Secure Downloads - Stopping users from downloading files to prevent copying

Most people associate the action of downloading a file to be making a copy of the file - and they are right! And in the world of Intellectual Property Rights (Copyright) we always talk about controlling making copies. But are these two things incompatible? This white paper discusses some of the reasons why downloading is necessary and the impact if it is stopped, and then concludes by looking at secure downloads, achieving a cost-efficient and secure approach.

As is so often the case in IT, it depends on what is actually happening at the technical level that decides success or failure. (Lessig's Law says what you do is decided by the programmer.)

Let's take the case of information being viewed in the browser.

Everything you view in a browser is automatically downloaded to a cache on your hard disk. The Temporary Internet Files (or cache) folder is used by browsers to store webpage content on the computer hard disk for speed up viewing. So the cache lets the browser download only content that has changed since you last viewed a webpage, instead of downloading all of the content every time the page is displayed. That makes downloading much quicker!

Now you can use ASP code to disable caching of web pages, but often this code fails to work as predicted because browsers are regularly updated, and new ones sometimes just don't obey or understand the code. This has been especially true of JavaScript 'fixes' which often work with one version of a browser but not another. So trying to stop browser caching as a security measure to prevent file downloading is unreliable and it is best not to rely on this method of 'protection'.

Depending on the information you are trying to protect, however, you might not want to stop it anyway. There are big overheads if you open a PDF on a server and then smooth scroll down the page with every single line of pixels being sent over the network. If you did this there could be major server overheads or response time problems, or likely both.

So if downloading is to solve a performance problem, why is controlling downloads such a problem?

The Internet was made to facilitate sharing information, so it is difficult to stop people from being able to download files, especially if you want some people to be able to download them, but not just anybody or everybody.

If you have your information on a server in an unprotected form, there are several free tools available (do an Internet search on 'download website') that will transfer a publicly accessible web site onto someone's hard drive, where they can be examined at leisure. So if you want to stop illegal downloading you will need to keep the information you want to protect in an encrypted form so it is of no use to anyone without information from you, or you put it on a

server that is not publicly accessible. But then you must be sure that the recipient is not going to misuse it.

There have been some ingenious approaches to try and solve this problem. Some have involved providing a one-time access to a server that is not publicly accessible, so the link can only be used while the approved download takes place. Others of uploading the file to be downloaded onto a temporary location that expires when the download finishes. And finally using specialist downloader applications that combine information in order to 'make' the downloaded file during the process, and controlling access to this downloader. And we must remember password access areas, although passwords may be given away and are difficult to manage.

Of course, none of these will prevent the recipient, once they have downloaded the file(s) from then passing them on, or uploading them to one of the torrent download sites. It may have been a lot of work for nothing?

So if you want to prevent other people from making your files available for illegal downloads you have to do something more to protect them. That will involve using encryption since that is the only really effective tool to stop people from using files they have got hold of when they should not. An encrypted file is no use to anyone without the software to decrypt the file (it need not be in a 'standard' format like OpenPGP), and also the keying information to go with it.

Now this all starts to get a bit complicated. Users have to be 'authenticated" in some way, keying information has to be given to them secretly (if they know what the information is they can give it away just like they could give away the unencrypted files). Also the keying information must not be part of the protected document or it can be too easily recovered and all protected documents compromised. Users also have to be prevented from being able to make uncontrolled copies or they can still compromise the system.

LockLizard has implemented a number of controls that prevent misuse of downloaded files.

Firstly, files are encrypted.

Secondly, and most importantly, decryption keys are not part of the downloaded file, so the system cannot be attacked through the key mechanism. Decryption keys are securely and transparently relayed to a keystore that is locked to individual computers so a keystore will not work if copied to another computer along with the encrypted files. This ensures that users cannot share encrypted files with others as they will only work on authorized computers.

Thirdly, LockLizard products only decrypt content in memory, so that there are no temporary files left lying around with unprotected information in them for someone to copy. So whilst you cannot stop the encrypted files from being copied, they are of no use to anyone but the authorized user.

LockLizard products therefore retain the efficiency of downloading without any loss of security and control over the downloaded file.