MULTI MODEL APPROACH AGAINST SOFTWARE PIRACY

Dr.KashifQureshi

ABSTRACT

It is bitter fact that Software piracy is impossible to stop, although software companies are launching more and more lawsuits against Pirates. Originally, software companies tried to stop software piracy by copy-protecting their software. This strategy failed, however, because it was inconvenient for users and was not 100 percent foolproof. Most software now requires some sort of registration, which may discourage would-be pirates, but doesn't really stop software piracy.

An entirely different approach to software piracy, called ***shareware***, acknowledges the futility of trying to stop people from copying software and instead relies on people's honesty. Shareware publishers encourage users to give copies of programs to friends and colleagues but ask everyone who uses a program regularly to pay a registration fee to the program's author directly.

Commercial programs that are made available to the public illegally are often called ***warez* .**

Key Words: Counterfeiting,OEM unbundling,Soft lifting,Hard disk loading,Corporate software piracy,Internet software piracy,Hardware locks, Water markers, embedding a unique identifier, watermark recognition or extraction algorithm, Biometric payment system, Fingerprinting, fuzzy commitment scheme, helper-data system, fuzzy extractors, fuzzy vault, and cancelable biometrics, Fingerprinting Software, Reed–Solomon codes, HASP SRM

Introduction

Some common types of software piracy include

- Counterfeit
- OEM unbundling
- Soft lifting
- Hard disk loading
- Corporate software piracy
- Internet software piracy

Counterfeiting

Counterfeiting means producing fake copies of software, making it look authentic. This involves providing the box, CDs, and manuals, all designed to look as much like the original product as possible. Microsoft products are the ones most commonly counterfeited, because of their widespread use. Most commonly, a copy of a CD is made with a CD-burner, and a photocopy of the manual is made. Counterfeit software is sold on street corners, and sometimes unknowingly sold even in retail stores. Counterfeit software is sold at prices far below the actual retail price.

OEM unbundling

Often just called "unbundling," this form of piracy means selling stand-alone software originally meant to be included with a specific accompanying product. An example of this form of piracy is someone providing drivers to a specific printer without authorization.

Soft lifting

The most common type of piracy, soft lifting, (also called soft loading), means sharing a program with someone who is not authorized by the license agreement to use it. A common form of soft lifting involves purchasing a single licensed copy of software and then loading the software onto several computers, in violation of licensing terms. On college campuses, it is rare to find a software program that has not been soft loaded. People regularly lend programs to their roommates and friends, either not realizing it's wrong, or not thinking that it's a big deal. Soft lifting is common in both businesses and homes.

Hard disk loading

Often committed by hardware dealers, this form of piracy involves loading an unauthorized copy of software onto a computer being sold to the end user. This makes the deal more attractive to the buyer, at virtually no cost to the dealer. The dealer usually does not provide the buyer with manuals or the original CDs of the software. This is how operating systems, like Windows 95, are often pirated.

Corporate software piracy

A type of software piracy that occurs when corporations under report the number of softwareinstallations acquired through volume purchase agreements. Corporate software policy may also be an offence when software is installed on a server with unrestricted staff access. May also be called corporate end-user piracy.

Internet software piracy

A type of software piracy that occurs when software, which is illegally obtained through Internet channels, usually through peer-to-peer file sharing systems or downloaded from pirate Web sites that make software available for download for free or in exchange for users who uploaded software. Internet software piracy also includes the sale of counterfeit software on Internet auction and classified ads Web sites. Counterfeit software with de-activate the copy-protection that can be obtained through Internet channels is called **warez**.

Renting

Renting involves someone renting out a copy of software for temporary use, without the permission of the copyright holder. The practice, similar to that of renting a video from Blockbuster, violates the license agreement of software.

Online piracy

The fastest-growing form of piracy is Internet piracy. With the growing number of users online, and with the rapidly increasing connection speeds, the exchange of software on the Internet has attracted an extensive following. In the past, bulletin board systems (BBS) were the only place where one could download pirated software. Currently, there are hundreds of thousands of "warez" sites providing unlimited downloads to any user. Often, the software provided through these "warez" sites is cracked to eliminate any copy protection schemes.

Mechanisms employed to maintain the privacy of software

1. **Hardware locks**

Hardware locking locks the software to a specific computer. Hardware locking is used so that a single license cannot be used on multiple computers. This is generally unpopular with users because they will often have difficulties if they upgrade their system to a faster system or if they have a hard drive crash and need to replace failing hardware. In addition to the headaches associated with a system crash or upgrade, they will have the additional burden of contacting the software manufacturer so they can get their software to run on their new hardware.

2. Uses the passwords

There is wide use of passwords to protect software from illegal use, but this is not enough precaution, hackers crack the passwords as well.

3. Water markers

Software watermarking involves **embedding a unique identifier** within a piece of software, to discourage software theft. Watermarking does not prevent theft but instead discourages software thieves by providing a means to identify the owner of a piece of software and/or the origin of the stolen software. It can then be extracted by an **extractor** or verified by a **recognizer** to prove ownership of software. The former extracts the original watermark, while the latter merely confirms the presence of a watermark. A **watermark recognition or extraction algorithm** may also been classified as blind, where the original program and watermark is unavailable, or informed, where the original program and/or watermark is available.

It is also possible to **embed a unique customer identifier in each copy of the software** distributed which allows the software company to identify the individual that pirated the software. It is necessary that the watermark is hidden so that it cannot be detected and removed.

In most cases the **watermark should be robust** - that is, resilient to semantics preserving transformations (such as optimizations or obfuscations). However, in some cases it is desirable

that a watermark is **fragile** in the sense that if semantics preserving transformations are performed on the software the watermark becomes invalid. This is useful in the context of software licensing where any changes to a program could disable it.

Watermarking techniques are used extensively in the entertainment industry to identify multimedia files such as audio and video files, and the concept has extended into the software industry. Watermarking does not aim to make a program hard to steal or indecipherable like obfuscation but it discourages theft as thieves know that they could be identified.

4.Partial Key Verification

A partial key verification is a protection scheme that only verifies certain digits of the registration key. The verification digits vary in different versions.

5. Hardware Locking

Hardware locking locks the software to a specific computer. Hardware locking is used so that a single license can not be used on multiple computers. This is generally unpopular with users because they will often have difficulties if they upgrade their system to a faster system or if they have a hard drive crash and need to replace failing hardware. In addition to the headaches associated with a system crash or upgrade, they will have the additional burden of contacting the software manufacturer so they can get their software to run on their new hardware.

6.Online Activation

Software activation services provide software developers access to a centralized license server on the Internet, preventing software piracy by means of online software activation. Software activation is the process of obtaining a license for your software so that it becomes active and ready to use on your computer. Many customers dislike software activation because it is a form of "phoning home" and though the access is declared, it can still be an inconvenience if the customer is off-line.

7. Separate Trial and Download

Another approach developers take to protect their software is that their trial version is completely different from their registered version. After purchasing, the customer is given a new download

location to download the full registered version. The trial version is different than the registered version and it cannot be cracked to increase functionality.

The registered download can be time limited and password protected to help minimize its effect in the "wild", should a license be obtained through the use of a fraudulent credit card. File Kicker, a third party file hosting service, has a number of optional controls in place to restrict download access.

8. Dongle Locking

Dongle locking is another form of hardware locking. The software will require the use of an external piece of hardware (either connected to the parallel port or USB port) to "activate" the software. This is unpopular simply because it causes additional points of failure and relies on something that can be easily misplaced or lost.

It is generally difficult to locate lots of detail on software protection. Software developers tend not to post or share their methods of protecting their software in the public. As a result the software conferences and private member only forums tend to be the best places for getting detailed information on the most effective ways to protect software.

Proposed Multi ModelSolution against software piracy:

Biometric payment system

Putting together a form of identification that will allow software manufacturers to identify a product by purchaser and not by product. The system is based on **Biometrics** and the company in question is associating itself with major companies, not only for the prevention of software piracy, but also for online music piracy, movie piracy, and adult content rejection policies and even to online transactions.

In the case of software purchases, the customer merely makes the purchase via a **Biometric payment system** that records the transaction and places the software usability at the setup instigation of the verified purchaser. The installation process requires the user to complete installation via an online verification system that already knows the user has legally purchased the software and checks it against the user installing the software. If the two match, the software continues the install. If it fails, the installation.stops.

Biometric enabled software could be up to 75% cheaper in cost as the cost of fraud is minimized, which benefits the customer in cost savings. The upside of purchasing **Biometric enabled software** is that the user then has access to a myriad of services based around Biometrics. The Biometric system can be incorporated into online game

play to allow a more secure/verified way of playing, whether its joining game server (with registered game software) or making virtual transactions in game play.)

Fingerprinting (Biometrics Trait)

A **fingerprint** in its narrow sense is an impression left by the friction ridges of a human finger. In a wider use of the term, fingerprints are the traces of an impression from the friction ridges of any part of a human hand. A print from the foot can also leave an impression of friction ridges. A friction ridge is a raised portion of the epidermis on the fingers and toes (digits), the palm of the hand or the sole of the foot, consisting of one or more connected ridge units of friction ridge skin. These are sometimes known as "epidermal ridges" which are caused by the underlying interface between the dermal papillae of the dermis and the interpapillary (rete) pegs of the epidermis. These epidermal ridges serve to amplify vibrations triggered, for example, when fingertips brush across an uneven surface, better transmitting the signals to sensory nerves involved in fine texture perception. These ridges also assist in gripping rough surfaces, as well as smooth wet surfaces.

Impressions of fingerprints may be left behind on a surface by the natural secretions of sweat from the eccrine glands that are present in friction ridge skin, or they may be made by ink or other substances transferred from the peaks of friction ridges on the skin to a relatively smooth surface such as a fingerprint card. Fingerprint records normally contain impressions from the pad on the last joint of fingers and thumbs, although fingerprint cards also typically record portions of lower joint areas of the fingers.

A common scan hackers perform nowadays is fingerprinting a system in order to figure out what operating system it is running. The two main types of fingerprinting are Queso, which sends weird TCP flags, and nmap, which sends weird TCP options. Narrowing down the operating system is important. For example, attempting Windows-specific hacks against a UNIX system is pointless. Fingerprinting is possible because the TCP/IP specifications do not fully define the behavior of a protocol stack. Therefore, by sending unusual (undefined) network traffic at a

system, the hacker will receive responses unique to that system. Key point: One of the key reasons for fingerprinting a system is to search for "old" or "unusual" systems. Non-computer devices like routers, printers, modem banks, etc. are not written to the same level of security standards as real computers. In addition, a hacker may be able to find old SunOS 4 systems which are rife with well-known security flaws.

In recent years, the **protection of biometric data** has gained increased interest from the scientific community. Methods such as the **fuzzy commitment scheme, helper-data system, fuzzy extractors, fuzzy vault, and cancelable biometrics** have been proposed for **protecting biometric data**. Most of these methods use **cryptographic primitives or error-correcting codes (ECCs) and use a binary representation of the real-valued biometric data**. Hence, the difference between two biometric samples is given by the **Hamming distance (HD) or bit errors** between the binary vectors obtained from the enrollment and verification phases, respectively. **If the HD is smaller (larger) than the decision threshold, then the subject is accepted (rejected) as genuine**. Because of the use of ECCs, this decision threshold is limited to the maximum error-correcting capacity of the code, consequently limiting the false rejection rate (FRR) and false acceptance rate(FAR) tradeoff. A method to improve the FRR consists of using multiple biometric samples in either the enrollment or verification phase. The noise is suppressed, hence reducing the number of bit errors and decreasing the HD. In practice, the number of samples is empirically chosen without fully considering its fundamental impact. In this paper, we present a **Gaussian analytical framework** for **estimating the performance of a binary biometric system** given the number of samples being used in the enrollment and the verification phase.

The error-detection tradeoff curve that combines the false acceptance and false rejection rates is estimated to assess the system performance. The analytic expressions are validated using the Face Recognition Grand Challenge v2 and Fingerprint Verification Competition 2000 biometric databases.

Extracting binary strings from real-valued templates has been a fundamental issue in many **biometric template protection systems**.

One of the most important codes **Reed–Solomon codes** have since found important applications from deep-space communication to consumer electronics.

They are prominently used in consumer electronics such as CDs, DVDs, Blu-ray Discs, in data transmission technologies such as DSL&WiMAX, in broadcast systems such as DVB and ATSC, and in computer applications such as RAID 6 systems), random codes etc.

Fingerprinting Software

Software manufacturers are now using software fingerprinting to stop software piracy. Prolok Magic, a software fingerprinting product, as a piece of software that "creates a unique 'fingerprint' on each media, and encrypts the software program so that it is tied to the presence of this fingerprint." To run the Prolok'ed program, the "related Prolok-formatted [media] is mounted on the default drive" thus requiring the user to have possession of the specially formatted media. Users are able to create backup copies of their Prolok'edsoftware, however these copies are also Prolok-formatted. This appears to be a good solution.

Iris Biometrics

Aladdin **HASP SRM** for IP protection, secure licensing, and product activation, allowing it to expand into high-piracy regions.

HASP SRM, is outstanding in software protection and licensing solution, software against piracy

Taking advantage of the unique **Cross-Locking** capabilities of HASP SRM to protect and enforce licensing.

Since we have seen throughout this paper that every where we are trying to implement identifiers by different methods, methodologies &techniques, but practically we found every time a single Model is not efficient enough to protect software, because hackers found parallel their solution, hence **generating identifier(s)** with the use of **Biometric Traits**(Fingerprints, Voice, Face Detection, Iris Detection) would be a strong solution for software piracy, in Multi Model we should incorporate all traditional competent methods also and encapsulate in a Multi Model System, It would be next to impossible to crack,copy,illegal use of software by any means .