Subject: copy protection (LONG)
Posted by gus on Wed, 13 Nov 1985 00:49:53 GMT
View Forum Message <> Reply to Message

Article-I.D.: Shasta.1530
Posted: Tue Nov 12 19:49:53 1985
Date-Received: Thu, 14-Nov-85 08:18:54 EST
References:
Distribution: net
Organization: Stanford University
Lines: 302

```
>>  Here are some program cracks for various Macintosh applications.
>>  ...etc.
>>
>    [ Suggestions for beating copy protection schemes were here ]
>
>   Hmmmmm.  Seems to me that this kind of thing is slightly innappropriate
>  for the net.  Besides, Sonny would never condone it.  And by the way,
>  there is no need to 'beat' "Griffon Terminal"(sic), since it isn't copy-
>  protected any more.
>
>  Scott Gillespie
>  Reed College
>
```

I don't see why the original article was inappropriate. True, I did expect
it to cause some conmtroversy.

net.anything.anything is a public forum where people can share information
and help solve other people's problems. Copy protection is a problem which
must be solved. Many programs are very much hampered by the mindless
placement of copy protection, presumably to hamper piracy. Developers who
use copy protection are only fooling themselves. Judging from experience of
recent months, it will take at most one month before someone (usually
Central Point Software) has defeated the protection system. Within another
month, anyone who would care to have a copy of the latest Copy II Mac
already has it. (Jazz lasted 2 weeks!)

From then on, the copy protection no longer protects the vendor, it actually
hurts him by
1) Making the program less usable
2) Generating animosity on the part of users toward the company
3) Making programs more fragile

I will now take these three points individually

1) Making the program less usable.

Programs are made less usable in at least three ways

1) Inability to fully use a hard disk
2) Inability for expert users to customize their environment
3) The high probability that protected programs will not be compatible with system upgrades.

The current trend in Microcomputers is that more and more people are using hard disks. This problem has perhaps affected IBM PC owners much earlier than it has Macintosh owners since the XT has always had a hard as standard equipment. Developers who still insist on protecting their programs have resorted to one of four methods.

1) Not allowing hard disk use at all
2) Configuring the program for a specific configuration when the program is transfered to hard disk
3) Requiring that a "key" disk be left in the floppy drive.
4) Writing non-standard information on the hard disk itself.

Choice 1 is clearly unacceptable as it defeats the entire purpose of a hard disk.

Choice 2 is unacceptable because the program becomes useless when hardware is upgraded. This may not be apparent to a user (not mentioned in the documentation, or burried in fine print) until it is too late.

Choice 3 is unacceptable in that it still requires use of floppies when otherwise, the system would be self-contained. This is only agravated by recent schemes which only require the key disk after 'n' invocations. A user, then, who did not originally install the program could suddenly get a weird 'please insert disk' message at any time which he did not know anything about!

Choice 4 is unacceptable because other files besides the protected program also live on the hard disk. This means that any bad marks would interfere with the work of programs such as hard-disk restore and backup programs which assume that the disk is correctly formatted. (or want to make it that way!) Also, this scheme assumes direct access to a hard disk and the file system. Although this may work for IBM PC's, this, in general, will not work in a Macintosh environment where a hard disk is a separate subsystem which varies from vendor to vendor.

The Macintosh environment is a rich one allowing for many variations in which individual users work with their machine. One of the most common variations is deciding what fonts and desk accessories reside on the system disk. A copy protected program makes these modifications a risky business at best since it requires that such modifications be made on the master disk. Programs such as switcher are made extremely clumsy to use when swapping between two or more copy protected programs. Programs such as RAM disks are un-usable since it is usually the applications, and not the data files that you want to place on a ram disk. Disk cache programs usually flush the entire RAM resident disk block information as soon as a disk is ejected, even if only temporarily in order to insert a key disk. While many of these problems are circumventable, they do pose a hefty inconvenience to the user.

Copy protection systems often have to make assumptions about underlying hardware and system software, which would otherwise be shielded by the underlying operating system. Thus when the system is updated, many of these assumptions no longer hold true. This situation occurrs every time Apple updates the Apple II ROM's. The first programs to be incompatible are those that checksum the ROM and thus fail. This creates large inventories of un-usable software on dealers' shelves. While incompatibilities may exist for other legitimate reasons, a large number are caused by copy protection.

This situation is about to occurr again as Apple is releasing the new hierarchial file system. Pat Dirks of Apple Computer spoke at the Stanford Mac Users Group Developers Subgroup meeting last Thursday (11/1) to talk about the HFS. He brought with him a list of the applications that Apple had tested for HFS compatibility. A large number of those that had failed, and certainly, the single most important reason, was copy protection.

2) Generating animosity on the part of users toward the company

This is largely a subjective issue. This has largely to do with prople not buying products from a particular company simply because those products are copy protected. Certainly copy protection cannot help a software review. Most recent reviews devote at least a few sentences to the way a program is protected, and may even go into great length describing how the protection impeded their worek with the product. I am glad to have had a part in improving this situation in at least one case. The fact that the current version of the TMON debugger by TMQ software is un-protected is a direct result of a review that I posted on net.micro.mac several months ago. Strong opposition from the early alpha testers of the Reed College/Metaresearch Rascal package caused all versions after the initial release to be unprotected. Manx software has finally decided to do away with their copy protection in the next release. They also publicly announced that they would provide a patch over the phone to all current users describing how to defeat the existing system. Many companies are advertising the fact that their program is not copy protected. The list goes on and one about case studies where it is (or was) simply to the companie's dis-advantage to copy protect the software.

3) Making programs more fragile.

Any reasonable manual for a major business software product suggests that
you back up your data disks regularly. This is good and sound advice. They
don't, however, suggest that you back up the master disk, for obvious
reasons. Everyone who has been around computers for a while knows theat
magnetic media (disks) can ware out or otherwise lose valuable data.
Copy protected disks are subject to this same problem. Most companies have
a disk replacement policy which allows you to obtain a replacement for
a bad disk. This generally requires that you mail in the old disk, along with
a fee, to obtain a single new backup disk. In the meantime, you have to
wait until the new disk comes back. Some companies even provided an extra
backup disk with the same information as the original so that you still
have at least one working copy while the other is 'out for repair.'
Unfortunately, hardware errors are such that when one disk becomes unusable,
there is a good chance that this may occurr on another disk. This is
compounded by the fact that many programs WRITE to the protected master
disk. This subjects the disk to destruction due to software errors, which is
a much more common problem. Finally, the "two disk" packaging is generally
available for the more expensive software packages. Medium and low-cost
programs generally do not have this 'feature.'

There is one instance, however, where copy protection is permissable,
however, and which may actually INCREASE the value of the product. This is
in the case of games. Generally, games do not require back-up dtat disk,
and do not make effective use of a hard disk, nor do they really need any
extra utilities such as disk caches and ramdisks. The reason why copy
protection is desireable in a game, however, is that when you buy a protected
game, you buy two games for the price of one! The first game is the one
shown on the front cover. The second is trying to break the protection
system!

The analogy between games, especially adventures, and protection
systems goes a very long way. Copy protection has gone far beyond simply
making a program un-duplicatable by normal means. Those people installing
the protection system have to guard against attacks from various directions.
The first is bit copy programs such as Copy II Mac. In this case, the
solution is to create stranger and stranger disk formats which are not
correctly interpreted by the latest version of the Central point product.
The second direction of attack is simple "crack" lists like the message that
prompted this reply. This is solved by requiring such 'cracks' to be
extremely long by providing multiple checks and checksums in the code. The
third direction comes from user who would go in and remove the copy
protection themselves.

This is where the true 'game' quality comes in since at this point, copy protection becomes a duel between two expert programmers: the one creating the protection system and the one removing it, just like a plyer tries to solve an adventure created by its author. In both cases, there is a guaranteed solution, but that solution may be arrived at by gathering clues, and making logical deductions to solve puzzles. In the case of an adventure, the 'solution' is carefully planned out by the author so that there are just enough clues to make the game possible to win at. In the case of protection systems, the "solution" is the path that the program takes when running from the original disk. If this did not exist, the program would never run. Both can have red harings and traps to confuse and slow down the player. In each case, it is the author's job to make the solution as difficult to arrive at as possible.

No copy protection system is un-beatable. This has been born out time and time again by so-called 'uncopyable' programs. No matter how many levels of encryption there are in the code, there is always a solution. It becomes a matter of pride for someone removing a protection system to complete his work and not give up in midstream. Such system can thus only thwart the efforts of novices but not of experts. Both sides have certain advantages. While the creater has the advantage of having the full source code available to him, and knowing the state of the machine at all times, the defeater has the second law of themodynamics behind him, in that it is easier to destroy something than to create it. The defeater exploits weaknesses in the system that he might find that the creater might not have thought of.

By this time this issue of copy protection has transcended all issues of piracy, which after all, which is what protection is supposed to prevent. Whatever the copy protection defeater's motives, he is certainly not going to change his mind about anything after he has defeated the protection system. Only his morals can decide that.

It thus may or may not be easy to thwart the efforts of would-be pirates by both making the program un-copyable Copy II mac, and by requireing that any patch be extremely long, but is impossible to thwart the efforts of those who have set their mind to defeating a protection system. Thus, there comes a point of diminishing returns after which ANY level of protection is no longer helpful, if not detremental because of the increase in code size required to support the copy protection, the time that it takes to execute it and perhaps most important for companies involved, the hours required to create it. This may translate into several thausands of dollars or many man-hours of employee time, depending on whether the protection system is done in house or out.

If the situation persists in the direction that it is going today, copy protection will become a more and more expensive proposition. It has proven easy to defeat the protection systems of programs which have used similar methods. Once one is defeated, the others become much easier, if not trivial to break. It has also proven easy to defeat systems which were applied by someone not directly involved in the creation of the program. "Drop-in" protection systems which make few assumptions about the host program are as easy to remove as a parasite which clings to its host by only a few points

instead of getting inside. Companies will either decide that enough is
enough and that copy protection is simply not worth all of its shortcomings,
or they will simply pass the higher and higher cost of such protection to us
users.

Hardware protection systems that use un-copyable 'keys' which are hooked
in to the computer through one of various i/o ports solve the "fragile disk"
and hard disk problems by making the key uncopyable instead of the disk. All
other problems still exist. Additionally, the increased cost of the key and
"key ring" may make this scheme impractical for all but the most expensive
software packages.

So far, I have only said negative things and have provided no positive
ideas. I believe that I ought to end this message with a positive suggestion;
an idea which software publishers may not have thought about as an
alternative to "hard" copy protection: Soft protection. I will write this
idea in Macintosh terminology, but the general scheme should be applicable
to other machines as well. Here is how it works.
Lets say that you want to publish a program called "Fantasic". There will be
two files on the disk: "Fantastic" and "Fantastic Copyright notice." Both
are executable programs. Both are clearly visible on the desktop. Niether
is copy protected, and there are no hidden keys on the disk. The program is
totally copyable and movable to a hard disk. The master disk need only be
used once, and then put away, except for emergencies. The "Fantastic" program
is the main program to be run. The "Fantastic copyright notice" program
simply displays a single screen stating that this program is copyrighted and
that it is illegal to copy it for anything other than your own personal use,
you being the original purchaser. The screen also describes the material
advantages of owning a legitemate copy, such as customer support and free
or low-priced updates. Now here is the interesting part: "Fantastic"
will NOT run unless "Fantastic copyright notice" is also on the
disk, and in the same folder as "Fantastic." This, program-wise, can easily
be done. I will not go into details how. Just suffice it to say that it
would take only a trivial amount of code. (Much less than the text of the
copyright notice.) The theory behind this scheme is that every time that you
launch the program from the finder, you also see the copyright notice
program. You may choose to ignore it, but it is always on your disk nagging
at you if you indeed have accepted a pirated copy. Some people will never
accept or give out pirated software. You don't have to worry about those.
Others will never buy something tat they can otherwise get a hold of by
copying. There is not much you cando about those because any protection
system will be broken sooner or later (but probably sooner). The people that
you need to worry about are those in-between folks who are undecided on the
issue. These are the people who you could gently sway or nudge in your
direction. The general philosophy is to get people with a carrot instead of
a stick.

This scheme is certainly not foolproof, technically.
A user may launch "Fantastic" from either the minifinder, or may
open a "Fantastic" document. and thus not be "reminded" by th

e copyright-notice file. However, requiring that it be in the
same folder as "Fantastic" would probably make the user see it
often enough assuming that much of the time that the Mac is used is spent
in the finder.

Nothing should be done to antagonize the user. This includes placing
page after page of threatening legal text in the copyright notice file
(a la smoothtalker) or requiring that the notice be read after every n'th
invocation of the program. Everything should be done to give the impression
that the publisher is there to help legitemate owners.

This must certainly be considered an experimental solution. I make no
guarantees that it will either help or hinder sales compred to a completely
protected or un-protected program. However, like most experiments, they
must be tried out to see if they work.

Copy protection is a sticky issue in today's software market. Claims by
software makers that it increases sales by disallowing illegal copies are
dubious. Meanwhile, users have to make due with a program hampered to
various degrees by it. Protection systems themselves, as they become more
sophisticated, become games played between the creators and defeaters of
protection systems. On the other hand, there are alternatives to copy
protection which may achieve the same results in less brute force ways.

I will be speaking on copy protection at the next Stanford Macintosh
Users Group Developers meeting next Thursday, November 21, in Polya 111,
(Turing auditorium) on the Stanford campus at 7:30pm.
Anyone in the San Francisco Bay area is welcome to come.

Gustavo A. Fernandez