A Taxonomy of Methods for Software Piracy Prevention

Gareth Cronin

Department of Computer Science, University of Auckland, New Zealand

gareth@cronin.co.nz

Abstract

The illegal copying of software - usually known as "software piracy" - is a major concern to the software industry, who estimate their losses due to piracy at over $10 billion per year. A wide range of theoretical and practical methods have been developed to prevent such piracy. A taxonomy of these methods provides a tool for increasing understanding of the ways in which piracy is currently addressed and directions for future development. This paper outlines a taxonomy based on the fundamental categories of ethical, legal and physical piracy controls, and provides examples to justify this classification.

1. Introduction

Software piracy is the act of making unauthorised copies of computer software. This paper will only consider software piracy where it is performed for profit. Consideration will not be given to computer users who “share” software for no financial reward (although an implicit reward exists in avoiding the purchase price of such software). The practice of reverse engineering and recompiling software then claiming the work as one’s own in its entirety or as part of a product will also be excluded from discussion.

This paper will survey some current methods of piracy prevention and attempt to classify them into a taxonomy of piracy prevention. While detailed explanations of the technical measures are beyond the scope of this article, an overview in the form of an example of each type of prevention is given. Such a taxonomy is useful to researchers selecting forms of piracy prevention to use, or for consideration of new piracy prevention systems.

Piracy controls may be ethical (moral-based), legal or technical. Piracy controls usually take account of the software manufacturer's desire to cause legitimate users as little inconvenience as possible (Jakobsson & Reiter 2001, Merkle 2002).

Piracy prevention systems target one or more parts of what I will term the "pirated software supply chain". The chain begins at the software supplier, whose goods are obtained by a pirate who duplicates them, and sells them to illegitimate users who may or may not be aware that the software they are purchasing is pirated. The chain becomes more complex when consideration is given to pirates who buy previously pirated software and duplicate this for their own profit.

2. Background

The value of global packaged software sales in 1997 was around $US135.4 billion, however, it is generally believed that the value of these sales and the associated benefits of increased employment and higher tax revenue could be much higher (Contributions of the Packaged Software Industry 2002). Global piracy is a reality:

> **"Estimated 1997 piracy rates for one software market segment, PC business software products, range from a low of 27 percent in the United States to as high as 98 percent in Vietnam. International Planning and Research (IPR) estimates the market value of illegal copying of PC business software at $11.4 billion worldwide in 1997." (Contributions of the Packaged Software Industry 2002)**

3. The Taxonomy

As mentioned earlier, the top level of the taxonomy yields the classes of legal, ethical and technical anti-piracy measures. Each of these classes has one or two associated *means* of preventing piracy. Following on from these means, the methods can be more finely separated, to the point where each example system for piracy prevention can be assigned to a single category. The diagram below presents this taxonomy.

Class	Means	Sub-categories	Sub-sub-categories	
Ethical	Making piracy morally unappealing	Amnesty		
		Appeal		
		Shareware		
Legal	Fear of consequences	Copyright		
		Patents		
		License agreements	Compulsory audits	
			Duplication restrictions	
Technical	Increasing difficulty of duplicating software	Obfuscation	Media obfuscation	
			Code obfuscation	Dynamic obfuscation
				Static obfuscation
		Encryption	Code encryption	
			I/O encryption	
		Simple checking	Dongle	
			Registration	
			Guards	
		Tethering		
	Increasing likelihood of being caught	Observation		
		Watermarking		

Figure 1. The taxonomy.

4. Related Work

Much research into piracy has focussed on how to manage the distribution of digital media, such as video, music and documents while retaining fair financial compensation for authors and publishers. This is an area often referred to as "Digital Rights Management" (DRM). A number of useful ideas that can also be applied to software have been raised. Current DRM systems include such technologies as digital signatures, fragile watermarks, serial numbers and traitor tracing (Bechtold 2002).

Lee and Kim suggest that piracy prevention can be classified as "copyright protection" or "copy protection", where the former is a measure to prove ownership in case of a dispute and the latter a barrier to piracy itself (Lee & Kim 1999).

Zamparelli cites three basic ways to prevent copying, these being *copy protection*, *copy identification* and *copy dissuasion*. He considers *protection* to be the encryption of the material such that it will only work on the purchaser's machine, *identification* as the watermarking of software, and *dissuasion* as the "indelible"

attachment of sensitive personal information to the software that the purchaser would not want to share through duplication (Zamparelli 1998).

5. Legal Measures

The law prevents piracy by creating a fear of the consequences that being caught pirating will bring. Legal systems in the United Kingdom, the United States of America, and countries that have inherited much of their legal systems from these two countries and their earlier origins provide legislation to allow prosecution of pirates through copyright, software patents (Nichols 1999), and software licensing. To impose a fear of consequences, there must be some form of liability for the act of software piracy and this liability must be able to be proven in a court of law. The liability may rest with the pirate, the provider of the distribution channel, the end-user, or a combination of these (Stern 1996).

Historically, copyright evolved out of the printing revolution that followed Gutenberg's invention of the printing press. Copyright law is based almost entirely around this "print paradigm" and its associated concepts of the permanence of a publication and its repeated availability (e.g. a book once purchased may be read many times) (Harvey May 2002).

The distinction between copyright and patents is an important one. In U.S. law a patent is a "legal monopoly granted for the use manufacture, and sale of an invention". On the other hand copyright "applies to a particular, tangible piece of work" (Nichols 1999). Copyright protection is automatic in the U.S. and in New Zealand. It affords the author of a work control over that work and gives them the right to transfer that ownership to another party. A breach of copyright is addressed in the courts, where the plaintiff must prove their copyright over the work in question. A patent holder's monopoly means they can prevent others using their patented device, even if others invent it independently (Nichols 1999).

Copyright and software patents can be considered an effective legal protection where the offender and manufacturer are located in the same country and are therefore subject to the same intellectual property laws. Difficulties arise however where the parties are subject to different legal jurisdictions. The Business Software Alliance (BSA), an affiliation of leading (mostly American) software companies fund surveys of international piracy and acts as a lobby group in

bringing political pressure against countries who do not cooperate with international copyright agreements. The piracy industry in China was virtually closed down due to threats of trade sanctions (Dakin 1997). The BSA's most recent global piracy survey found that the highest rates of piracy occur in Asian and Middle Eastern countries, so the effectiveness of legal piracy prevention is dependent entirely on these countries cooperating with international copyright agreements (Global Piracy 2000).

Software licences are essentially a contract between the user (or users) of a given item of software and the manufacturer or distributor. Most commercial software requires some form of acknowledgement from the user that they have read and understood the terms of the accompanying licence and agree to abide by these terms when they use the software. Many types of licensing are available, ranging from "site licences" that allow all users in a given geographical location or set of network addresses to use copies of software, to simple single-user agreements that prevent any duplication at all (Macromedia 2002, Microsoft 2002).

Some software licences include clauses that require the owner of the software licence to submit to regular audits of their premises to determine their compliance with the licence conditions. While such clauses are not usually found in single end-user licences, an attempt to do so was recently made. Inprise – the manufacturers of the Borland brand of development products (Borland 2002) – inserted a compulsory audit clause in the single-user licence for JBuilder 5 and Kylix 2. After much protest from the user communities of these products the clause was withdrawn (Duchene 2002).

6. Ethical Measures

Pointing out the ethical issues of software piracy to members of the piracy supply chain is another way to counter piracy. Pfleeger suggests that the "right to fair compensation" is a basic principle of "universal" ethics (Pfleeger 1997). If we consider that purchasing software is the only fair compensation for the authors and distributors of that software, then software piracy is in breach of this set of ethics.

One type of ethical piracy counter-measure is an appeal in which a publicity campaign attempts to persuade pirates, distributors or end-users of the error of

their ways. Another measure is an amnesty, where possessors of illegitimate software are encouraged to surrender the goods with no risk of prosecution. “Shareware” may also be considered an ethical counter-piracy control, where the software itself presents an appeal to the user to pay for the products they are using “if they like it and continue to use it”. For example, the software may display a message to this effect each time it is run (MiRC 2002).

The BSA fund frequent advertising campaigns in an attempt to steer public thinking towards the view that software piracy is an illegal and economically damaging activity. These campaigns are in the form of letters to legislators and prominent newspapers, and paid print and media advertising (BSA Creates a Buzz 2002).

The members of the BSA also hold pirate software amnesties. Such an amnesty was recently held in several states in the U.S. (Software Truce 2002).

7. Technical Measures

A large number of ways to prevent software piracy at the technical level have been researched and implemented. A distinction can be drawn between those controls that act to prevent the duplication of software, and those that increase the likelihood of offenders being caught and prosecuted.

7.1 Prevention of Duplication

Technical measures that increase the difficulty of duplicating software can be categorised as obfuscation, encryption and simple checks.

Code obfuscation is the deliberate altering of program code, whether at the source, object or machine code level. The idea is to hide the very purpose of the code, thereby making it more difficult to understand and alter (Collberg & Thomborson 2001). This is a protection against reverse engineering, which may allow a pirate to duplicate more easily by analysing the code for protections against duplication and circumvent these. Obfuscation may be carried out statically, or it may be introduced as part of the executing code (Collberg & Thomborson 2001).

I introduce the term “media obfuscation”, to describe the alteration of the media that the software is distributed on to make it difficult to duplicate using standard

machinery. There exists a wide range of commercial products offering various ways to obfuscate media. The most common distribution media for software at present is compact disc (CD), so most current systems operate by tainting pressed compact discs in some way that is not easily reproducible using a standard compact disc writer. “Laserlock” is an example of such a system. Manufactured by MLS LaserLock International, LaserLock embeds a digital signature during the glass mastering process. A routine is added to the software being protected to check for the presence of the signature. The signature can not be copied by current CD writers (MLS LaserLock 2002).

Encryption techniques include systems where the code to be executed is encrypted in some way and requires the correct key and subsequent decryption to run. Lee and Kim propose a system based on the World Wide Web Consortium’s Public Key Infrastructure (Public Key Infrastructure 2002). Users of software must possess their own unique public key certificate. The software provided by the distributor is encrypted with the user’s own public key. Only the user possesses the private key and is therefore the only person who can execute the software. This system is of course vulnerable to key loss and assumes that all users of a given piece of software have a public key certificate issued by a trusted certification authority (Lee & Kim 1999).

Encryption can also be used to encipher program input and output streams, whether they are streamed to disk files or to other devices. A key is then required to be able to make use of the encrypted matter. Such a system is described by Jakobsson and Reiter where programs that read and write from files as part of their normal operation (such as word processors) encrypt the files using a key that is unique to each version of the program. As the program “ages” and requires updating due to defect fixes and new feature implementations, a new key provided with the update ensures that files created by the new version can not be read by older versions. This disadvantages possessors of illegitimate software in that they lose the ability to read files produced by legitimate users. The authors hope that the system would force pirates to be responsible for providing updates to their customers. This would demand an ongoing relationship between pirates and their customers, making piracy a more risk-filled and less economically viable activity (Jakobsson & Reiter 2001). Tim Budd suggests the use of a “digital battery” in the

form of a smart card that contains decryption algorithms and keys for decrypting media to be used by software. The battery's vendor is the only point where the end-user pays for the media, and the vendor distributes some portion of the money collected to the manufacturers and distributors of the software (Budd 2001). Although his examples are specific to music media and therefore fall into DRM rather than software-piracy prevention, the idea could easily be applied to application software. Prevelakis et al suggest a similar system to Budd, but one that employs an individual *software* agent distributed with each item of media used by the software (documents in this case). The agents communicate with a "billing agent" on the user's computer and manage encryption and decryption (Prevelakis et al 1997).

"Simple checks" include the infamous "dongle", a hardware device connected to a port on a computer whose presence is probed for by a program that will only execute if the dongle is found. Dongles have a reputation for unreliability, inconvenience to legitimate users and high cost. Ralph Merkle suggests a system where a "billing computer" is used for the simple checking. The billing computer communicates with the distributor of the software to discover whether the user has paid for what they intend to use and the software checks with the billing computer before it will run (Merkle 1993).

Another simple check is registration systems, which involve the user of the software acquiring a special unique string of characters and digits that the program demands are entered before it is first run, or at the time of installation (Maña et al 2001). From my own observation, this occurs in most popular business and development software, for example Microsoft Office (Office 2002) and Macromedia Dreamweaver (Dreamweaver 2002).

"Guards" are hardware or software modules that monitor the running program and ensure that it has not been tampered with in any way. Chang and Atallah propose a system to guard against illegitimate modification of code by implementing a network of mutually reinforcing software agents within an executable program. The guards use checksums on key areas of code and on each other to ensure no modifications have taken place and take some action such as crashing the program if a modification is detected (Chang & Atallah 2001)

"Tethering" is the practice of associating a piece of software with a particular piece of hardware. For example, on installation, the program may read the "CPU ID" found on newer processors and record this ID. The program can then refuse to operate on any machine other than the machine it originally recorded the ID from. Microsoft uses a form of tethering branded as "Product Activation". A number is generated based on a hash function that makes use of the computer hardware, most likely the CPU ID or an Ethernet card MAC address if it is present. The number is transmitted to Microsoft and a further non-invertible function is applied to it to create a second unique number. The product will only function when both these two numbers have been entered (Product Activation 2002).

7.2 Increased Likelihood of Getting Caught

The two classes of controls that increase the likelihood of an offender being caught are observation and watermarking. I use the term "observation" to describe the inclusion of monitoring programs in a software package that check whether or not the program is a legitimate copy and report the offence to another party if it is not.

Software watermarking is the hiding of messages in program code. Watermarks may or may not be visible and have varying degrees of robustness to tampering (Collberg & Thomborson 2002). One approach to watermarking as a piracy prevention is to embed a robust and invisible watermark that states the rightful owner of a given piece of software. Applying a unique watermark (unique in that the watermark bears the name of each individual owner) to each distribution of the software is known as "fingerprinting" (Collberg & Thomborson 2001). Later extraction of this watermark can then be used as evidence of piracy in a court case.

Observation is the monitoring by software agents of the state of installed software and the logging and perhaps reporting of this state. Monitoring is currently employed in some commercial software to track users' activities and report them to advertising agencies to aid in profiling audiences and other marketing intelligence. Naturally this practice raises privacy issues as far as legitimate users being unaware what information is being transmitted and to whom, as would any form of monitoring (Spyware 2001).

8. Limitations

This paper has argued that piracy counter-measures may be separated into individual, distinct classes. There is some ambiguity introduced when encryption is considered as a distinct class. Encryption may be used in other classes of piracy prevention, for example by dongles to hide their key that the software requests or by guards to hide their own code.

This paper has considered obfuscation of software code as a protection against duplication, but it has only briefly considered more complex arrangements whereby obfuscation and other controls are combined. Dynamic obfuscation can be used as a form of watermarking, by for example, introducing data structures into the running program that reveal a hidden message (Collberg & Thomborson 2001).

9. Conclusion

While this taxonomy is of the methods for software piracy prevention themselves, it may be useful to also consider a taxonomy of which parts of the pirate software supply chain each method attacks. A taxonomy of the classes of intent behind software piracy (e.g. to make a profit, to impress friends, to anger large companies) and which piracy counter-measures address each type of intent would also be of use when investigating improvements to software piracy prevention from a behavioural perspective.

Although it is beyond the scope of this paper, it would be useful to enlarge the current taxonomy to include “social” and “economic” piracy preventions. This could include the Open Source movement’s philosophy of compelling manufacturers to make source code freely available, thereby nullifying the piracy problem. Consideration of “added-value” measures (such as supplying printed manuals with software) as an incentive to purchase legitimate software would also be an interesting exercise.

It is hoped that this taxonomy can act as a useful tool in analysing current software piracy prevention methods and developing future measures.

References

References with specified authors are listed first, alphabetically ordered by those authors. References with no specified authors are listed alphabetically by title below these.

Bechtold, S. From Copyright to Information Law – Implications of Digital Rights Management. Retrieved Jun 2002 from http://www.star-lab.com/sander/spdrm/papers.html.

Borland, J. (May 2001). "Spyware" piggybacks on Napster rivals. Retrieved Jun 2002 from http://news.com.com/2100-1023-257592.html.

Budd, T. (August 2001). Protecting and Managing Electronic Content with a Digital Battery. IEEE Computer.

Chang, H. & Atallah, M. (2001). Protecting Software Code by Guards. Retrieved Apr 2002 from http://www.star-lab.com/sander/spdrm/papers.html.

Collberg, C. S. & Thomborson, C. (2002). A Functional Taxonomy for Software Watermarking.

Collberg, C. S. & Thomborson, C. (2001). Watermarking, Tamper-Proofing, and Obfuscation.

Dakin, K. (Jan.-Feb. 1997). What if there were no software piracy?. IEEE Software, 14(1), 20 -21.

Duchene, T.J. (January 2002). An Open Letter to Borland/Inprise Concerning Licensing. Retrieved Jun 2002 from http://freshmeat.net/articles/view/369/.

Harvey, D. (May 2002). Circumvention of Digital Rights Management Systems. (Pending publication in Butterworths Tech Law Forum).

Jakobsson, M. & Reiter, M. (2001). Discouraging Software Piracy Using Software Aging. Retrieved Jun 2002 from http://www.star-lab.com/sander/spdrm/papers.html.

Lee, B., Kim, K. (Jan.26-29., 1999). Copyright Protection of Software using Public Key Infrastructure. Proc. of SCIS99.

Maña, A., Pimentel, E. (2001). An Efficient Software Protection Scheme. Retrieved Jun 2002 from http://polaris.lcc.uma.es/~amg/papers/IFIPSEC01-SoftProt.pdf.

Merkle, R.C. (1993). Protected Shareware: A Solution to the Software Distribution Problem. Retrieved Jun 2002 from http://www.merkle.com/protectedShareware.pdf.

Nichols, K. (April 1999). The Age of Software Patents. IEEE Computer, 25-31.

Pleeger, C. (1997). Is there a security problem in computing?. Security in Computing, Chapter 1 1-19.

Prevelakis V, Konstantas D, Morin J-H. (1997). Issues for the commercial distribution of electronic documents. Communications and Multimedia Security, 3, 265-76.

Stern, R.H (Feb. 1996). Bulletin boards and net sites. IEEE Micro, 16(1), 7-9, 70-2.

Zamparelli, R. (December 1998). Digital Distribution Models and Copyright Enforcement. Retrieved Jun 2002 from ftp://ftp.cogsci.ed.ac.uk/pub/roberto/diglib.ps.

–

2000 Global Software Piracy Study. Retrieved Jun 2002 from http://www.bsa.org/resources/2001-05-21.55.pdf.

BSA Creates A "Buzz" About Software Piracy In New York City. Retrieved Jun 2002 from http://www.bsa.org/usa/press/newsreleases/1999-07-15.220.phtml.

Contributions of the Packaged Software Industry to the Global Economy. (April 29, 1999). Retrieved Jun 2002 from http://www.bsa.org/usa/globallib/econ/pwc1999.pdf.

Dreamweaver (Macromedia). Retrieved Jun 2002 from http://www.macromedia.com/software/dreamweaver/.

MACROMEDIA Electronic End-User Software License Agreement.

Microsoft Licensing. Retrieved Jun 2002 from http://www.microsoft.com/licensing/.

Microsoft Product Activation: Frequently Asked Questions General Questions. Retrieved Jun 2002 from http://www.microsoft.com/nz/piracy/mpafaq.asp.

MiRC Homepage. Retrieved Jun 2002 from http://www.mirc.com/register.html.

MLS LaserLock International. Retrieved Jun 2002 from http://www.laserlock.com/main.htm.

Office (Microsoft). Retrieved Jun 2002 from http://www.microsoft.com/office.

Public-Key Infrastructure (X.509) (pkix). Retrieved Jun 2002 from http://www.ietf.org/html.charters/pkix-charter.html.

Software Truce Campaign Hits Halfway Mark . Retrieved Jun 2002 from http://www.bsa.org/usa/press/newsreleases/2001-05-17.561.phtml.

The Free Software Definition. Retrieved Jun 2002 from http://www.gnu.org/philosophy/free-sw.html.

U.S.Software, State Piracy Study. (November 2001). Retrieved Jun 2002 from http://www.bsa.org/resources/2001-11-01.65.pdf.

Borland. Retrieved Jun 2002 from http://www.borland.com.