Copyrights & DRMs

LIOR NEUMANN (ליאור נוימן)

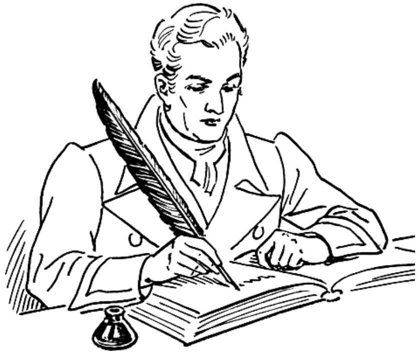
11.6.2014

Software Copyrights

- Software for early computers was given away for free by the hardware vendors or by the users who wrote it.
- Copyrights between companies was not an issue since programs were too specialized, too poorly documented, or just too hard to adapt.
- When minicomputers arrived in the 1960’s, software cost has started to become significant.

Software Birthmarks

- Software “Birthmarks” are unique identifying features in the way the program was implemented.
  - For example, the order in which registers are pushed and popped.
- It was used to detect code theft between companies, mostly caused by workers who moved from one company to another.
- Many code comparison tools were developed in universities to detect students cheating on programming assignments.



Time Bomb

- The first and simplest consumer copyrights mechanism.
  - After one year of using the program, the program will prompt an error message such as: “Error #XYZ - Please contact technical support”.
  - “XYZ” was an encrypted version of the license serial number which was then validated to ensure the copy is valid.
  - If the copy is valid the technical support would give the customer a series of instructions, in order to reactivate the product.
- This mechanism could be easily detected and removed using modern techniques.

The Growth in Need

- Software piracy really started to become an issue with the creation of a mass market in the late 1970’s and early 80’s, as software developers started to ship products that did not require support to install and run.
- There was a famous open letter from Bill Gates in 1976, in which he complained that less than 10% of all microcomputer users had paid for BASIC.
  - ‘Who cares if the people who worked on it get paid?’ he asked. ‘Is this fair?’

Modern Copyrights Protection Schemes

- There were three general approaches:
  - Add uniqueness on to the machine.
  - Create uniqueness in it.
  - Use whatever uniqueness happened to exist already by chance.

Hardware Dongle

- The standard way to add hardware uniqueness was a dongle: a device, typically attached to the PC’s parallel port (or USB port today), which could be interrogated by the software.
- The simplest just had a serial number, the most common executed a simple **challenge-response protocol**.
- Some top-end devices actually performed some critical part of the computation.

Naïve Copy Protection

- A common strategy was for the software to install itself on the PC's hard disk in a way that was resistant to naïve copying.
- For example, a sector of the hard disk would be marked as bad, and a critical part of the code or data written there.
- If the product were copied from the hard disk using the utilities provided by the operating system for this purpose, the data hidden in the bad sector would not be copied and so the copy would not work.
- Another similar strategy is the requirement of a master disk which was manipulated in such way that naive copy would not work.

My Computer

H:\ is not accessible.

Error performing inpage operation.

PC’s Configuration

- Another simple mechanism is to store the PC’s configuration at the time of installation
  - What cards were present, how much memory, what type of printer, etc.
- If this changed too radically, it would ask the user to phone the helpline.
- Similar mechanism is used in recent Windows operating systems.

Schemes Comparison

	Advantages	Disadvantages
Dongle	Provide the best protection among the three, if well implemented.	Expensive, therefore not feasible in many cases.
Naive Copy Protection	Very cheap and provides good protection against the average user.	Today it is almost impossible to implement due to high level of OS.
PC Configuration	Cheap and provides the required uniqueness to detect pirate copies.	Might cause false alarms, in case of legit PC major configuration change.

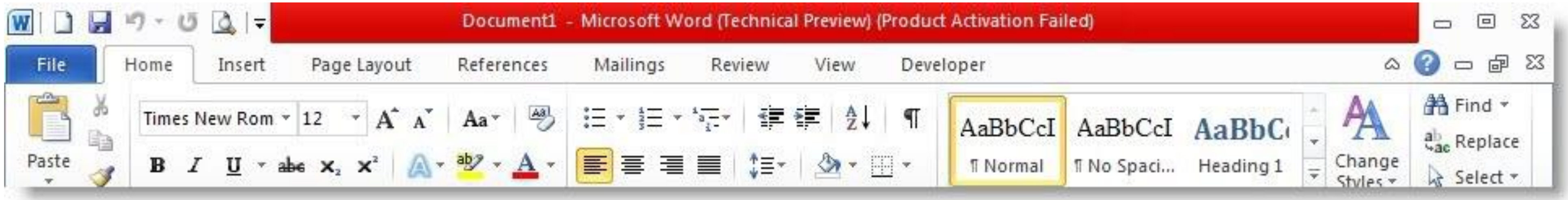
Where it Fails?

- A generic attack that works against most of these defenses is to go through the software with a debugger and remove all of the calls made to the copy protection routines.
- Many hobbyists did this for sport, and competed to put unprotected versions of software products online as soon as possible after their launch.
- This can be prevented by making critical code uncopiable, such as placing it in a dongle.

OllyDbg

Psychological Approach

- The vendors also used psychological techniques such as:
  - The installation routine for many business programs would embed the registered user’s name and company on the screen, for example, in the toolbar.
  - If early Microsoft software (Multiplan, Word or Chart) suspected you were running it under a debugger, trying to trace through it, it would put up the message *‘The tree of evil bears bitter fruit. Now trashing program disk’.*
  - If a modern version of Windows believes you are running unauthorized copy, it will show you a black desktop background. Microsoft Office might display a red status.



Document1 - Microsoft Word (Technical Preview) (Product Activation Failed)

SaaS Approach

- Software as a Service (SaaS) is a software licensing and delivery model in which software is centrally hosted on the cloud by independent software vendors (ISVs) or application service providers (ASPs).
- This model allow the provider ultimate control over the users credentials and overall use.
- Copy is not an issue anymore, since none of the critical code is visible to the user at anytime.
- The obvious downside of such a service is the requirement for continues online connection.
- Moreover, the maintenance of such a service is very expensive for the provider.

Summary – Software Copyrights

- With the exception of SaaS, **none of the mass-market protection technologies available today is foolproof**.
- But by using the right combination of them a large software vendor can usually get a tolerable result— especially if prices are not too extortionate and the vendor is not too unpopular.
- Small software companies are under less pressure, as their products tend to be more specialised and the risk of copying is lower, so they can often get away with making little or no effort to control copying.

Content Copyrights

- The major difficulty facing the content industry in contrast to the software industry lies on the difference between software and content.
- Software is basically a series of instructions, therefore the copyright protection is integrated as part of the product.
- Content on the other hand, does not contain information about how it should be played, therefore the protection responsibility is moved from the content maker to the hardware and platform vendors.

General Platform DRMs

- **Digital Rights Management** (**DRM**) is a class of technologies that are used by hardware manufacturers, publishers, copyright holders, and individuals with the intent to control the use of digital content and devices after sale.
- The definition of DRM is very wide, it had many different approaches and implementations through history.
- Examples of General Platform DRM:
  - iTunes
  - Windows Media Player (WMRM)