Memorandum

To: *Steve Sabbath*
From: *Robert Swartz*
cc: *Doug Michels*
Date: *October 4, 1999*
Subject: *Linux*

Privileged And Confidential

Draft

This work was done at the request of Steve Sabbath

Dear Steve:

As you requested below is a draft of my report on existence the of Unix derived code in Linux. What we tried to do is to determine if there was any material from Unix in the Red Hat Linux release 5.2. To make this determination we used a copy of Red Hat Linux which was purchased from the local Best Buy. We then compared it to multiple copies of Unix. We received sources from SCO of OpenServer 5.0 dated January 27, 1998, Gemini Source dated January 27, 1998, it being UnixWare 7, UnixWare 3.2, UnixWare 4.0 and UnixWare 4.2. Additionally we received various versions of files which were not on the release described. To perform this work we unpacked and obtained the sources for Red Hat from the CDs which are provided with the Red Hat release. We then compared for each similar set of programs and files in Linux and the various versions of Unix. For example we compared Unix *yacc* to the Linux *bison* and the Unix *awk* to the Linux *mawk*. We performed this comparison on all files which had similar functionality.

We used the following method to determine whether there was any similarity between the Linux and the various releases of Unix. First we found the comparable files in the various version of Unix and Red Hat. This might not always be files with the same name. Further in general for the purposes of this work we would often concatenate the files together which represented a single program. We then used a program call *ef* to perform the comparison.

ef works by looking for the number of consecutive line in two files which are identical. So for example if you have two files **A** and **B.**

a	*K*
b	*l*
c	*m*
d	*duplicate1*
duplicate1	*duplicate2*
duplicate2	*x*
e	*y*
f	*z*

Then the program *ef* would report that the lines, *duplicate1* and *duplicate2* where in both files. The program *ef* can detect similarities as small as one line. It would also output the entire file with a '>' in front of each line that was identical in both programs. We tested our methods on known files to be certain that they worked, and also repeatedly checked our work. In this way the hundreds of thousands of lines of code in the various files could be processed. Note however that this is a character by character comparison. We have not attempted to read each program looking for similarities, but rather to find exact matches, where someone took code from Unix and put it into Linux. Given that the comments could be identical and the code different we also wrote programs that would

simply compare the comments in the two sources and the code in the two sources. We then performed this comparison on all files which appeared to have similar functionality. We then flagged any similar code and examined the files to determine the similarity. Often it would turn out that the code that compared came from Berkeley or was in the public domain. In the case of one line matches we discarded matches for **C** idioms like "*c++ "or "include <stdio.h>" additionally we discarded multi-line matches which were clearly not due to copying. We then looked at the remaining lines. If there was a match we examined the match and determined whether it warranted further investigation. Here if the code looked like it was a **C** idiom of the sort described above or appeared not to be relevant we skipped it. Otherwise we examined the code to determine if there was substantial similarity in the structure. If there was, we reported this. Additional we reported any other substantial matches. By this means we found two kinds of copying, the first where they were a number of lines which were identical, on a character by character basis, and those where the code was different, but where the code or its structure were substantially similar. The report attached at the end of this memo shows the similarities which we found.

Additional we investigated the settlement of The Regents of the University of California and BSDI. It is my understanding that anything in BSD Lite tape which was distributed by the University of California, is free of any legal encumbrances from SCO. Further any code which is necessary to meet the POSIX standard is also free of encumbrances. This is a simplification of the settlement but is a valuable way to look at the conditions of the settlement. The site www.cdrom.com has files it claims are the BSD Lite distribution. We have compared these files with those listed below. Those files with a "*" next to their name have similar code in the BSD Lite distribution. Assuming that the files on this site are the ones referred to in the agreement, then it is my understanding that according to the settlement agreement they would be exempted.

Given all of this, and subject to the further analysis of Mike Davidson, I have reached the following preliminary conclusions. First many portions of Linux were clearly written with access to a copy of Unix sources. This of course would be a violation of the License agreements under which Unix is distributed. Second there is some code where Linux is line for line identical to Unix. This is not entire programs but fragments of code and programs from various areas of the operating system. Thirdly there are also portions of the programs which appear to have been rewritten, perhaps only for the purposes of obfuscating that the code is essentially the same. These techniques also imply that whoever modified the code, did just that because there are few lines which are completely identical. This means that they started with a source file which apparently came from Unix and is thus the property of SCO. We did not look

through the programs to find substantial similarities or structural similarities, and there may be portions of the code which are modification or rewriting of the original Unix source but have not a single line identical. Further it is possible that we missed comparing two files. That we did not find similar modules, or that we did not use the appropriate version of Unix, since Linux could have potentially come from any number of versions of Unix that exist. We used a few representative versions of Unix to do the comparisons. However given the number of versions of Unix, it is quite possible that there could be copying which was not detected because we did not have the appropriate version. Additionally to the extent that the code was rewritten based on Unix or otherwise modified we would not of detected this. I suspect that this is the case in certain instances. However to do a complete review on this basis is a substantially larger task than the one which we performed.

One of the questions which remains to be answered is what is the history of the identical code. It is possible that some of the code came from Berkeley or other third party. It is also possible that the code is exempted by the BSDI/Berkeley settlement. Additionally there are a number of other legal issues. I am awaiting an analysis from Mike Davidson on some of these issues, since he has a better feel for the history of much of this code.

The question now arises how serious are the similarities that have been found. Not being a copyright lawyer I cannot comment on this. The fact however that there are pieces of code which are identical to those in the Unix source and others which appear to be simply a rewriting of Unix code is clearly disturbing. It is also clear that in certain instances whoever wrote the code started with Unix source and modified it. Thus there can no doubt that parts of the Linux distribution were derived from Unix. Additionally in areas where the code is identical for compatibility reasons, the code in certain instances is character by character identical. There is a Grove Press case where the court found that making plates from the pages of an out of copyright book was a violation of law. This practice here may be similar.

I hope I have answered your questions, and look further to discussing this with you further. This a second draft report based on additional information which I have obtained.

Bob Swartz

Program Name	Redhat Filename	Redhat File Location	UnixWare Filename	UnixWare File Location	UW Version
curses	read_entry.c	ncurses	tic_read.c	libcurses/screen	2nd Set
curses	write_entry.c	ncurses	tic_parse.c	libcurses/screen	2nd Set
curses	comp_hash.c	ncurses	tic_hash.c	libcurses/screen	2nd Set
curses	comp_parse.c	ncurses	tic_parse.c	libcurses/screen	2nd Set
curses	comp_scan.c	ncurses	tic_scan.c	libcurses/screen	2nd Set
kernel	acct.h	linux/include/linux	acct.h	usr/src/uts/i386/sys	4.0
kernel	elf.h	linux/include/linux	elf.h	usr/src/uts/i386/sys	4.0
kernel	if.h	linux/include/linux	if.h	usr/src/uts/i386/net	4.0
kernel *	in.h	linux/include/linux	in.h	usr/src/uts/i386/net-inet	4.0
kernel	ip.h	linux/include/linux	ip.h	usr/src/uts/i386/net-inet	4.0
kernel	socket.h	linux/include/linux	osocket.h	usr/src/uts/i386/sys	4.0
kernel *	time.h	linux/include/linux	time.h	usr/src/uts/i386/sys	4.0
kernel	ufs_fs.h	linux/include/linux	ufs_fs.h	usr/src/uts/i386/sys/fs	4.0
kernel	quota.h	linux/include/linux	ufs_quota.h	usr/src/uts/i386/sys/fs	4.0
kernel	elf.h	linux/include/linux	elf.h	usr/src/uts/i386/sys	4.0
libc	gcvt.c	cvt	gcvt.c	port/gen	2nd Set
libc	bitops.h	linux/include/asm	async.h	usr/include/sys	4.2
libc	compat.h	include/pthread/mit/sys	isocket.h	usr/include/sys	4.2
libc *	ftp.h	include/arpa	ftp.h	usr/include/arpa	4.2
libc *	nameser.h	include/arpa	nameser.h	usr/include/arpa	4.2
libc *	syslog.h	include/sys	syslog.h	usr/include/sys	4.2
libc *	tcp.h	include/netinet	tcp.h	usr/include/net-inet	4.2
libc	telnet.h	include/arpa	telnet.h	usr/include/arpa	4.2
libc	ruserpass.c	inet	crypt.c	port/gen	2nd Set
libc	iovfprintf.c	libio	doprnt.c	port/print	2nd Set
libc	drand48.c	misc	drand48.c	port/gen	2nd Set
libc	test_ctype.c	ctype	ecvt.c	i386/gen	2nd Set
libc	_errlist.c	sysdeps/linux	errlst.c	port/gen	2nd Set

List of files and version where similarities were found

libc	tsearch.c	misc	tsearch.c	port/gen	2nd Set
kernel	in_systm.h	linux/include/linux	in_systm.h	usr/src/uts/i386/netinet	4.0
kernel	ioctlp.h	linux/include/linux	ioccom.h	usr/src/uts/i386/sys	4.0
libc *	gmon.c	sysdeps/linux/i386/gmon	mon.c	port/gen	4.0
kernel	nfs.h	linux/include/linux	nfs.h	usr/src/common/uts/fs/nfs	4.2
kernel	elfcore.h	linux/include/linux	procfs.h	usr/src/uts/i386/sys	4.0
libtermcap	tparam.c	libtermcap	tgoto.c	libtermcap	2nd Set
lpr	displayq.c	common_source	displayq.c	usr/src/cmd/lp/lib/bsd	4.0
troff	parse.c	groff/xditview	ta.c	usr/src/ucbcmd/troff/troff.d	4.0
yacc	lex.c	bison	y2.c	as/i386/big_yacc	2nd Set
kernel	sem.c	linux/ipc	sem.c	usr/src/common/uts/proc/ipc	4.2
kernel	msg.c	linux/ipc	msg.c	usr/src/common/uts/proc/ipc	4.2
kernel	read_write.c	linux/fs	osocket.c	usr/src/uts/i386/io	4.2
kernel	shm.h	linux/include/linux	shm.h	usr/src/uts/i386/sys	3
kernel	ipc.h	linux/include/linux	ipc.h	usr/src/uts/i386/sys	3
kernel	debugreg.h	linux/include/linux	debugreg.h	usr/src/uts/i386/sys	3
kernel	stat.h	linux/include/linux	stat.h	usr/src/uts/i386/sys	3
kernel	utsname.h	linux/include/linux	utsname.h	usr/src/uts/i386/sys	3
kernel	if_arp.h .c	linux/include/linux	if_arp.h	usr/src/common/uts/net/tcpip	4.2
uucp	dialHDB	contrib	dial*.c	usr/lib/uucp	Open Server

Comment: </TDf