DRM and GPLv3

Proposed changes in GPLv3, for addressing Digital Restrictions Management
With comments from Richard Stallman and Eben Moglen

Compiled by Ciaran O'Riordan, FSFE

This document shows the changes which are proposed for version three of the GNU General Public License which deal with Digital Restrictions Management (DRM). It includes public comments made by Richard Stallman and Eben Moglen.

Published by Free Software Foundation Europe

Table of Contents

About this document 1

Defining complete source code 2

Eben Moglen speaking at the GPLv3 launch, January 16th 2006 2
Richard Stallman, speaking in Brussels, February 25th 2006 2
Richard Stallman, speaking in Turin, March 18th 2006 3

Permission to circumvent restrictions 6

Eben Moglen, speaking at GPLv3 launch, January 16th 2006 6
Richard Stallman, speaking in Turin, March 18th 2006 6

Closing comments 7

About this document

This document is published by FSFE to aid understanding of the proposed changes to the GNU General Public License (GPL) with respect to Digital Restrictions Management (DRM). It quotes version two of the GPL, the first published draft of version three of the GPL, Richard Stallman (the author of the GPL), and Eben Moglen (the legal consel of FSF).

The above people had no part in the compilation of this document or the selection of material for inclusion. Ciaran O'Riordan was responsible for those activities.

Defining complete source code

GPLv2	**GPLv3, first draft**
(no corresponding section in version two)	Complete Corresponding Source Code also includes any encryption or authorization codes necessary to install and/or execute the source code of the work, perhaps modified by you, in the recommended or principal context of use, such that its functioning in all circumstances is identical to that of the work, except as altered by your modifications. It also includes any decryption codes necessary to access or unseal the work's output. Notwithstanding this, a code need not be included in cases where use of the work normally implies the user already has it.

Eben Moglen speaking at the GPLv3 launch, January 16th 2006

http://www.ifso.ie/documents/gplv3-launch-2006-01-16.html#em-auth-codes

I would translate that for vernacular use, into "plays all the same movies".

Richard Stallman, speaking in Brussels, February 25th 2006

http://www.ifso.ie/documents/rms-gplv3-2006-02-25.html#tackling-drm

[GPLv3] doesn't limit what technical jobs the software can do, because that's another principle of free software, people should be able to use the software and run it for any purpose, and change the software to do whatever they like. So we have not said "you can't change this software to do the jobs that you would do if you were trying to restrict people". Instead, we focussed on a different aspect of DRM, which is, stopping the users from controlling the software that runs in their machine. Treacherous Computing is designed such that if you modify the software, it won't be able to do the job.

They use things like checksumming the software and checking whether it has been signed and authorised, and so if you modify the program and you install it, since your version hasn't been signed by them, your version isn't really authorised, so either it won't run, or it won't be able to open the files that you want to open or the network server will refuse to talk to it, or in one way or another you will be blocked from really doing the job that the original version was set out to do.

So what we're doing in GPL version three is, we're saying they're welcome to design free software to do whatever it is they want, and they're welcome to set up the machine such

that it won't run a program unless it's been signed, but they have to give you the signature key so that you can sign your own version.

They must give you the signature key so that you can authorise your version at least to run on your machine.

Now, if each machine has a different signature key, they only have to give your signature key to you. Your machine's signature key. They have to give it to you, they don't have to publish it, they don't have to give it to anyone else. They can even promise you that they won't give it to anyone else, but they have to give it to you.

So this ensures that you really have, practically speaking, the freedoms to modify the program, put it into the machine, and have it really run and it must be able to do the same jobs on the same data, which means it must be able to access the same files that the unmodified version would have accessed, or talk to the same network server that the unmodified version would have talked to.

So whatever the program needs to be able to give the network server so that the network server will talk to it, they have to give that to you.

[An audience member asks if this will stop distributions from distributing signed binaries]

Not at all. When the purpose is to prove this is "my" version of the binary and it was not changed by someone else, that's a completely different scenario because they're not giving you a machine that refuses to run the software unless it's been signed by them ...and therefore there's no harm in that. So, for instance, if you want to distribute binaries or sources of your program, and sign them so that people can tell they're your authentic versions, you're still free to do that, and they can still get the corresponding public key and check your signature.

The reason that these requirements in the GPL don't apply in this case is that, in this case, the user checks if he wants to. It's not a machine that's checking it and refusing to run the software if it isn't signed by you. If you give them a machine that won't run the program unless the binary has been signed, then you have to give them the signature, but if you just sign your binaries or your sources and say "check them if you wish", the requirements don't apply in that case. The requirements say you must include whatever keys or codes are necessary to authorise a modified binary so that it will function the way the original binary would. If it isn't needed to make the modified binary function, there's no requirement.

This is an issue that we worked on quite a bit, figuring out how to block the corruption of our software for Digital Restrictions Management without limiting the technical features it can have. So instead of attacking the technical features of DRM, we attack the thing that makes DRM evil, which is the fact that it has been taken out of the user's control. So we thwart DRM by insisting that the user must retain the control of the machine and as long as you respect that, you can program the software to do whatever you like.

Richard Stallman, speaking in Turin, March 18th 2006

http://fsfeurope.org/projects/gplv3/torino-rms-transcript#drm-and-laws

Freedom zero says you are free to run the program as you wish for any purpose. We are not limiting freedom zero. If someone wants to run a program to encrypt something, that's fine. If someone wants to run a program to decrypt something, that's fine. If somebody

wants to run a program to produce an encrypted medium that's difficult to access, that's fine. If somebody has some other GPL covered program to access that media and he wants to run it to access the encrypted data, that's fine too. And distributing software that could be used for those purposes is also entirely permitted, and will be permitted by GPL version 3.

However, freedom zero does not include imposing your purposes on someone else who is going to run the program, because his freedom zero is the freedom to run the program for any purpose of his. So, there is no such thing as the freedom to use any software to impose your purpose on someone else. In fact, that should be illegal. I'm serious. And that's what DRM is.

When somebody distributes a player, that has DRM in it, what he's doing is trying to restrict your running of your computer for his purposes, which is directly in conflict with the four freedoms that you should have.

And that's what GPLv3 is in certain ways trying to stop and it does this simply by assuring you all four of the freedoms when you use the software. You see, because DRM - Digital Restrictions Management - is a plan to restrict the public, anyone distributing a version of a GPL-covered program as a player for DRM media always does something to stop the public from modifying the player, because his purpose in distributing a DRM player is to restrict you, he has to make sure you can't escape from his restrictions, from his power. That means he is always going to try to deny you freedom one. Freedom one is the freedom to study the source code of the program and change it to do what you want. What you want, might be, to escape from his restrictions, and if you have freedom one, you can escape from his restrictions. So his goal is somehow or other, for practical purposes, to deny you freedom number one.

Now, what he might do is, use non-free software, and then completely deny you freedom number one. In fact, that's what they usually do. We can't change that with the GPL because they're not including any GPL-covered code. They don't have to pay attention to the GPL. There should just be a law against it. It should be illegal. DRM should be illegal, but we can't change laws by modifying the GPL.

However, there are those that want to use GPL-covered software for this purpose, and they want to do so by turning freedom number one into a sham, a facade. So they plan to do something like, make a modified version of the GPL-covered program, which contains code to restrict you, and distribute that to you and somehow arrange that you can't really modify it, or if you modify it it won't run, or if you modify it and operate it, it won't operate on the same data.

They do this in various ways. This is known as Tivo-isation because this is what the Tivo does. The Tivo includes some GPL-covered software. It includes a GNU+Linux system, a small one, but it does, and you can get the source code for that, as required by the GPL because many parts of GNU+Linux are under the GPL, and once you get the source code, you can modify it, and there are ways to install the modified software in your Tivo and if you do that, it won't run, period. Because, it does a check sum of the software and it verifies that it's a version from them and if it's your version, it won't run at all. So this is what we are forbidding, with the text we have written for GPL version three. It says that the source code they must give you includes whatever signature keys, or codes that are necessary to make your modified version run.

In other words, it ensures that freedom number one is real. That you really can modify the source code, install it, and then it will run and not only that, we say, they must give you enough to make the modified version operate on the same range of data. Because, you see, Microsoft's plan, which they call Palladium, and then they change the name - they change these names frequently so as to evade criticism, to make criticism difficult, to make any kind of comment on their plans difficult. You talk about their plan and they say "Oh, we've dropped that, we have a different plan now". And probably it is different in some details, but the point is that they generate encryption and decryption keys using a check sum of the program which means that a different program can't possibly access the same data. Although, that's just the base level, and then on top of that they implement other facilities where the program simply has to be signed by the authorised signer in order to be able to access the data.

Well, GPL version three says that if they distribute a GPL-covered program in this way, they must provide you with the key necessary so that you can sign your version and make it access the same data. Otherwise, they would say "Yes, you can run your modified version, but it will have a different check sum, so your version will only operate on data files made for your version, just as our version only operates on data made for our version". And what that means is that all the available files will only work with their version and your changed version will not be able to access them. That's exactly, in fact, how Treacherous Computing is designed to work. The plan is that they will publish files that are encrypted and it will be impossible to access those files with any other program, so GPL version three is designed to ensure that you really, effectively, get the freedom to take the program you were given, modify it, and run the modified version to do a different thing on the same data on the same machine.

Permission to circumvent restrictions

GPLv2

(no corresponding section in version two)

GPLv3, first draft

No covered work constitutes part of an effective technological protection measure: that is to say, distribution of a covered work as part of a system to generate or access certain data constitutes general permission at least for development, distribution and use, under this License, of other software capable of accessing the same data.

Eben Moglen, speaking at GPLv3 launch, January 16th 2006

http://www.ifso.ie/documents/gplv3-launch-2006-01-16.html#em-not-a-tpm

That is to say, distribution of a covered work as part of a system to generate or access certain data, constitutes general permission, at least, for development, distribution, and use, under this license, of other software capable of accessing the same data.

In the United States, this language, we believe, has specific consequences with respect to the Digital Millennium Copyrights Act. We wish to point out that no GPL'd program can be regarded as a measure in circumvention of any other GPL'd program's access protection schemes. We believe that this language will also provide some assistance in achieving similar results under statutory enforcement schemes in pursuance of the EUCD and other international regulation meant to assist disablement of users.

Here again we are simply speaking to courts to explain how we understand the intent of licensors. We have no power to change local law, but we do have power by giving permission to make clear where our permissions should not be mis-read under local laws that presume user disablement from mere technological existence.

Richard Stallman, speaking in Turin, March 18th 2006

http://fsfeurope.org/projects/gplv3/torino-rms-transcript#drm-and-laws

There's one other way that we're trying to thwart DRM. You see, one thing they do is, some countries, including, I'm sad to say, this one, have adopted unjust laws that support DRM. The exact opposite of what they ought to do, which is prohibit DRM, and what they say is: when media have been encoded for DRM, then writing another program to access that media is illegal, and the way they do this is they say that DRM constitutes an effective, they call it "protection" I call it "restriction", measure. So, what we say is, by releasing a program under GPL version three, you agree that it is not an effective restriction measure. In other words, you authorise others to develop on their own software to read the output of your program.

This also is a matter of recognising and respecting their freedom to develop software and use their computers.

Closing comments

A year-long consultation is being held to spread awareness of the proposed changes to GPLv3, and to solicit comments. This document was produced as part of FSFE's efforts to assist this process. For more information about FSFE's efforts, see:

http://www.fsfeurope.org/projects/gplv3/

The official website of the GPLv3 process is:

http://gplv3.fsf.org

Free Software Foundation Europe e.V.
Talstraße 110
40217 Düsseldorf
Germany
Phone: ++49 700 - 373387673 (++49 700 FSFEUROPE)
European office e-mail: office@fsfeurope.org