one bakes a pie from a recipe, he is clearly 'using' the recipe, noted the court. But what about the blogger who eats the pie? Is he 'using' the recipe?

Silvaco alleged that Circuit Semantics Inc. (CSI), a competitor, stole its human-drafted source code and used it to create a competitive software product. Silvaco eventually obtained an injunction against CSI. But that's not the interesting part. The good part — the pie — comes next.

After obtaining the injunction, Silvaco sued Intel. Why? Because Intel had purchased and was using the software that CSI had created from Silvaco's source code. The software did not actually contain the source code, but it was using executable, machine-readable code that had been derived (via the compiler) from the source code. Silvaco claimed that by using the software, Intel was "using" its trade secret source code, and hence was in violation of California's Uniform Trade Secret Act (CUTSA).

Intel begged to differ and filed for summary judgment. Of the many arguments set out in the briefs, one was simply that Intel hadn't "misappropriated" Silvaco's trade secret source code. Under CUTSA, to "misappropriate" a trade secret, one must either "acquire," "disclose," or "use" the secret.

Silvaco argued that Intel "used" the source code when it ran the software since even though the software was executing the object code, the object code was based on the stolen source code. This is not a wholly frivolous argument.

But the court wouldn't hear of it. Instead, in granting Intel's motion, the court decided to talk pies and pie recipes. When one bakes a pie from a recipe, he is clearly "using" the recipe, noted the court (to which most bakers would, I presume, agree). But what about the blogger who eats the pie? Is he "using" the recipe? Or just enjoying the fruits (or chocolate creams) of the end product? Mmmmmm.

The court held that the eater of the pie is simply a happy diner...even if he knows the baker stole the pie recipe in the first place. He is not a "user" of the recipe itself: "One who bakes a pie from a recipe certainly engages in the 'use' of the latter; but one who eats the pie does not, by virtue of that act alone, make 'use' of the recipe in any ordinary sense, and this is true even if the baker is accused of stealing the recipe from a competitor, and the diner knows of that accusation."

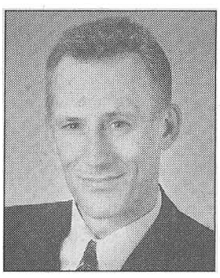
The court also employed a "stop watch" analogy, though the imagery is much less delectable: the coach who "uses" a stopwatch, according to the court, cannot be said to be "using" the trade secrets that went into manufacturing the stopwatch.

Finally, the court looked to "public policy" to support its conclusion — if a software user (like you, for instance) were considered to be "using" the underlying source code that was used to generate the object code that was running the software, then every software user (like you, for instance) could be liable for trade secret misappropriation if it later turned out that the software manufacturer utilized purloined source code. (Did you follow that?) If software end users like you and me are at risk of trade secret misappropriation for using software based on stolen source code, "this risk," according to the Court, "could be expected to inhibit software sales and discourage innovation to an extent far beyond the intentions and purpose of CUTSA."

From a logical perspective, is the court right? Can't it reasonably be argued that a software user does in fact "use" the underlying source code that allowed the software to be developed? Doesn't the pie eater "use" the recipe when he or she eats the pie? The court relies on the dictionary definition of "use" to support its interpretation, but doesn't this definition still beg the question: "As it appears in the act, the noun 'use' is surely intended in the ordinary sense, i.e., '[t]he act of employing a thing for any (esp. a profitable) purpose; the fact, state, or condition of being so employed; utilization or employment for or with some aim or purpose, application or conversion to some (esp. good or useful) end.' (19 Oxford English Dict. (2d ed. 1989), p. 350.)"

It seems to me that "use" is more of a continuum than a bright line, like proximate cause. The pie eater is "using" the recipe, but not as much as the baker "used" the recipe. The personal trainer who is getting paid to help the pie eater lose weight is also "using" the recipe, as is the personal trainer's tanning salon (hey, you've got to look good in that business). At some point, though, you've got to cut off "use" just as the courts have to cut off proximate cause. The use of "public policy" to help define where that cut-off should be certainly seems to make sense.

Personally, if it means I can bite into that lemon meringue without fear of being sued, I'm all in favor of it. So is my trainer.



Michael Young is a partner in Alston & Bird's Los Angeles office with particular expertise trying and resolving employment, trade secret and intellectual property disputes. He is also the editor of the firm's California employment law blog at www.alston.com/laborandemploymentblog. He can be reached at mike.young@alston.com.