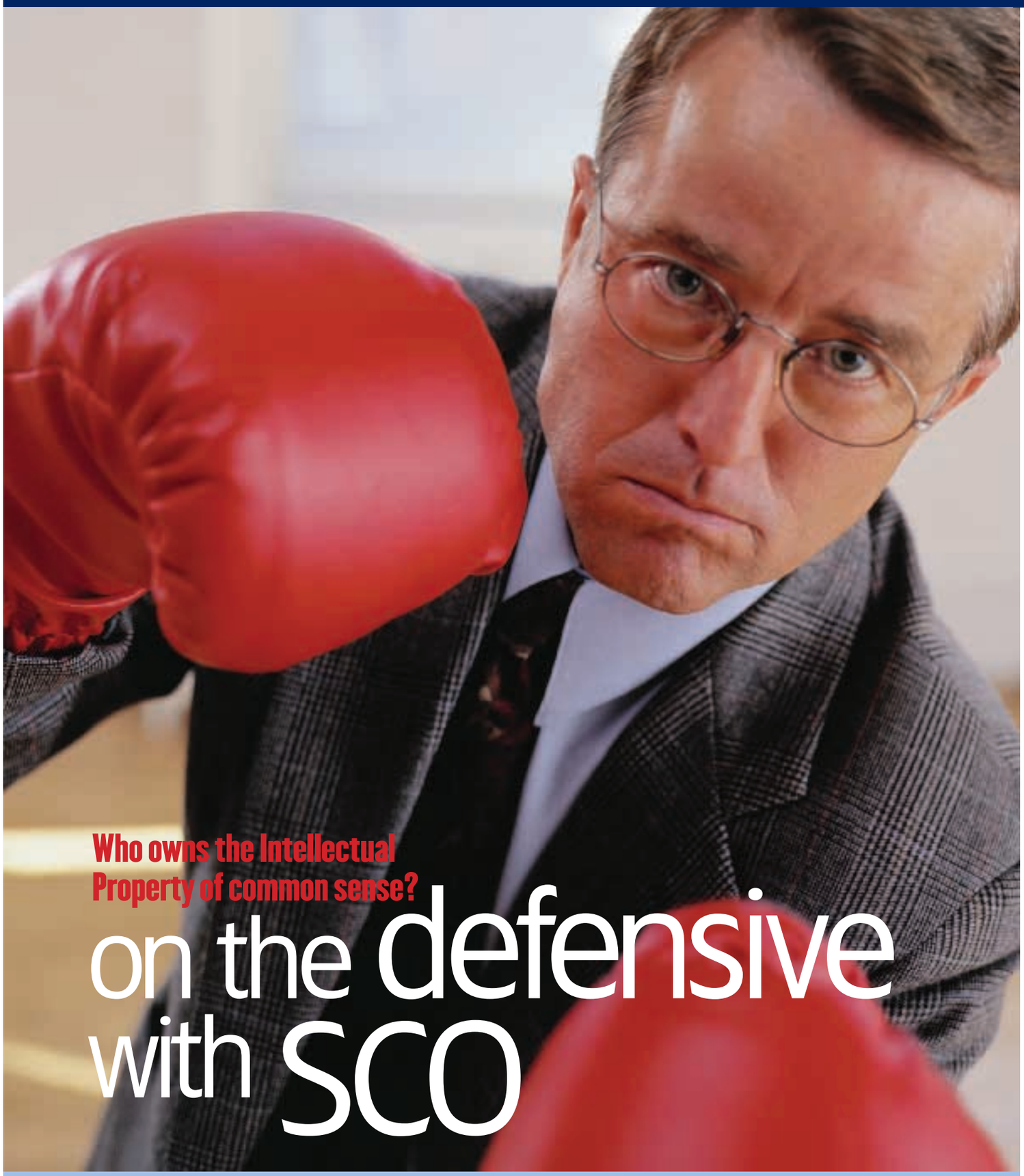


Who owns the Intellectual Property of common sense?

on the defensive with SCO

The SCO vs IBM case is unlikely to damage Linux, but as DAVID HARRIS explains, as well as being handy publicity for alternatives to restrictive copyright, it does highlight the need for careful coding practices and some of the risks that must be considered by FOSS developers.

Well, SCO has certainly put the cat among the pigeons. Back in issue 40 we discussed liability in general terms; peer-to-peer, criminal liability and so on. *SCO v IBM* had just kicked off and was looking like an important – albeit not central – issue in Free software. Since then, IBM has refused to either buy SCO or pay it off, causing the company to have a temper tantrum and throw its rattle out of its pram. SCO has now decided that Linux is a hippie pot-smoking communist conspiracy to steal its code and undermine the American way.

SCO has retained well-known lawyer David Boies, famous for losing against Microsoft, losing for Napster and losing for Al Gore; Linux fans can only hope his performance remains consistent. It is still way too early to be certain of the outcome of the case, regardless of the apparent inadequacies of SCO's position so far. Nonetheless, the whole debacle has raised more uncertainty among some commercial users of Linux than would appear from the outward expressions of faith and confidence among the user community.

Money talks

Who would be best served by such uncertainty? Well, SCO would if it caused people to buy their licence at US$700 a pop, but thus far there is virtually no sign of them doing so. However, there is someone else lurking under the eaves: our old friend Microsoft. It may well be that despite paying $10,000,000 in supposed license fees, they are entirely uninterested in the outcome of the case. Equally, since Microsoft uses relatively little Linux, the view is that they are seeking to wage war by proxy. If SCO succeeds in its lawsuit, its next attack is likely to be on the 2.2 & 2.3 kernels, and eventually Linux becomes a hostage under their control. SCO would then be bought out, marginalised or undermined by anti-competitive pricing by Microsoft, and the threat of Linux would finally be neutered. At worst, Microsoft buys time while the Free software community tries to replace Linux with an alternative such as a fully clean Linux or HURD.

What is more important – and a greater threat – is that the core of Free software risks being tainted. For several years, Microsoft has waged a campaign about the risks of Free software to intellectual property. It says that Free Software will contaminate proprietary products and make IP ownership uncertain; or it says that the use of Free software will cause proprietary rights to leak into the public domain via the GPL. With the SCO case, MS has a real example to point to and say "Ah-ha! See, we told you. If you dabble in the GPL this is what will happen. You risk both being sued and having your IP stolen". As we all know, society will collapse if Microsoft can no longer make monopoly profits and its hegemony of the IT industry is threatened...

What has happened is that a change of management at SCO has lead to a re-evaluation of profitability, resulting in a view that cannot see revenue potential from FOSS – possibly since the competition is too great and its own offerings are too poor. The intent is to revert to older business models, and so SCO needs to renege on its deal with the FOSS community. At the same time, SCO needs to eliminate the competition that previously devastated them by making highly questionable claims about Linux IP. Dubious claims of copyright infringement appear to be a part of Microsoft's attempt to effect that revisionism.

Clearly, the potential ramifications of SCO are serious and profound, and may prove damaging to FOSS in ways as yet unclear, but at this point I shall remove my tin-foil hat and get down to the legal analysis and commentary.

Several companies have asked me questions on the SCO case, like:

- Does SCO have good prospects of winning?
- Should they stall implementation while SCO goes on?
- What can be done if SCO do win?
- What are the implications of mixing of proprietary products and the GPL?

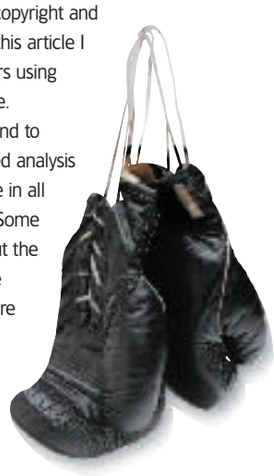
DISCLAIMER

THIS ARTICLE IS BASED ON UK law except where otherwise indicated. Substantial differences exist between UK law and that elsewhere. The consequence of all law varies greatly with individual circumstances and thus nothing in this article is intended as or should be construed as advice or acted on without seeking your own lawyers' advice. The author regrets that he cannot give personal legal advice, except in very specific cases, as is stated in outline at the bottom of page 12 of this issue.

"SCO has retained well-known lawyer David Boies, famous for losing against MS, losing for Napster and losing for Al Gore."

A number of common themes in copyright and licensing arise in these questions; in this article I hope to guide companies and hackers using contributing and writing Free software.

First, I should say that I do not intend to provide an exhaustive or even detailed analysis of SCO's claims in their entirety, since in all truth, they vary from week to week. Some cynics have pointedly remarked about the regularity of announcements and the consequent see-sawing of SCO's share prices, but we don't want to go into that. No, I intent to examine a few SCO complaints that illustrate more general problems in FOSS.



SCO'S COMPLAINT

SCO claims to own the intellectual property of UNIX as a result of its purchase in 1995 from Novell of the original AT&T codebase and patents (see timeline *SCO vs Linux* on page 11).

This is however a deceptive aspect of SCO's case: it has claimed in several statements to own the UNIX operating system, and through numerous contractual arrangements to be entitled to control the rights of all vendors to use and distribute UNIX. At the same time it has acknowledged that the Open Group is the owner of the UNIX trademark. The Open Group has however stated that *UNIX* is not a particular code implementation but rather a trademark and an associated specification. The Open Group certifies certain implementations of the standard as being conformant to it and hence being *UNIX*. There can thus be multiple Unices; this has already happened with IBM's OS/390 which has been certified as UNIX 95 and thus *UNIX*. Clearly, SCO does not own UNIX (the trademark) but one

particular historical codebase. As we discuss below, it may not even own all of that. Further, SCO claims to be able to licence further use of UNIX but this appears not to be the case: UNIX is a trademark that they do not own, but that they merely have a *licence to use*. SCO has no power to sub-licence the trademark from the Open Group, and hence on one argument, it has no power to grant rights to derivatives of UNIX. There is a countervailing argument that they do have an entitlement to grant a licence to the UNIX trademark if their original contract with the Open Group is to have any commercial meaning. However, even if true this would not give them the *exclusive* power to licence anyone else's UNIX implementation. Someone such as Linus Torvalds, the FSF or a wealthy benefactor could pay the Open Group to certify Linux as UNIX-complaint.

"This isn't a complex case – it's just that the facts have been hidden by SCO, and as a result, the case may be long and costly."

Of course, none of this is relevant to the issue of code copying. If SCO's version of events is correct, and those 'Linux hippies' have stolen SCO's valuable IP and added it to the Linux kernel, the above arguments would be an irrelevant side-issue. It does however seem a little surprising that SCO, who have hired a supposedly top-notch team of litigators, haven't fully debugged their case statement and seem to be indulging in some generally sloppy work.

Copyright infringement

Direct copying

Is there any real basis for SCO's claims of copyright infringement? SCO seems to think so, but the evidence so far has not been entirely supportive of the thrust of its claims. During its resellers show, to support its vague claims of millions of lines of stolen code in the kernel, SCO revealed a sample code fragment partially obfuscated in a Greek font. This was a mistake, since hackers decrypted it rapidly and demonstrated it to be code whose heritage went back as far as 1968 to Donald Knuth. The code in question bears an SGI copyright notice, but SGI seems to have been somewhat cavalier with copyrights; it appears to have stripped the old copyright notice and replaced it with its own. This was clearly wrong and a copyright violation on SGI's part. This is in breach of both the advertising clause of BSD and the non-removal of notice clause. This file has been removed from the kernel, but that doesn't alter the malfeasance in SGI's act. Simultaneously, there is yet another version of this same code fragment obtained elsewhere under a BSD licence that would be perfectly legal to put in the kernel.

Taken at its worst, this infraction would result in SGI facing damages of a few tens of thousand of dollars, and probably much less, for altering attribution and what was a technical infringement of the licence; after all the code dissemination in that file had been agreed to by SCO (or its forbears). Would this merit a Linux licence fee of several hundred dollars a CPU to SCO? Not in any universe near me.

The other code fragment SCO laid claim to at the show is the *Berkley Packet Filter* (*BPF*). This has been released on a BSD licence that Linux and SCO have always been able to use, and it seems that SCO copied it into System V. The Linux implementation of *BPF* however is an original work, based on a specification not owned by SCO. When SCO points to code similarity, it may well be not so much that Linux has copied SCO code, but that both have the same parents.

Clearly, this indicates how complex the factual matrices in litigation can arise, and this is not really a complex case – it's just that the facts have been hidden by SCO, and as a result, the case may be lengthy and costly. One can speculate however; as Bruce Perens has said, it is plausible that SCO have chosen the best examples they have and these are they. If so, then the danger to Linux seems minimal. None of this would justify the removal of 2.4++ kernels under an injunction nor any substantial damages. One hopes that any other examples are no more damaging. The apparently aberrant behaviour of at least one SGI employee does raise serious questions about code control, and is one of the more reasonable points made by SCO that should be absorbed, as I discuss later.

SCO's claims of infringement rest at two levels. One is the literal copying we have just discussed. We have seen several published examples so far; however, the code copied is either in the public domain, or the copying and derivation of the code by Linux developers has been agreed to at a conceptual level – it is merely that SGI took credit where they were not entitled to. It is not as though vast tracts of source code that had cost SCO a fortune to develop and had laboured hard to keep secret were stolen by industrial spies. We shall see in the fullness of time

whether SCO actually has better examples of copying. Nonetheless, copyright infringement is wrong, it should not happen and the FOSS community needs to be continually aware of the risk and make efforts to avoid it.

We can, I think, be less diffident about the other claim of infringement. The *Berkley Packet Filter* claim seems to be plain wrong on the facts so far disclosed. *BPF* was based on code written by Carnegie Mellon University put into 4.3 BSD and then copied by SCO into System V. They were entitled to do that, but not to claim, as they are now doing, that they own it and can eject others from its use.

Non literal copying

Not all copying is line-by-line, sometimes there is a deliberate attempt at obfuscation by infringers attempting to cover their tracks. At its crudest, this may involve no more than stripping off copyright notices changing variable names and shuffling lines of code around. This is crude, not uncommon and effective enough in most cases. Equally however, there can be a copying of an idea and significant chunks of its expression in reworked form. When this type of case comes to court, the assertion of copying is tested by first asking whether there is any causal connection; did the defendant copy it from the plaintiff, did the plaintiff copy it from the defendant or did they both copy it from some common source? If this sounds familiar in the context of the SCO complaint so it should. To determine such causation, questions are asked such as 'how similar is the code' 'were the *dramatis personae* linked in some way?' – *eg* was the alleged copier an ex-employee of the plaintiff? 'Was there an opportunity to copy the code? Did the plaintiff show it to the defendant for some reason?' In the SCO case, the broad accusation against the FOSS process is that various people in companies with access to System V source have, in solving coding problems, peeked at the System V source and obfuscated it into Linux.

Subconscious copying

Another form of indirect copying often seen is subconscious copying. It may well be that the defendant believes – in all good faith – that what they produced was their own, but in reality they have seen another work, forgotten it and then reproduced it. Most of the cases have come from the music industry for example George Harrison subconsciously copied parts of the tune for *My Sweet Lord*. This form of copying is less frequently seen with software, possibly because the more sophisticated subject matter means the scope for substantial taking (see below) is reduced.

Core protect-able ideas

The other main form of non-literal copyright infringement is indirect copying, and it seems that SCO originally seemed to be accusing Linux of this. If it were not possible to infringe copyright by copying a copy, the practical effectiveness of copyright law would be very damaged. One could turn Free software into a proprietary product in defiance of GPL merely by making a copy of someone else's infringing source code. Not a good thing.

A simplistic truism has developed in the FOSS community that one can only protect the expression of an idea rather than the idea. To paraphrase one judge: *"yes, but it rather depends what you mean by 'expression'"*. The point being that there is seldom a fixed dividing line between an idea and a fully developed exposition of an idea. True, no one can have a copyright in detective novels, but equally one cannot write a Sherlock Holmes novel with a hound and a moor. What about an English detective who employs forensic techniques and has a male sidekick? The point here is that in reality, there is a sliding scale between an idea and a its concrete expression, and whether there is infringement depends on where on that line you fall. This is referred to as *substantial* taking, and at varying stages appeared to be a SCO claim: that they owned System V and all Unix IP. SCO says that by copying core concepts, using the inspiration of Unix, or subconsciously copying Unix as result of seeing it at college (euphemistically called 'learning'), there is infringement. In assessing alleged copyright infringement, the principles we've mentioned would be used to assess whether there had been substantial taking. In the US, when software copying is alleged the complicated 'Abstraction Filtration comparison' test, from *Computer Associates v Altai* is used. This involves a three-step process of:

1 Breaking the component into its functional structures, then;
2 Examining those parts for mere ideas, matter incidental to an idea or matters in the public domain; finally
3 Comparing the remaining kernel/s of potentially protect-able matter with the alleged infringing program. This test is not without its critics, and the overly mechanical approach is said not to catch all infringement. In other jurisdictions, *eg* the UK, a less formal holistic approach is generally adopted; *eg* that from *Ibcos Computers v Barclays Mercantile*.

Another argument that has appeared to be offered by SCO is that it owns the IP in System V Unix, and since Linux derives (at the very least conceptually) from Unix, it is a derived work for the purposes of copyright law. SCO says that this has happened either as a result of adaptations made to its copyrighted source code, or because many programs that run on Linux also interface with elements SCO claims to own; for instance, APIs or methodologies. The former argument will be resolved on the facts in court, but the latter argument is almost too easy to defeat. Much of the supposed IP is based on general methodologies developed since the 1960s, and much that has been taught on computer science courses in universities to countless generations of students. The scope for arguing that any of this is, ever was or remains, proprietary is, I would suggest, highly limited. The suggestion that because a program uses an API or methodology, it becomes a protectable derivative work, is absurd. To the extent that there is property in the API or methodology itself, then anything other than the most trivial of implementations would probably involve sufficient work to give it a self-contained copyright that is distinct from any in the API or methodology.

However additionally much of the Linux kernel and associated code is crafted to comply with a Posix API, not some imaginary SCO API; in any event as discussed earlier it is X/Open that owns the API not SCO. As for the derived work argument, we have already seen that in reality Linux is based on the X/Open API's, public

Darl McBride, SCO's CEO – stock-pumping son of Satan or ultra-shrewd businessman, depending on exactly who in the computing world you ask.

domain code, the general state-of-the-art, BSD code and the ingenuity of Linux coders.

It is absolutely impossible to know at this stage what they outcome of this analysis would be; not least because at the time of writing, SCO still won't indicate which specific part of the (publicly available) code it claims to be the infringement.

Doctrine of pre-emption

Mark Heise, a lawyer with Boies' firm, has impeached the validity of the GPL in America in a novel way; he says *'Section 301 of the Copyright Act says the Copyright Act pre-empts any claims that are governed regarding use, distribution and copying. We believe that although the GPL is being tossed into the fray, it is pre-empted by federal copyright law.'* He goes on to say that US copyright law under section 117 of the *Copyright Act* permits the making of only one copy of software and federal law pre-empts attempts to forbid that. So what is pre-emption? It is an American legal construct governing any conflict between contracts, licences, common law, state law and federal laws (purists will argue that pre-emption applies within the EU also as between EU legislation and national law). At a very crude level, it states that where these limit the rights given by Federal law, Federal law prevails; if they try, directly or indirectly, to remove rights granted by Federal law, they are invalid. Since the GPL allows any number of copies and federal law permits one copy, the GPL is thereby unlawful. This is nonsense of the most arrant sort. The purpose of the section on any sensible reading is to give people a minimum right to a backup copy, and neither the intent of the Act nor its text prohibits greater copying. How any lawyer can offer such an argument is truly perplexing. There is, of course, the possibility that SCO has formulated some clever argument relating to the viral element of the GPL with the intent of neutering it, but if they have, they have not disclosed it nor is it easy to guess. A more realistic hypothesis is that Heise is not a copyright lawyer. He has been given a brief and has done what research and analysis he can, but has made mistakes. Were I asked to handle an ecclesiastical case, I'm sure I'd make a proper mess of it too.

Trade secrets

I have seen it said that by publicly distributing code SCO has foregone any trade secret rights. Indeed the judgement in the BSD case, *USL v BSDi*, supports this view; in that case the judge said that header files, filenames and function names were *'...not secret since these were all available from unprotected files...'* However, the fact that code is available everywhere without a confidentiality obligation is not relevant if the overall structure of the infringed code is such that it is difficult to determine that structure, unless an existing copy is downloaded to circumvent the experimentation or development needed to get that structure from the code. Whether this argument applies to an entire operating system code is untested, but it would seem to be a viable argument in principle.

The second argument that SCO is using is that it cannot disclose which source files are infringing, since this would be tantamount to disclosing their source; at the moment, they say, it is hiding in plain view, and until they disclose it, no-one knows which bits are theirs. The difficulty with this argument is that while true only up to a point, you have to step back and ask *'what is the purpose of a trade secret'?* It is to make sure that no one can copy your source code or use it to their advantage or your disadvantage. If that code is already in the public domain, they can do this anyway; they can rifle through the Linux source taking whatever bits they want while probably being indifferent to its SCO origins. Only if an adversary were specifically interested in knowing what SCO used would this be an advantage, and it's a bit of a stretch to think up anything credible. One imagines however, that SCO would be more interested in ensuring its code was stripped from Linux rather than allowing its misuse in order to keep it secret and maintain some unclear trade secrets advantage.

"The suggestion that because a program uses an API or methodology, it becomes a protectable derivative work, is absurd."

Furthermore, there is a principle in law that where one suffers damage, one must attempt to minimise it using whatever steps are reasonable in the circumstances; the principle of 'mitigation of damage'. Whether the continued refusal to disclose allegedly infringing files breaches this principle will depend on whether SCO's arguments that it is reasonable not to do so hold water. I'm a little sceptical given their previous statements and the general demeanour of the company.

An additional point about trade secrets is that to protect them completely there needs, broadly, to be either a relationship between the person disclosing and the person disclosed to – *eg* seller and buyer – or a relationship of sufficient proximity that the court would feel it equitable to

bind the recipient of the trade secret in the absence of any other relationship. That is not to say that an innocent recipient of a trade secret is automatically home free in the UK: an injunction and damages might well be had; however, in US that is the best presumption. In the US, it has been said that a trade secret once publicised is lost: '*...once that trade secret has been released into the public domain there is no retrieving it...*' (*Religious Technology Centre v Netcom Communications*). However people have to have seen it for it to be *'released'* which raises the issue of whether a trade secret hiding in plain sight, such as in kernel source, has been released. For historical or other reasons, some of what SCO claims to be its trade secrets have leaked into the public domain; either through general education of the

IT industry, the improper access to SCO source code or because of discovery in litigation (eg *USL v BSDi*). Where this is unlawful the remedy is likely to be an award of damages against any unlawful discloser. It is possible that a court might injunct the distribution of code but it would have to balance the effectiveness of the remedy against the effect on innocent third parties and whether it is a proportionate measure. Additionally, if the innocent kernel developers relied on the *bona fides* of ideas they received and so developed other code based on it, there is the equitable doctrine of *'change of position'*, an estoppel to defend against an injunction or damages being granted. Again, I'm somewhat sceptical that they would get anything other than damages at the very, very best.

SCO'S PROSPECTS

I'll be honest: I don't know if SCO will win, fully or partially. Insufficient papers have yet been disclosed via discovery to find out (discovery is the process whereby parties to a case are required to reveal the evidence they will be relying on, and pertinent material the other side wishes to see). IBM has just applied for what is a very broad level of discovery, and I imagine some of this will become available for analysis. However, the evidence is so far mixed for both sides. SCO has made some very peculiar assertions, and failed to undertake a sufficiently clear analysis to distinguish between historic, public domain and SCO non-proprietary or proprietary code. The court atmospherics of much of this would, despite SGI's behaviour, seem poor for SCO; though between now and the hearing, additional evidence or better arguments may compensate. Certainly SCO should not have employed a media legal star with negligible technology experience just for the PR value. The case needs a specialist technology law firm (Sorry SCO, we're busy).

SCO vs LINUX

History of the dispute

1968

Donald Knuth writes (probably reworking earlier papers) some packet filter software. Thompson and Richie at AT&T write portions of code used in UNIX.

1985

IBM take a UNIX licence from AT&T.

1993

Novell buys UNIX source & patents from AT&T.

1994

Novell sells full UNIX Licence to Sun.

1994

Novell sells UNIX trademark and UNIX specification to the Open Group.

1995

SCO buys UNIX from Novell – some confusion over whether all patents and copyrights transferred – Novell can't find their copy but confirm the signature is valid, so that some copyrights may be owned by SCO.

Feb 2001

SCO & Caldera merge. Later release pro-Linux statements.

June 2002

Darl McBride takes over as SCO CEO.

2002

Caldera releases code as Open source.

Jan 2003

LinuxWorld Expo: IBM makes a keynote speech about "Linux coming of age" that reportedly upsets McBride by stating their intent to "obliterate UNIX". McBride then hires Boies' law firm.

Jun 2003

SCO begins showing, under non-disclosure agreements, selective limited code fragments bearing a close similarity to code in the kernel.

Jul 2003

Evidence comes to light of SCO forbears directly contributing code to the Linux kernel. SCO begin offering a 'get out jail free' licence to Linux users. SCO continues to offer the Linux kernel from its own servers. SCO refuses to disclose alleged infringing code.

Aug 2003

Red Hat and IBM counter-sue. SCO declare their strategy will defend in part on the invalidity of the GPL, dubbed by some as the 'Chewbacca defence'. SCO reveals two examples fragment of code which are traced to public domain code or no-copied code based on public domain specifications.

WHAT IF A UK LINUX USER IS THREATENED BY SCO?

I do not think SCO presents too much of a problem for European or UK users at present. If SCO approaches anyone in the UK, then the chances are that it will ask for a licence fee based on the argument that all (under a derivatives works theory) or some of Linux is their copyright for which, therefore, a licence is required. In the US, Eben Moglen, Counsel for the FSF, has offered the view that one need not pay since the US Copyright Act permits the running of a

computer program by copying it onto a computer without the need for a licence. This may be good law in the US, but in the UK it doesn't run; section 17(6) of the *Copyright Design and Patents Act1988* provides that copying includes transient copies such as those made by copying into RAM: the opposite position from the US. In the UK mere use of the kernel irrespective of whether the source code is used would be a copyright violation *if* SCO's contentions are right.

Nonetheless, paying up has one advantage; it will at least alleviate you from the risk of litigation, but the price is a heavy one since SCO are requesting $700 for single-user licences. Multiply this across an enterprise, and it will make Windows look cheap. Nor does SCO promise you your money back if the licence turns out to worthless, indeed it explicitly says the opposite. They exhort users to act honestly and with integrity without feeling the need to do the same: if they were as confident as they claim they would put their money where their mouth is. Of course there are arguments to make that in these circumstances any contractual licence is void. Under the doctrine of frustration for example one might argue that the purpose of the contract was frustrated because the alleged SCO IP rights forming the basis of the contract had vanished and that SCO therefore has to reimburse the contract price.

Were I to be approached to buy a licence, I would ask them to provide detailed proof of the alleged infringement including a listing of infringing portions of the code; tell them to sue me; or wait until an English court declares that there is an infringement. If threatened with litigation, I would also seek discovery of the alleged infringing code with a view to validating the claim of infringement or getting that code removed to provide a 'clean' kernel. This is of course at the heart of the dispute between the FOSS community and SCO, and it is this that is the proof that the whole SCO circus is a protection racket. An honest company, on discovering their code had been stolen as SCO alleges, would be keen to sue for damages and an injunction; to protect their IP they would also be only too anxious to get it removed and for infringing

products withdrawn and destroyed. This is all standard fare for IP actions. SCO however has proven that it is a shyster by attempting to maintain the supposed infringement in the hope of extracting ongoing licence fees from a wide variety of sources. If they sued for damages and won they would probably only get a one off damages payment of X million dollars; perhaps the calculation is that they can get much more by extracting licence fees from all current and future users. This Machiavellian goal would, however, be shot to pieces by FOSS hackers doing the honourable thing and removing any supposedly infringing code; indeed an honest victim of copyright infringement would insist on it.

COUNTER LITIGATION

Linux is of course an international phenomenon, and the SCO case is a US one; but the substance of the case is not such that it that would prevent SCO repeating litigation in Europe or elsewhere. The result may not be entirely the same, though an analysis is beyond the scope of this article. LinuxTag in Germany has already obtained an injunction against SCO Germany for unfair competition, since SCO is engaging in FUD and not disclosing proof of the alleged infringement. SCO's failure to respond substantially so far is consistent more with its focus on litigation in America, and containing the issue elsewhere. SCO has a litigation cash pile of about US$10M dollars and it would be nice to see that eaten up by Linux users around the World taking SCO to court; though one suspects Microsoft would merely find an excuse to buy another US$10M licence. As an intermediate step this litigation in Germany is a good thing and it would be nice if it were happening elsewhere; I'd be happy to provide *pro bono* advice to any company wanting to do this in the UK.

Criminal sanctions

I have seen comments from some people who (understandably, but a little unrealistically) want to press criminal sanctions against SCO. Putting the share-pumping allegation to one side and considering just the issue of allegations of copyright infringement, I'd regard this as wildly unrealistic. While many would like to see Darl McBride share an intimate, if coerced, moment with Bubba and his friends in the showers of a Utah penitentiary, it is unlikely to happen. Suggestions have been made of blackmail, fraud, extortion and such like. These certainly capture the feelings of Linux users, and at a colloquial and pejorative level they'd be accurate. Unfortunately, the standard of proof required to prove criminal allegations is quite substantial: *beyond all reasonable doubt*; and I doubt if sufficient proof exists to meet that burden. It would be necessary to demonstrate that what was said was false, known to be false (or without any belief in its truth) with a dishonest intent to deceive and obtain a financial advantage. No, I do not think so, unfortunately.

CONTRIBUTING CODE TO FREE SOFTWARE

The risks

Finally, we come to honest participants in the FOSS process. SCO started off – albeit reluctantly and through necessity – in this process several years ago. It was not until Darl McBride joined SCO and tried to hijack Linux JFS and NUMA that significant thoughts about the resilience of the process arose.

Part of the problem here is to avoid adverse consequences, such as SGI's possible malfeasance in relation to the inclusion in FOSS of other peoples code by stripping out copyright notices. The issue is how to avoid such things in future. In part Google, education and open mailing lists are the answer. Linus Torvalds has said he is well aware of the need for clean code and accountability, and I have heard that the process for inclusion of code in the kernel involves elaborate and lengthy public discussion along with assurances of code provenance. This is clearly a good – if minimal – approach. If someone does include code that they should not in a FOSS project, the project team may all be liable for copyright infringement in certain circumstances. It may seem a bit bizarre to say that all x-thousand kernel developers may be liable if copyright infringing material appears. In principle however, there is an argument for finding joint and several liability attaching to some or all involved members of a guilty project. Of course, for such a large project as a kernel, this would seem an unreasonable and severe approach, given the massive size and complexity. A more realistic approach would be for liability to attach to all or some of the members of a subsystem team, *eg* the USB or Ext3 team. Scary? Yes somewhat. Education has a vital role. If the SCO affair has a positive function for Linux, it is to reinforce the message that not everyone likes FOSS. Some see it as a lethal and pervasive challenge to their business that must be killed by all means fair or foul. Well then, it is understood.

ABOUT THE AUTHOR

DAVID HARRIS IS AN IP barrister, practising at www.ukitlaw.com. When not suing people, he writes 'depressingly poor code'.

THE CONTRIBUTION PROCESS

A collection of coders' caveats

For any collaborative project of mine, I would do or attempt some or all of the following (though this is a minimal approach);

- **Create a rigorous but minimally complex process for accepting contributions: knowing exactly who one's contributors are and who employs them. Essentially, this would be for the purpose of flagging up potential issues such as confidentiality, competence and ownership of code.**
- **Undertake an educational process for private contributors about the importance of copyright and other IP issues. This would be public, and all contributors would be invited to view it. While far from a defence, it does at least set the correct parameters.**
- **For corporate contributors with legal departments, I would engage them in a proper process of auditing submitted code. SGI did this with XFS for example, and it delayed its contribution for quite a while; likewise IBM with its kernel contributions. In both these cases I have faith SCO will not prevail for these reasons.**
- **My own preference for code that has no potential to be commercialised would be to assign copyright to the FSF. This does not, under English and US copyright law, require formalities beyond putting down in writing a clear intent to transfer fully permanently and irrevocably all rights to the FSF. Then the copyright owner must physically sign the document and should send it snail mail to them together with a disk of the code.**

For code where there is, however remote, the potential to commercialise, I would not assign the totality of copyright. The legal options are numerous: granting joint ownership, offering the FSF a contractual GPL licence, creating a commercial derivative and assigning the Free version to the FSF plus a number of other variants.

- **If creating code based on the works of others, I would attempt to use publicly available non-confidential documents in order to obviate trade secret accusations: Usenet postings, mailing lists would be good for this purpose.**
- **When coding to standards, specifications or APIs, I would attempt to evaluate whether the owner claims them to be nonetheless proprietary. Obviously there is no generic approach to this, and the options would depend on the circumstance and who the owner was. People such as the ISC are probably OK, but asking Microsoft for formal consent to follow its 'standards' is going to be rather problematic; however that has not yet hampered *Samba* or *OpenOffice.org* (though that may well change). Whatever the response I would print off and archive the response as I would to any issue that may require proof several years later.**
- **Pray to St Isidore (believed by some to be patron saint of computing and the Internet) and a broad selection of the most powerful gods I can find that there are no software patents. ■**