國 立 交 通 大 學

資訊科學與工程研究所

碩士論文

利用Windows Hook技術與Windows Service設計與實作之防側錄系統

Design and Implementation of Screen Capturing/Recording Prevention Software with Windows Hook Technology and Windows Service Technology

研 究 生：何彥霖

指導教授：陳登吉 教授

曾建超 教授

中 華 民 國 一 百 年 七 月

利用 Windows Hook 技術與 Windows Service 設計與實作防側錄系統

Design and Implementation of Screen Capturing/Recording Prevention Software with Windows Hook Technology and Windows Service Technology

研 究 生：何彥霖　　Student：Yen-Lin Ho

指導教授：曾建超　　Advisor：Dr. Chien-Chao Tseng

陳登吉　　Dr. Deng-Jyi Chen

國 立 交 通 大 學

資 訊 科 學 與 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

July 2011

Hsinchu, Taiwan, Republic of China

中華民國一百年七月

利用 Windows Hook 技術與 Windows Service 設計與實作之防側錄系統

學生: 何彥霖　　指導教授: 曾建超 博士與陳登吉 博士

國立交通大學 資訊科學與工程研究所

摘要

在這資訊膨脹的時代，各行各業不斷地加速數位化的腳步，使數位內容日漸普及，人們在彈指之間就可以獲得許多訊息。隨著網際網路傳輸速度的增加，大量的資料可在世界各地迅速地傳遞，若是未經過保護的數位內容，如：圖片、聲音、視訊檔案等等，就很容易被下載與複製而導致數位內容盜版猖盛。根據 Net Application 數據顯示 Windows 作業系統的市占率高達九成以上，所以數位內容的使用主要在 Windows 作業系統上，因此亟需在 Windows 上保護數位內容的智慧財產權，以預防惡意的侵權行為以及保護數位內容產業的發展。

原本智慧財產權的演進已經跟不上數位時代的變遷，為了保護數位內容的智慧財產權，進而開始衍生了許多的數位內容保護技術，如：數位版權管理系統(*Digital Rights Management System*)、選擇性編碼(Selective Encryption)技術與可變轉碼(Scalable Encoding)、數位浮水印（watermark）、數位指紋(Fingerprinting)、防側錄技術等研究，目的都是在保護數位內容產業的發展以及兼顧創作者權利。

其中防側錄技術也是目前數位內容保護的一項重要技術，目的是為了防止使用者開啟數位內容之後，使用桌面或視窗錄影程式錄製數位內容。目前的防側錄技術作法有兩種：利用 API Hook 技術與特徵碼比對技術。API Hook 技術主要用來監視系統中有關擷取螢幕畫面之 API 內容，進一步攔截或阻擋此行為之發生。特徵碼比對技術主要利用資料庫與系統中程式之特徵碼作比對，以達到辨識是否是側錄程式。

但是使用 API Hook 技術與特徵碼比對技術無法滿足所有防側錄之需求。例如：API Hook 技術有可能被防毒軟體視為攻擊行為而無法發揮防側錄之功能、借由修改程式執行檔而產生不同特徵碼可躲過特徵碼比對技術之側錄偵測。所以本研究將在 Windows 上研發一套完善的防側錄系統，以偵測出所有側錄行為。

Design and Implementation of Screen Capturing/Recording Prevention Software with Windows Hook Technology and Windows Service Technology

Student: Yen-Lin Ho

Advisors: Dr. Deng-Jyi Chen
Dr. Chien-Chao Tseng

Department of Computer Science and Information Engineering

National Chiao Tung University

ABSTRACT

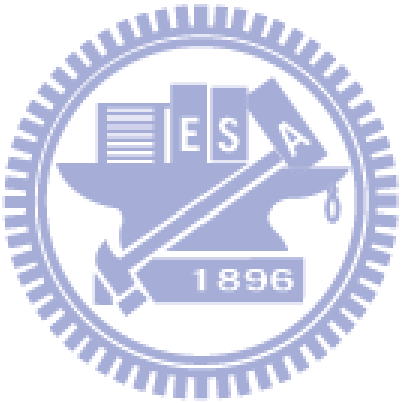
In this age of information expansion, every aspect of technology continues to accelerate its digital footsteps and become easily accessible. People can easily retrieve information at their fingertips. As Internet speeds increase, a huge amount of information can be passed quickly around the world. Without the protection of information, pictures, audio, video files and much more, can be easily downloaded and copied. The Net Application data that is based on the Windows operating system has over 90 percent of the market share; therefore, the digital content mainly used on the Windows operating system, is in urgent need of protection for its intellectual property rights of digital content. The purpose of this protection is to prevent malicious violations in the digital content industry.

The evolution of intellectual property rights has been trailing behind the rapid development of the digital age. The purpose is to protect the digital content industry and the rights of the creators. This has lead to a gradual rise in digital content protection technology, such as: *Digital Rights Management System*, Selective Encryption, Scalable Encoding, Watermark, Fingerprinting and Screen Capturing/Recording Prevention (SCRP) Technology.

SCRP is a vital part of protection of digital content. Its objective is to prevent

desktop or window recording program users to from recording digital content. The current SCRP Technology has approaches: using API Hook technology or characteristic code matching technology. API Hook technology is operated to monitor the APIs of screenshots on the system and to further block or stop the occurrence of this behavior. Characteristic code matching technology compares the program's characteristic code and its database to identify whether or not it is a screenshot program.

However, using API Hook technology and characteristic code matching technology may not sufficient for the needs of SCRP, such as: API Hook technology which may be regarded as aggressive behavior by anti-virus software and disabling it from implementing the function of SCRP technology. Through the modification of the executable program file, a new characteristic code can be generated that allows it to escape the characteristic code matching technology. Therefore, this study will develop comprehensive SCRP software on Windows to detect all behaviors of screen capturing/recording.



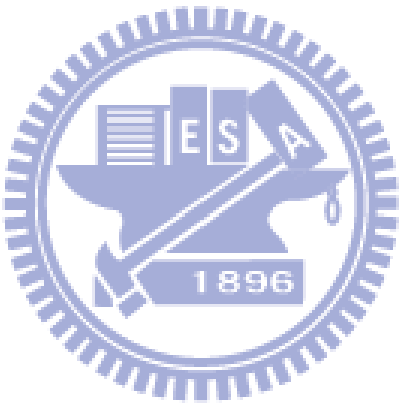
致謝

首先感謝我的指導教授陳登吉老師，老師無論在課業、或研究上，每當我遇到問題與挫折時，都會耐心的指導及教誨，並給予我實質的建議與方向。即便在研究的過程中受到飽受打擊，甚至得重新開始，老師依舊對我有信心，也給了我更多的鼓勵，讓我能順利的走下去，我由衷地感謝老師，也祝福老師的身體安泰。

再來要感謝的是我的同窗好友們，能夠在課業上相互扶持，有問題大家一起討論解決；當研究上遇到瓶頸時，這些好同學能給我很多建議，讓我能夠完成這項研究。

最後我要感謝在背後默默支持我的家人，不斷地給予我正面的鼓勵，也相信我一定可以做得很好；朋友們也時常噓寒問暖，了解我的近況，分享我的心事，做個專職的傾聽者。有了支持我的家人與朋友，讓我更堅決自己所要走的路，並能義無反顧的做下去。

圖目錄



目錄

一、 緒論

1.1 研究動機

隨著科技的進步以及網際網路的發展，人們將傳統紙本的課程內容漸漸以數位內容的形式呈現。例如:多媒體互動學習課程、電子文件與網頁等等，可讓閱讀者透過網際網路直接線上閱讀觀看或下載到自己電腦以供以後閱讀。如此的學習方式打破了地點與時間的限制。學習的地點可以在辦公室會議室甚至自己的房間內，不再局限於教室、圖書館等等；學習的時間可以配合自己的作息而安排，不再限制於上課時間，讓學習更加的多元化。

因為學習方式的變化，對於未經過保護的數位內容，如：圖片、聲音、視訊檔案等等，就很容易被下載與複製。加上網際網路傳輸速度的增加，大量的資料可在世界各地迅速地傳遞，導致數位內容盜版猖盛。因為數位內容取得之便利而打擊了數位內容的智慧財產權(IPR)，更加抑制創作者的創作意願，也影響到原創的發展。因為 Windows 作業系統的市占率高達九成以上，所以數位內容的使用主要在 Windows 作業系統上，因此亟需在 Windows 上保護數位內容的智慧財產權，以預防惡意的侵權行為以及保護數位內容產業的發展。

為了保護數位內容的智慧財產權並預防惡意的侵權行為，衍生了許多的數位內容保護技術。如：數位版權管理系統(Digital Rights Management System)、選擇性編碼(Selective Encryption)技術與可變轉碼(Scalable Encoding)、數位浮水印（watermark）、數位指紋(Fingerprinting)、防側錄技術等研究，目的都是在保護數位內容產業能的發展以及兼顧創作者權利。

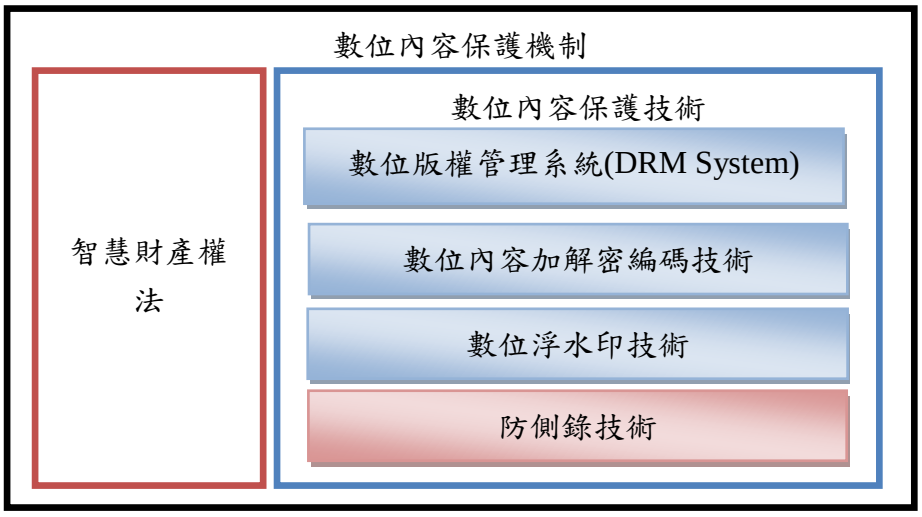


圖 1 數位內容保護機制架構圖

在數位內容保護技術中分成兩種類型，分別為數位內容使用前的保護與數位內容使用中的保護。前者是針對數位內容檔案本身作進一步的處理已達到保護之目的，例如：使用加解密演算法將數位內容檔案做編碼，並由數位版權管理系統管理加解密金鑰或是在數位內容中加入可視或不可視的浮水印驗證等等，後者是針對已經透過正當認證程序而開始播放的數位內容作保護，例如防側錄技術。此技術之主要目的是為了防止使用者開啟數位內容之後，使用桌面或視窗錄影程式錄製數位內容。

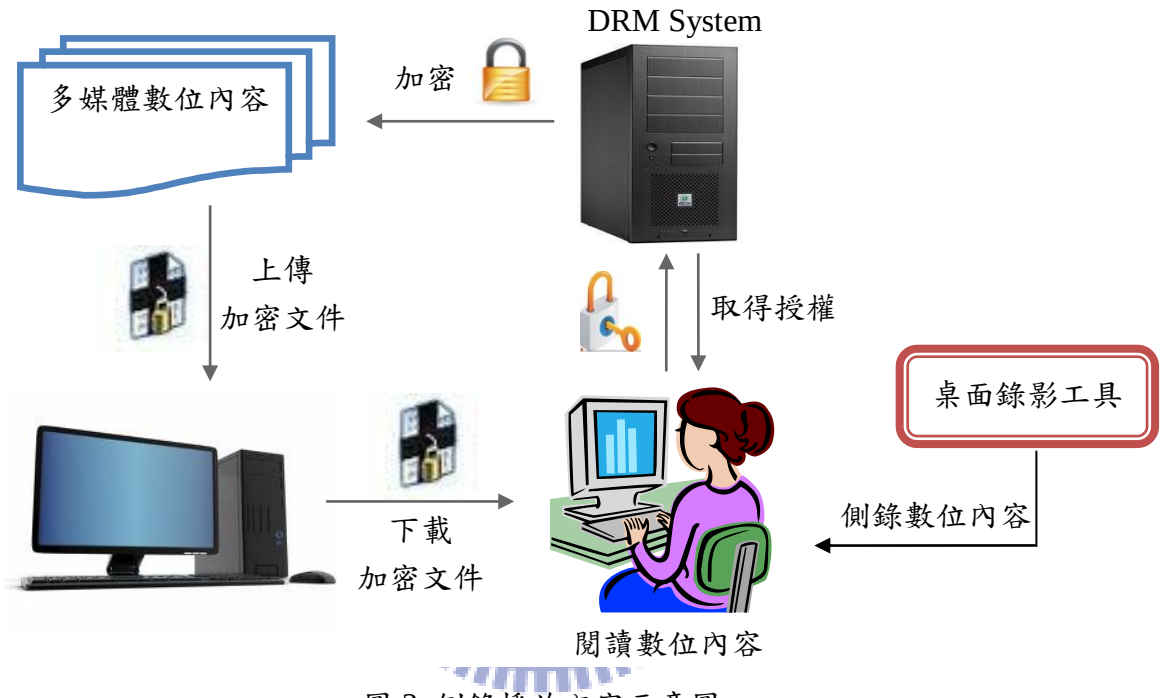


圖 2 側錄播放內容示意圖

目前的防側錄技術作法有兩種：利用 API Hook [6]技術與特徵碼比對技術。API Hook 技術主要用來監視系統中處理程序之 API 內容，是否有擷取桌面畫面之行為，若有擷取畫面之行為，則可進一步攔截或阻擋此行為之發生。特徵碼比對技術主要利用資料庫與系統中處理程序之特徵碼作比對，以達到辨識是否是側錄程式之效果，若被辨識為側錄程式，則可馬上停止展示數位內容。

但是使用 API Hook 技術與特徵碼比對技術無法滿足所有防側錄之需求。例如：API Hook 技術有可能被防毒軟體視為攻擊行為而無法發揮側錄偵測之功能、借由修改程式執行檔而產生不同特徵碼可躲過特徵碼比對技術之側錄偵測。所以需要在 Windows 上研發一套完善的防側錄系統，以偵測出所有側錄行為。

1.2 研究目的

本研究主要目標:「在 Windows 上設計與實做一套可偵測其他程式行為的防側錄系統」。

最新的 Windows 作業系統－Windows 7 應用了一系列的新技術,其中為了提高使用者的安全水準而生的新技術－使用者帳戶控制(User Account Control; UAC)[19],但此技術卻增加防側錄系統之側錄偵測的困難度。

大多使用者都只使用管理者帳戶登入 Windows,當使用者受到網路攻擊時,網路攻擊者就可毫無限制地存取使用者的相關訊息或做其他惡意的破壞行為,UAC 技術就是為了防止以上的情況而產生的。

UAC 技術簡單地說就是預設所有使用者程式只具備標準使用者(Standard User)權限,使受限制的使用者程式無法存取需要更高權限才能取得的資源。但是可以額外賦予使用者程式具備管理者(Administrator)權限,在使用者程式啟動之前可詢問使用者是否同意對於提升權限,前提是該使用者必須是能夠取得管理者權限。

因為受限制的使用者程式無法存取需要更高權限才能取得的資源,使低權限的使用者程式無法取得高權限的使用者程式之相關訊息,所以使用者可以借由指定側錄程式具備管理者權限而避開只具備標準使用者權限的防側錄系統之側錄偵測。因此,使防側錄系統必須具備管理者權限,才能偵測到其他程式之行為。

1.3 研究方法與步驟

因為現有的防側錄系統不計其數,因此本研究將使用智勝國際科技股份有限公司所研發的 BestWise 防側錄系統作為主要的實作以及實驗對象,並說明防側錄系統中的防側錄模組是如何運作。研究方法與步驟分為以下幾點:

(a).了解與評估現有之系統

針對現有的防側錄系統中的防側錄模組了解其防側錄技術的作法,並找出該防側錄技術作法之缺點。再針對其缺點而衍生出本研究之設計方式與運作機制。

(b).防側錄系統分析

了解 BestWise 防側錄系統之架構與其防側錄模組之流程，再加以分析本研究設計的防側錄模組所需要的功能，以及模組之間的溝通方式，設計一套新的防側錄模組運作流程。

(c).防側錄模組實作

依照之前分析的防側錄模組需求以及實驗對象，建置開發環境與工具，並著手實作出防側錄模組。為了達到偵測其他程式行為之目的，將防側錄模組以管理者權限運作並加強防側錄系統之辨識力

(d).應用實例與結論

最後會逐步說明本研究所設計的防側錄機制如何在 Windows 7 上偵測其他程式之行為，並使用側錄程式測試與驗證其偵測能力。

在最後一個章節中，會總結本研究在數位內容課程保護規劃上的彈性，對數位內容原始作者提供更高的安全性，並說明與其他防側錄模組之不同，以及改善與補強舊有防側錄模組之不足。

1.4 相關名詞解釋

本研究根據 1.2 節所提出的研究目標，提供相關名詞解釋，茲描述如下：

- API Hook：

  在 Windows 系統下編成，都會接觸到 API 函式之使用，常用的 API 函式大概有 2000 個左右。透過 Hook 技術可動態連接到需要修改的 API 函式入口點，並修改它的位址而指向新的自定義之函式。

- 特徵碼：

  利用由檔案中取出部分可以代表此檔案特徵值的內容，透過演算法計算而產生一固定長度的字串碼，用來代表此檔案，稱之為「特徵碼」。可用於病毒碼掃毒、軟體特徵碼之製作。

1.5 章節說明

本論文共分為六個章節，以下簡單說明各章節內容：

第一章，首先介紹數位內容的普及與數位內容保護技術，並說明目前防側錄技術的發展與重要性，進一步由目前技術的問題中導引出本研究的動機與目標，簡略介紹本研究的方法與步驟。

第二章，介紹目前的防側錄模組技術，如：BestWise、TrustView、OsafeMirage、W&Jsoft 等公司所研發的防側錄模組，並以 BestWise 與 TrustView 說明目前主要的作法，進一步了解這些技術在實際應用上所遇到的問題。

第三章，了解實驗對象的系統架構與防側錄模組運作機制，並根據第二章所提出的問題，衍生本研究所提出的防側錄模組的運作機制，並規劃出所需要的功能模組，最後開始著手分析各模組之功能，以及各模組之間的溝通方式。並介紹本研究的防側錄系統架構，以及防側錄模組的運作機制。

第四章，詳細介紹各個模組的設計概念與實作方式，並說明如何與現有的 DRM Server 做溝通，以達到對文件的控管；另外，詳述如何與智勝國際科技股份有限公司的講解手撥放軟題作配合，完成對數位內容的防側錄保護。

第五章，將實作出來的防側錄系統，透過畫面擷取與文字解說，逐步說明實際在閱讀端模擬不同的側錄作法，驗證防側錄系統是否能夠偵測出側錄程式之行為，以有效防止數位內容被錄製。

第六章，本章為總結，說明本研究之研究結果，所達到的研究目標，以及未來展望。

二、 相關研究

在 Windows 作業系統中，大多數用來錄製桌面或視窗畫面的應用軟體，都是透過使用 Windows API 技術完成，進而再儲存成視訊或圖片等檔案格式，因此防側錄軟體為了能及時偵測出 Windows API 之行為，所以大多都使用 API Hook 技術來監控執行程式裡所有運作中的 Windows API，若是發現使用與擷取畫面相關的 Windows API 時，則會立即阻擋該行為之結果，例如：對被截取之畫面的影像內容做修改，並以自定的影像取代被截取畫面之影像內容。目前有許多的數位內容保護控管公司，例如：Osafe Mirage[9]、TrustView[10]、W&Jsoft[11]等等，都是採用此項技術來開發防側錄功能，以保護這些經過權限管控之文件或課程內容。

但是只使用 API Hook 技術無法完全滿足防側錄之需求，例如：無法有效阻擋遠端遙控的側錄軟體之側錄、或是可能被部分的防毒軟體式為可疑行為。因此，有其他公司使用特徵碼比對技術來開發防側錄軟體，例如：BestWise。借由資料庫的特徵碼與應用軟體執行檔之特徵碼做比對而達到辨識應用軟體之身份，若辨識結果是側錄軟體，則立刻停止播放數位內容。

在接下來的小節中，將使用 TrustView 與 BestWise 所提供的受數位版權保護(DRM)文件，針對其防側錄的運作方式做說明，並歸納出該使用上之問題。最後，再對本研究所使用的相關技術做詳細的介紹與探討。

2.1 現有防側錄模組的運作方式與問題

2.1.1 TrustView 之運作方式

在此章節將展示由 TrustView 所提供的受數位內容保護文件，此文件為一份經過編碼的 Word 檔案，因此無法直接使用 Word 開啟，需要再安裝一個外掛模組才可解密文件。接下來將針對防側錄模組之運作方式逐步做說明，在說明中將會安裝 TrustView 的外掛模組 TrustView For Office and PDF 3.5.0，並使用 Microsoft Office Word 2003 開啟受保護的文件，最後再歸納使用時所發現的問題。

Step1、首先開啟 TrustView 所提供的受數位版權保護之 Word 文件，並使用未安裝 TrustView 外掛模組的 Microsoft Office Word 開啟該文件，由圖 3 可以看到該文件是無法正常觀看之提示訊息，訊息其中指示須先 TrustView 的外掛模組才可正常觀看文件。

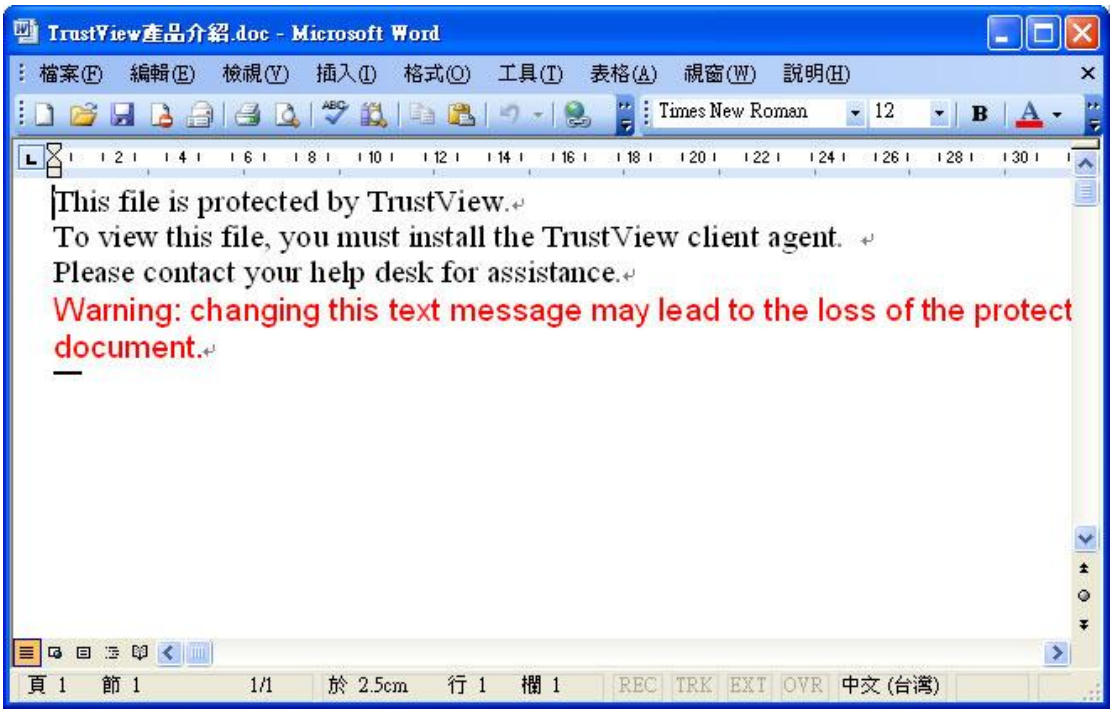


圖 3 開啟 TrustView 的受保護 Word 文件

Step2、開始安裝 TrustView 外掛模組，圖 4 為安裝畫面。

圖 4 TrustView 外掛模組安裝畫面

Step3、安裝完外掛模組之後，再次打開該份文件，此時可由 Microsoft Word 中的工具列中發現 TrustView 的外掛模組選項，若有則代表外掛模組安裝成功，如圖 5 所示。

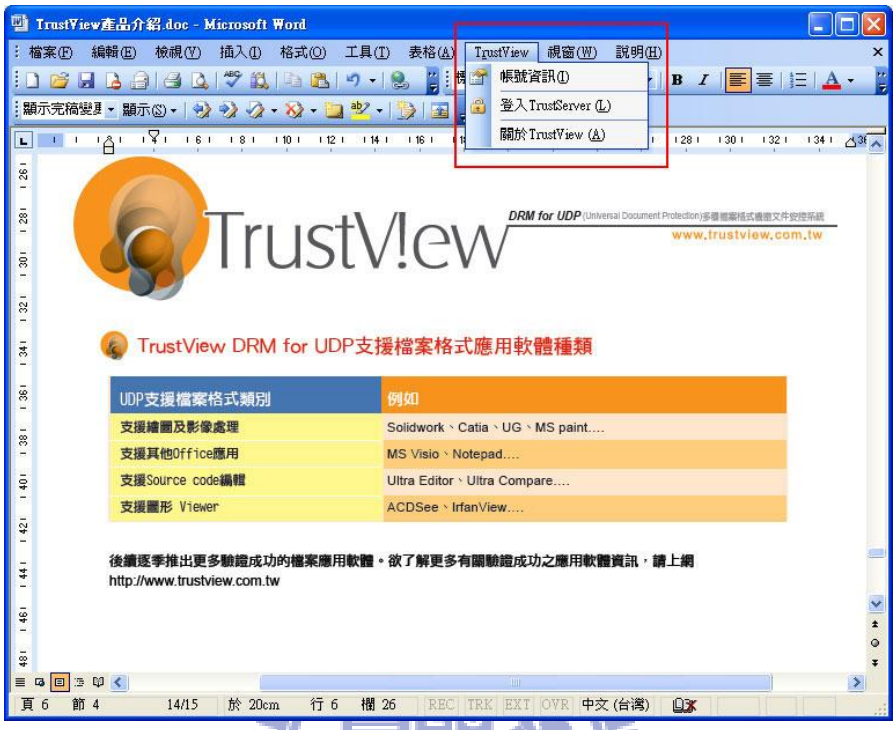


圖 5 開啟受保護之文件

Step4、開啟具有擷取桌面及視窗畫面之功能的側錄軟體，用來測試 TrustView 外掛模組之防側錄效果。此步驟中我選用 WMSnap 3.0.0.74 側錄軟體，此軟體界面如圖 6 所示。

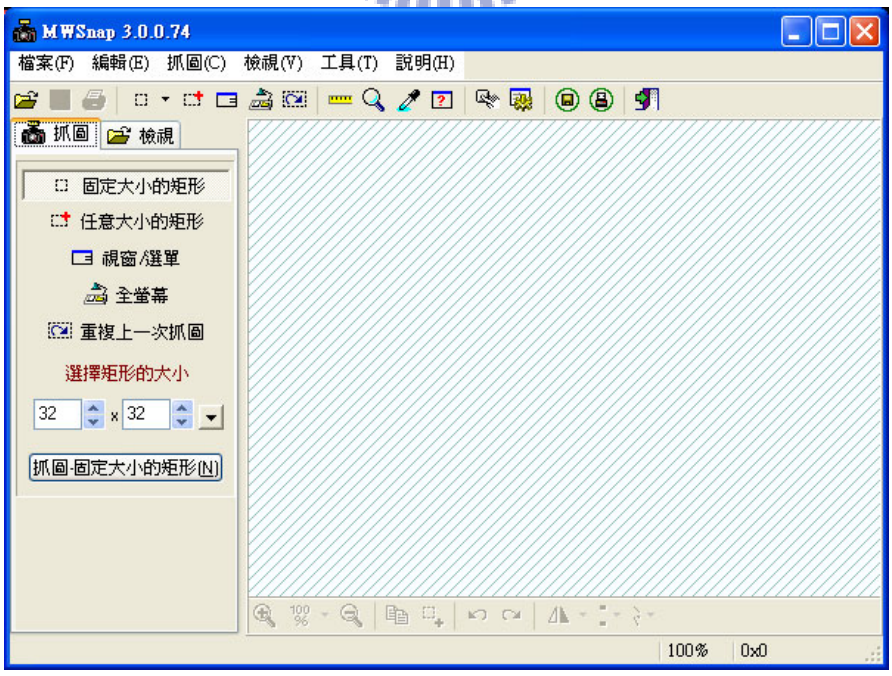


圖 6 開啟 MWSnap 3.0.0.74 側錄軟體

Step5、為了測試 TrsuView 外掛模組保護之防側錄效果，因此先開啟一份未受保護的 Word 文件，再使用 MWSnap 3.0.0.74 的畫面擷取功能。此時可以成功擷取該 Word 文件之內容，如圖 7 所示。

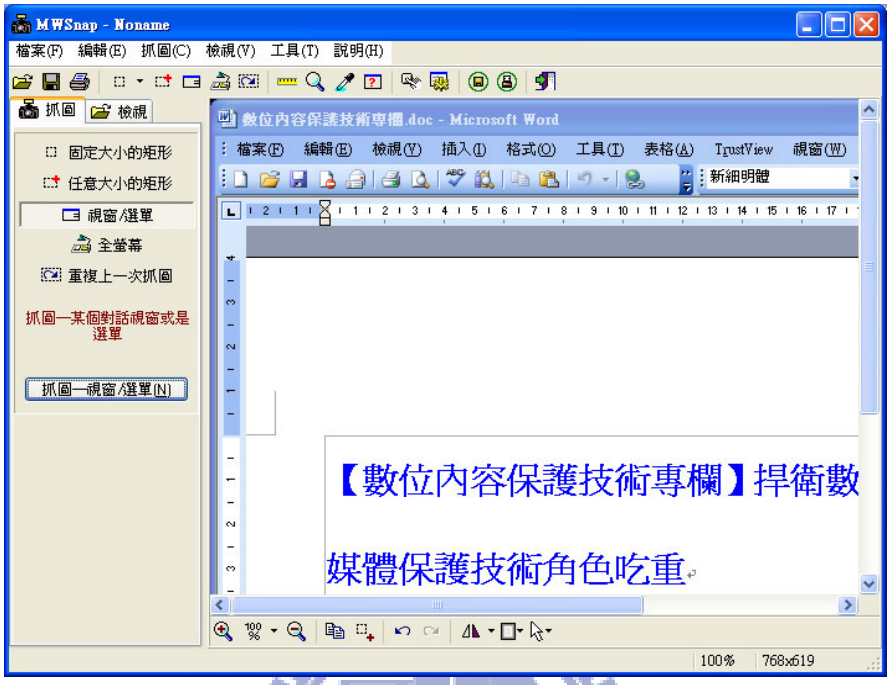


圖 7 未受 TrustView 保護之 Word 文件

Step6、再開始受 TrustView 保護之 Word 文件，並使用相同的側錄軟體與操作步驟，擷取該文件之內容。其結果顯示該側錄軟題執行擷取畫面之動作時，被 TrustView 外掛模組修改其擷取畫面之結果，以達到防側錄之效果。如圖 8 所示。

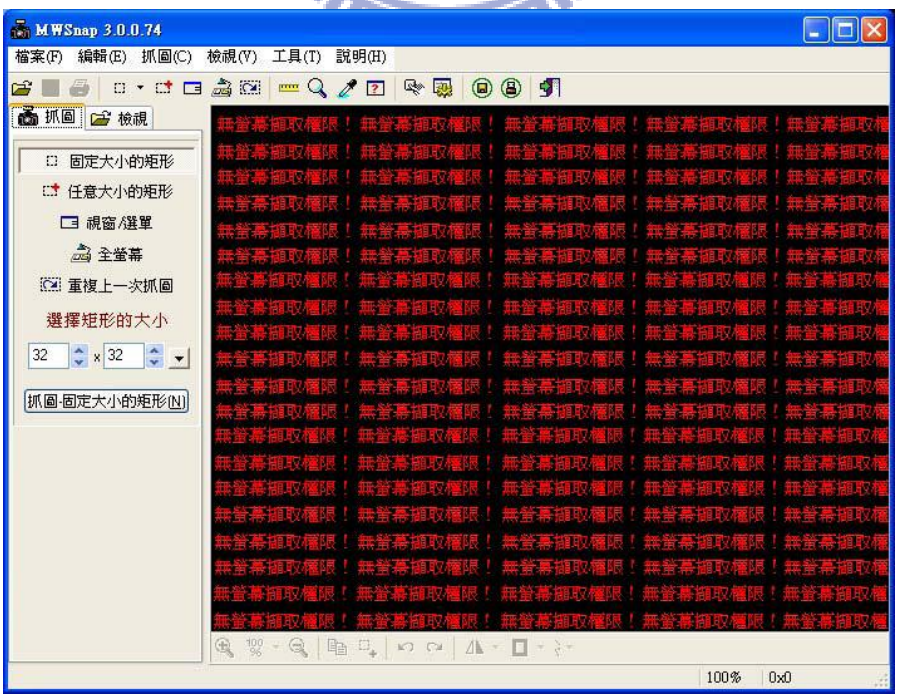


圖 8 受 TrustView 保護之 Word 文件

由 TrustView 所提供的防側錄模組只針對受 DRM 數位版權管控的文件，因此可以推論，該防側錄模組的系統架構是以【外掛】的模式依附在該文件的閱讀軟體上，例如:Microsoft Word。也就是說當使用 Word 開啟受保護之文件的同時，也會開啟該防側錄模組並且執行。

所謂的【外掛】技術，即是由非原軟體中所提供的功能，而是由第三方開發並提供特殊目的功能之模組，此模組無法獨立執行，必須由被依附之軟體啟動時，再將此特殊目的功能之模組載入該軟體中。

在 Windows 作業系統中，【外掛】技術的實作方法除了使用原軟體所提供的介面之外，其他最常用被使用的方法就是利用 DLL Injection 技術[13]。簡單的來說就是將自行開發的功能模組包裝成 DLL 檔案，在找出在作業系統中獨立運作的 Process，將自行開發的 DLL 檔案載入該 Process 中，使其成為該 Process 定址空間的一部份。因此可以 Process 中的指定執行函式之記憶體位置，修改成該 DLL 中的函式位址。

為了驗證以上的推論，因此使用 Process Monitor 這套軟體工具來做實驗。Process Monitor 是一套進階的系統監視工具，可顯示出即時系統中所有正在運作的 Process，並且可以顯示該 Process 中總共 Inject 了多少 DLL 模組。借由此工具可以清楚的看到當受保護之 Word 被開啟時，是否有被 Inject TrustView 的外掛模組。圖 9 為開啟 Process Monitor 軟體工具之後，列出目前所有正在執行的 Process。

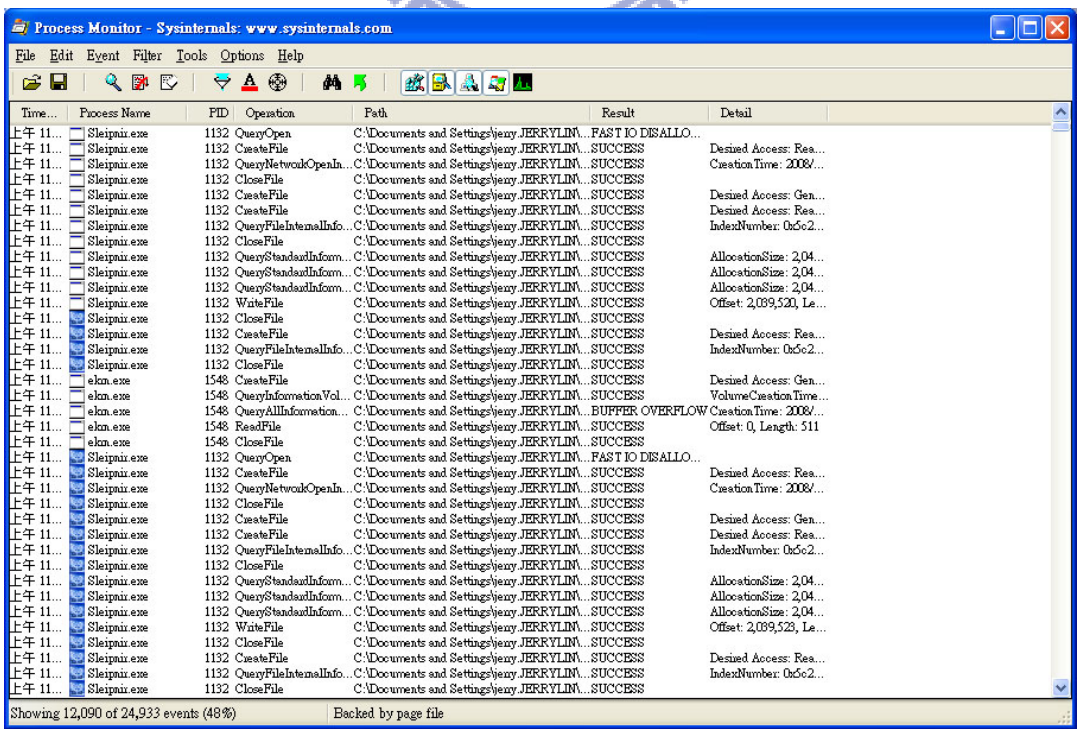


圖 9 開啟 Process Monitor 視窗

開啟受保護的 Word 文件，並查看該閱讀軟體中共 Inject 了多少 DLL 模組。發現其中有 HOOKTOOL.dll 正在運作，該 DLL 之實際位置即是安裝在 TrustView 的安裝資料夾中，如圖 10 所示。

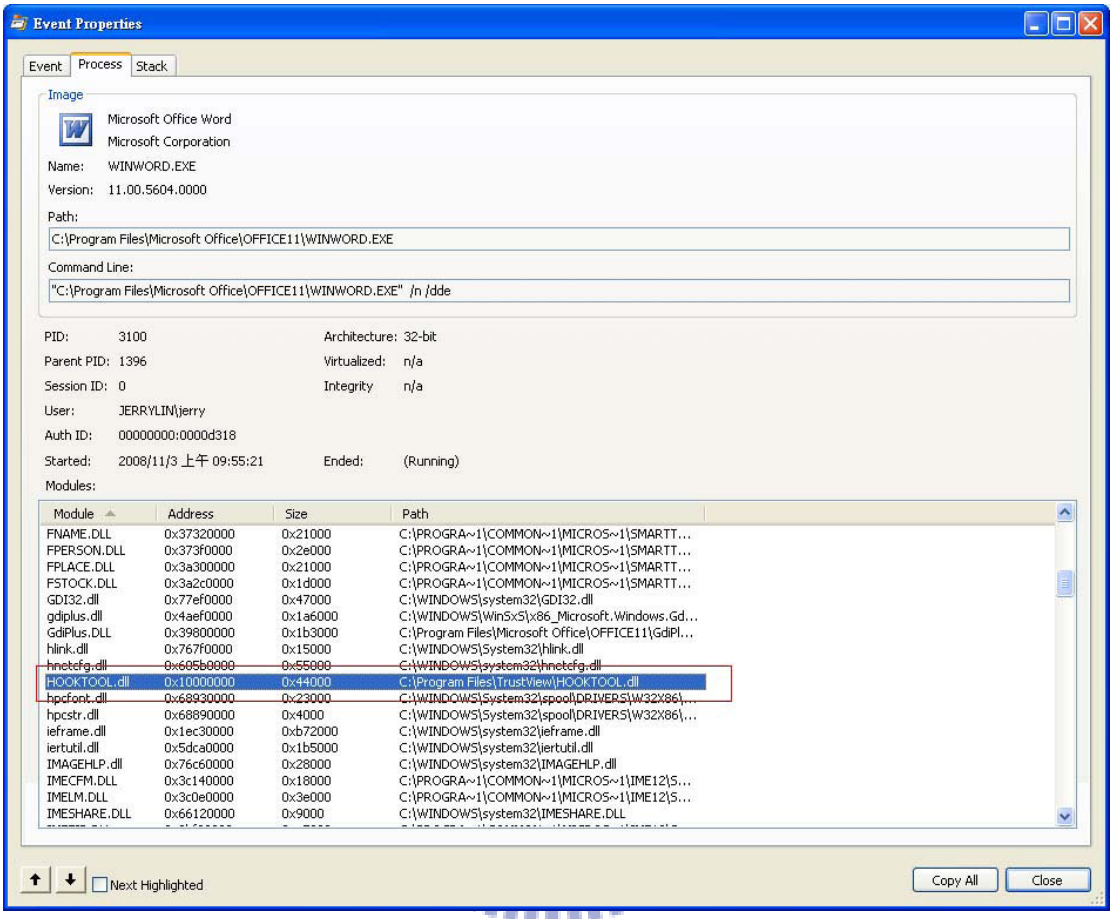


圖 10 在 Microsoft Word 中發現有 HOOKTOOL.dll

因此可以確定該防側錄模組式使用了 DLL Injection 技術，將防側錄功能外掛在 Process 中，除了可以監控側錄行為之外，還可以對被側錄之結果內容做修改。

2.1.2 TrustView 之問題

由 2.1.1 節中大致確認了 TrustView 的防側錄作法是使用 DLL Injection 技術，並利用 API Hook 技術來監控所指定的 API，進而改變 API 之執行結果為自行定義之內容。因此當該防側錄模組發現有對桌面或視窗進行側錄行為時，該防側錄模組會對被擷取的畫面內容做清除,並畫上自訂的內容，並提示該份文件已受防側錄模組之保護。為了確認以上技術之防側錄效果，利用其他不同種類的桌面及視窗錄影軟體對此受保護之文件進行測試，發現都可以有效的防止桌面即視窗被側錄。因此可以確定在 Windows 作業系統上使用 DLL Injection 技術與 API Hook

技術可有效防止大部份的側錄軟體，TrustView 防側錄模組作法如圖 11。

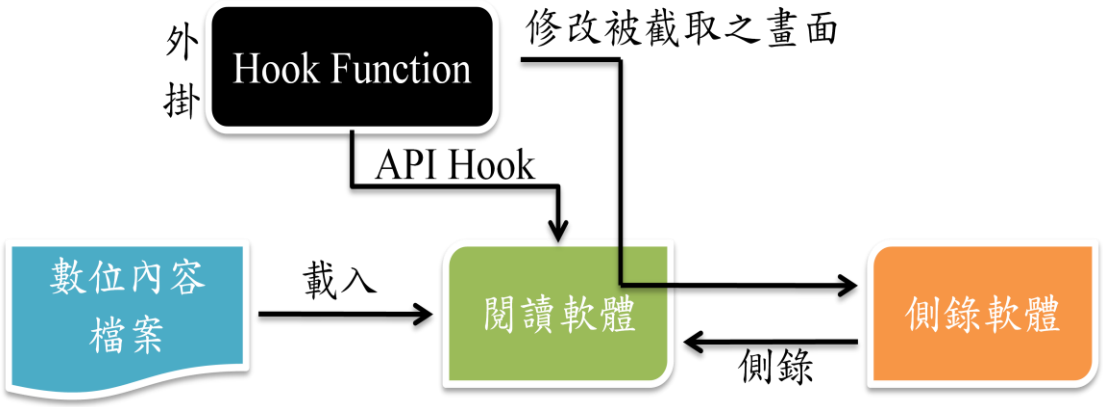


圖 11 TrustView 防側錄模組作法

但仍有其他的可能性或方法是使用 API Hook 技術之防側錄模組所無法阻擋的。因為 API Hook 僅能監控所在的作業系統上之所有 Process 的 API 行為，所以可以透過遠端連線之手法進而躲過該防側錄模組之偵測。為了驗證以上的說法，建構了以下之實驗。

利用 VNC (Virtual Network Computing) [14]遠端遙控軟體做實驗，VNC 的架構包括 VNC Server 以及 VNC Viewer。在 Server 端主要負責傳送電腦桌面畫面以及接受 Viewer 端的指令，而 Viewer 端則是接收 Server 端傳送過來的畫面並且傳送指令給 Server 端，環境設置如圖 12 所示。

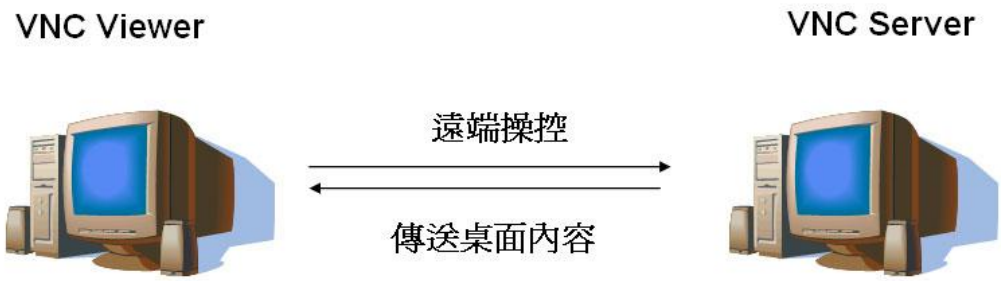


圖 12 VNC 軟體環境設置圖

實驗結果驗證以上的做法是可以躲過 API Hook 之監控，以側錄受保護之文件的內容，圖 13 為擷取文件內容之畫面。

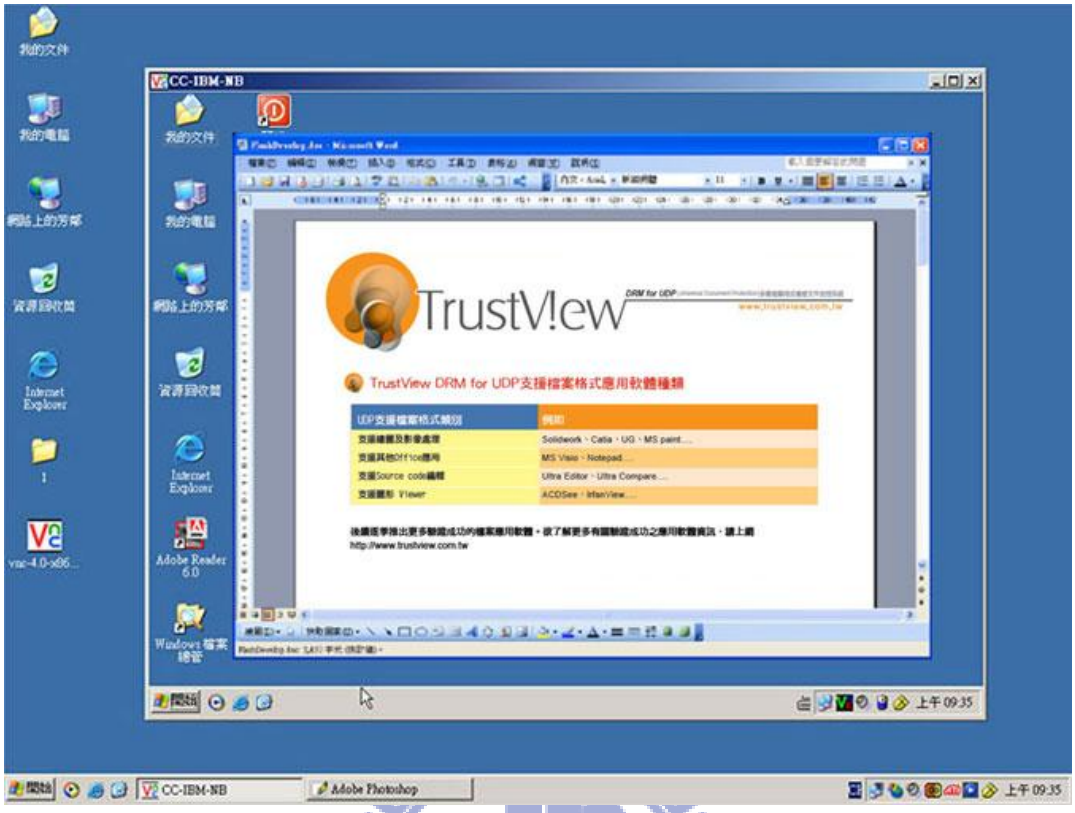


圖 13 透過遠端遙控軟體擷取受保護之文件

另外存在一個問題就是 DLL Injection 技術會被某些防毒軟體視為侵害電腦之攻擊行為，因為 DLL Injection 技術被廣泛運用在 Win32 病毒中，將病毒碼本身藏於 DLL 檔中，當 Process 啟動時再將此病毒 DLL 加載至 Process 中並開始運行。

加上，TrustView 防側錄模組中利用 DLL Injection 技術與 API Hook 技術，影響 Process 中某些 DLL 中的 API 之執行結果，以達到防側錄之目的。因此會導致該防側錄模組無法正常運作而失去保護文件之目的。

2.1.3 BestWise 之運作方式

在此一章節中，我將展示由 BestWise 所提供的受數位版權保護文件，此文件為一份經過編碼的多媒體檔案，必須使用 BestWise 所提供的閱讀軟體-講解手瀏覽器才可播放，所以該防側錄模組直接包在該閱讀軟體中，不需要額外安裝。接下來我將針對防側錄的流程逐步作說明。

Step1、首先開啟 BestWise 所提供的受數位版權保護文件，接著會開啟所對應的閱讀軟體-講解手瀏覽器，該閱讀軟體之介面如圖 14 所示。

圖 14 BestWise 所提供的閱讀軟體之介面

Step2、開啟具有擷取桌面及視窗畫面之功能的側錄軟體，用來測試 BestWise 閱讀軟體之防側錄效果。此步驟中我選用 WMSnap 3.0.0.74 側錄軟體，此軟體界面如圖 15 所示。

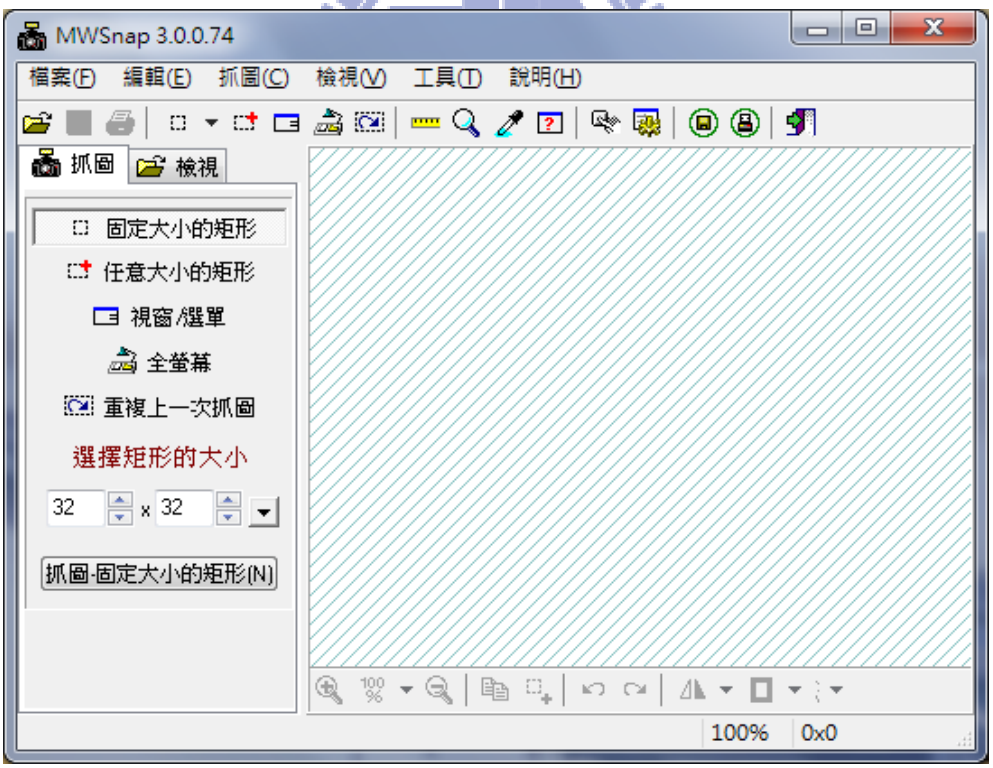


圖 15 開啟 MWSnap 3.0.0.74 側錄軟體

Step3、為了測試受保護文件之防側錄效果，因此我利用 MWSnap 3.0.0.74 的擷取視窗功能，此時播放軟體中的防側錄模組偵測出有側錄軟體正在運作，並同時停止播放該受保護文件之內容。

該防側錄模組主要是利用特徵碼技術與資料庫達到辨識側錄程式之目的。特徵碼技術是將處理程序（Process)之執行檔透過特徵碼演算法運算，產生一組可代表該處理程序的字串碼，稱之為特徵碼。將已知的側錄程式利用特徵碼技術產生所對應的特徵碼，並同時儲存至特徵碼資料庫中。該防側錄模組之側錄偵測是將所有正在執行中的處理程序透過特徵碼技術生成代表該處理程序之特徵碼，並且與特徵碼資料庫進行比對，便可辨識是否有側錄程式。若有，則馬上停止播放該受保護文件之內容，BestWise 防側錄模組作法如圖 16。

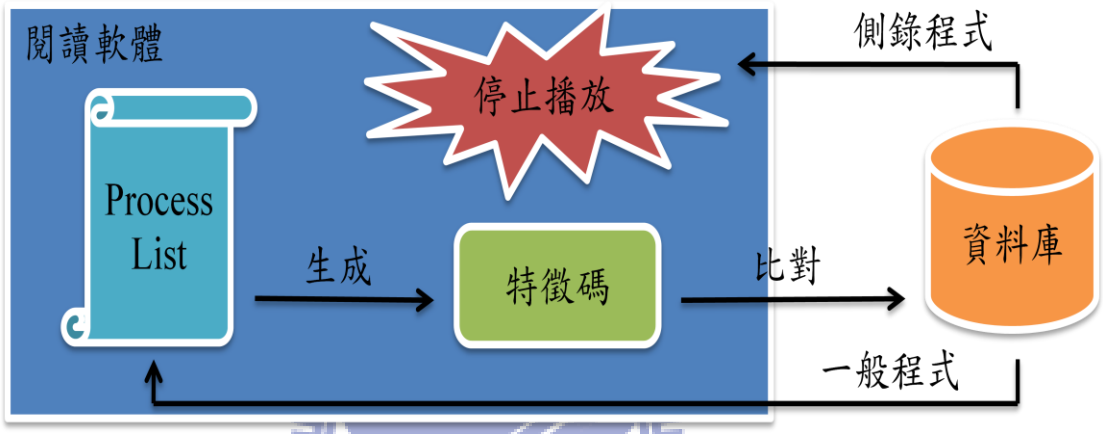


圖 16 BestWise 防側錄模組作法

2.1.4 BestWise 之問題

BestWise 的防側錄模組最主要是利用特徵碼技術，並與資料庫進行比對而達到防側錄之目的。許多防毒軟體也是利用相同的手法進行防毒，例如:病毒碼掃毒法。病毒碼掃毒法是將新發現的病毒加以分析後，根據其特徵，編成病毒碼，加入資料庫中。以後每當執行掃毒程式時，便能立刻掃描程式檔案，並作病毒碼比對，即能偵測到是否有病毒。因此，病毒攻擊者衍生出許多避免被病毒碼掃毒法偵測的技術，其中最為廣泛的就是特徵碼修改技術。

特徵碼技術主要是對執行檔的內容作運算而產生特徵碼。因此，特徵碼修改技術也就是對執行檔的內容作修改，進而產生不同的特徵碼，使該執行檔之特徵碼不再是只有唯一。特徵碼修改技術[17]的作法十分多樣化，這邊只介紹幾種，如:修改入口位址法、增加無意義指令法、加殼或加偽裝殼法等等...。

因此，BestWise 的防側錄模組也有相同的問題，當防側錄程式之執行檔遭受竄改之後，便可以躲過特徵碼資料庫之比對，進而可以成功側錄受保護文件之內容，導致失去保護文件之目的。

2.2 相關技術介紹

此章節將詳細說明幾項在本研究會使用到的技術。首先是本模組所需要的特徵碼之相關技術，例如: PE File 檔案格式的說明、MD5 雜湊演算法。另外，也將介紹 Windows Service 技術[12]，此技術將用於開發本研究的防側錄模組。

2.2.1 PE File 檔案格式說明

PE 為 Portable Executable 之縮寫，為 Microsoft Windows 中可執行的二進位碼，例如: EXE[2]和 DLL[3]檔案。此格式為 Microsoft 所設計，並在 1993 年由 TIS(Tool Interface Standard)標準化。在 B. Luevelsmeyer 之【PE 檔案格式 1.9 版】[1]中對 PE File 的檔案架構有完整的說明，圖 17 為 PE File 的檔案格式。

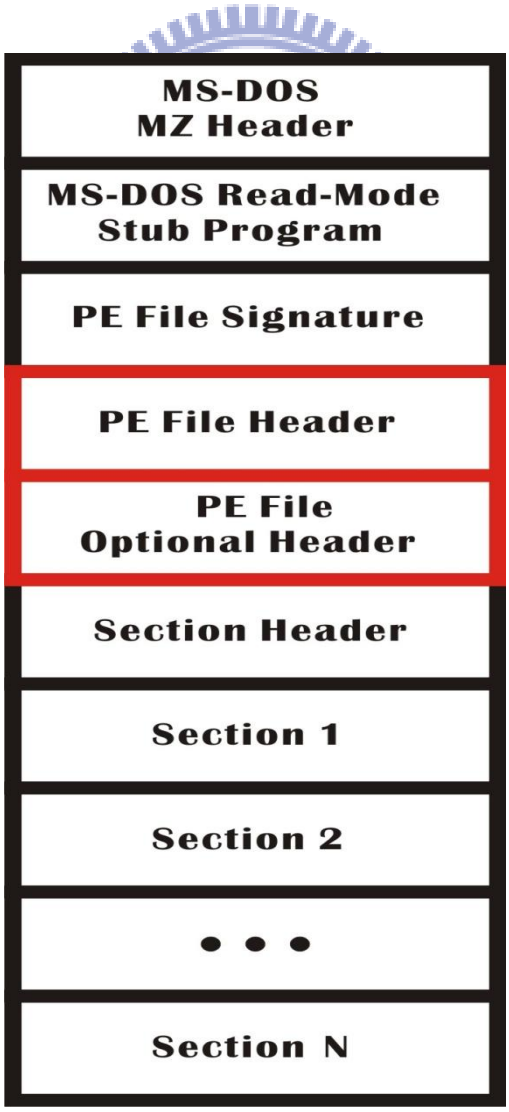


圖 17 PE File 的檔案格式

在圖 17 的檔案格式說明圖中，DOS MZ Header 為 IMAGE_DOS_HEADER 結構，前兩個位元組必須為 MZ 開頭，用於檢查是否為 Windows 的執行檔。接下來的 DOS STUB 內容代表用於 OS/2 可執行檔、自解壓縮檔和其他程式。對於 PE File，它是 DOS2 相容可執行檔，包含 100 位元組的內容，若是執行時的作業系統不相容，則會輸出一個錯誤訊息:"this program needs windows NT"。PE Header 中則記錄被仔入記憶體時所會用到的相關訊息。當程式開始執行時，PE 裝載器將從 DOS MZ Header 中找到 PE Header 的起始偏移量，進而跳過 DOS stub 直接定位到真正的 PE Header，並且從 PE Header 之後為該檔案實際內容。

圖 18 為使用 Ultra Edit 實際開啟一份可執行檔案，並以 16 近位顯示其內容，圖中有標示 1 的地方，即為該檔的 PE Header 位置，從 PE Header 之後為該檔案實際內容。

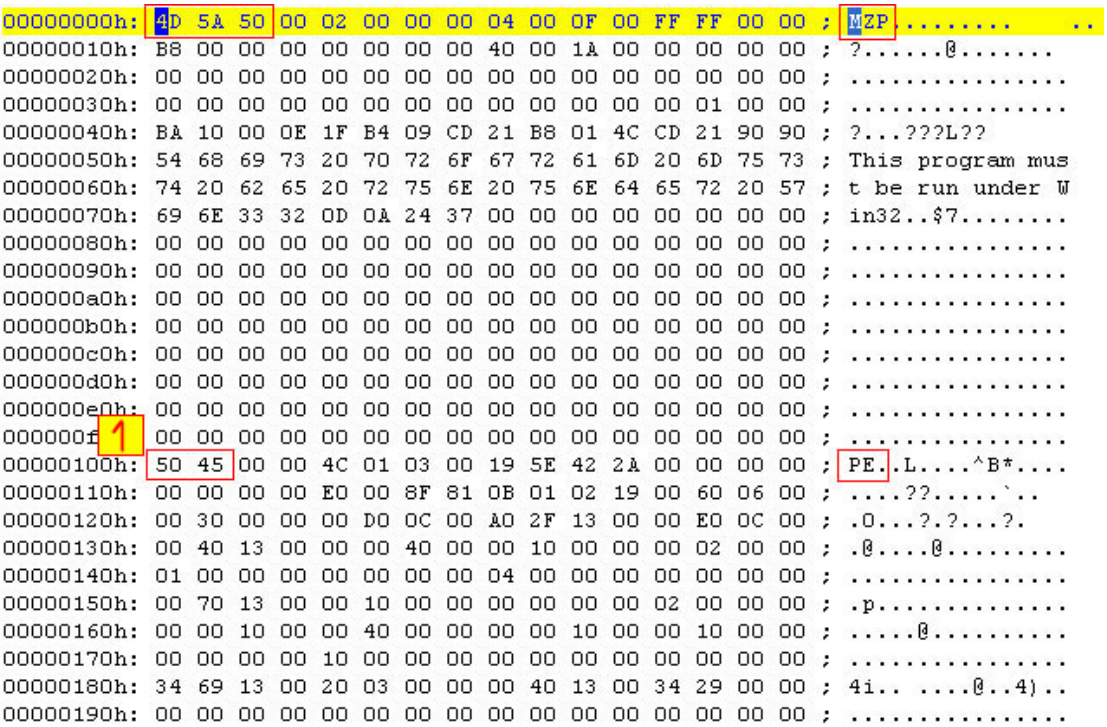


圖 18 以 Ultra Edit 開啟可執行檔案內容

2.2.2 MD5 雜湊演算法

MD5 (Message-Digest algorithm 5)雜湊演算法可將任意長度的資料，以 MD5 雜湊演算法運算，得到一組固定長度為 128 位元的結果。

- 雖然不同的資料經由 MD5 雜湊演算法計算所得到的結果有可能相同，但是根據統計顯示，重覆的機率低於百萬分之一。

- MD5 為一個單向雜湊演算法，亦即不易以逆向運算得到原始資料，例如：要計算字串 abc 的 MD5 結果很簡單，但是要將 MD5 計算後的結果逆向運算得到 abc 卻相當困難。

本研究將採用 MD5 演算法，將取得的資料雜湊計算出一個固定長度的字串，當作代表該側錄程式的唯一特徵碼，未來若有更高安全性的考量，可改用 SHA-1、或 SHA-2 雜湊演算法。

2.2.3 Windows Service

在 Windows 作業系統中，Windows Service 為可以長時間在 Session 0 之下運作且不需要與使用者互動，並且提供特殊功能之具備管理者權限的 Windows 程式。Windows Service 可以設定在作業系統開啟時開始運作，或者是由使用者手動啟動它，因此它可以在使用者為登入之前就開始運作。

Session[15]為由已登入使用者的 processes 和 system objects 所組成。當使用者登入作業系統直到該使用者登出作業系統的這段期間都是屬於 Session。Session 底下的架構依序為 Windows station, Desktop, User application，如圖 19。User applications 由 Desktop 所管理，例如:管理 User application 在桌面上的相關訊息。Desktops 由 Windows station 所管理，例如:管理不同 Desktop 之間的切換。Windows stations 由 Session 所管理，例如:管理在不同 Windows station 下的程式之間的溝通。Session 1 是第 1 個登入者所對應之 Session，Session 2 是第 2 個登入者所對應之 Session，以此類推。但是 Session 0 是保留給 System process 和 Windows Service 使用。

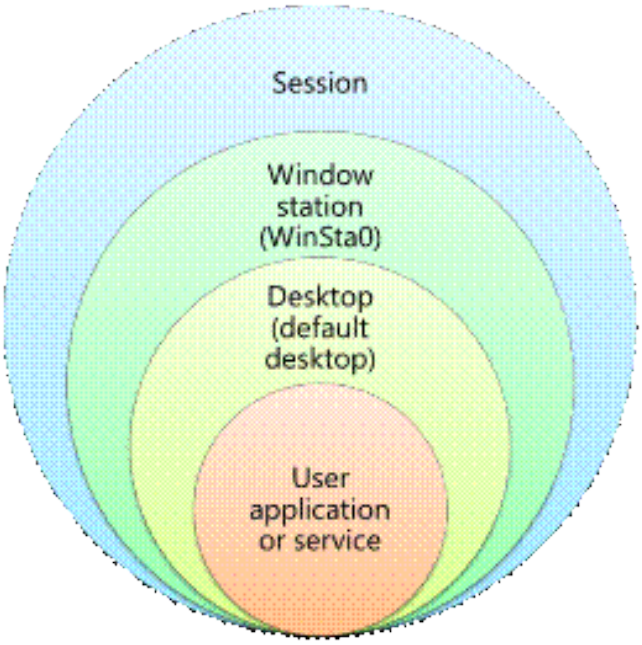


圖 19 Session 架構圖

Windows Service 的管理與其他 Windows 程式不同，因為 Windows Service 不與使用者互動，所以 Windows Service 的管理皆是由 Service Control Manager (SCM)所管理。可以透過 SCM 來 install, start, stop, pause, 或 restart Window Services，或是設定自動啟動或是手動啟動 Window Services。

因為 Windows Service 的權限與一般 Windows 程式不同，導致會成為那些正在伺機提升自身權限級別的惡意代理的目標，所以 Microsoft 設計了 Session 0 Isolation 技術[18]，用於提高 Windows Service 的安全性。在 Windows 作業系統之下，不同程式之間的互動必須藉由 Windows API 才能完成，例如:FindWindow, SendMessage 等等 Windows API。因此，Session 0 Isolation 技術主要就是禁止不同 Session 之下的程式利用 Windows API 來進行互動，使得 Windows Service 與一般的視窗程式隔離，以提高安全性，Session 0 Isolation 的示意圖如圖 20。但是可以利用其他的作法進行互動，例如:Named Pipe, Sockets 等等的 Inter-process communication (IPC) 作法。

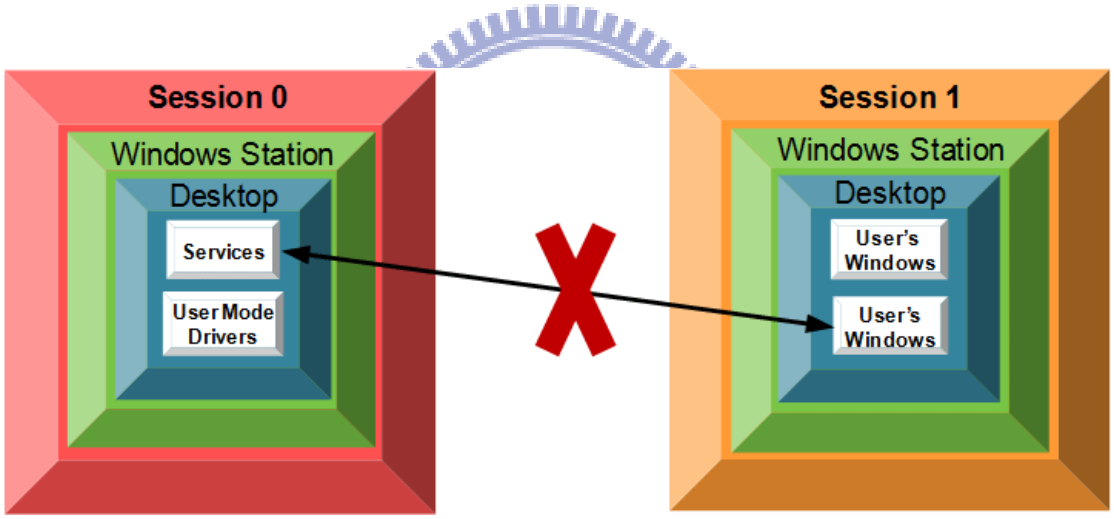


圖 20 Session 0 Isolation 示意圖

三、 防側錄系統分析

在本章節會針對本研究所實驗的對象的系統架構作說明，以及說明夠模組的運作方式，接著以這些模組之分析為基礎來建構防側錄系統，對本研究的系統架構、功能模組之間的溝通，以及系統流程做詳細的說明，並且說明防側錄模組在功能上的設計考量。

3.1 BestWise 防側錄系統

先了解實驗對象的系統架構與該防側錄模組的運作流程，並加以分析。

3.1.1 系統架構

BestWise 防側錄系統的系統架構如圖 21 所示，其中包含製作端、伺服器端與閱讀端。

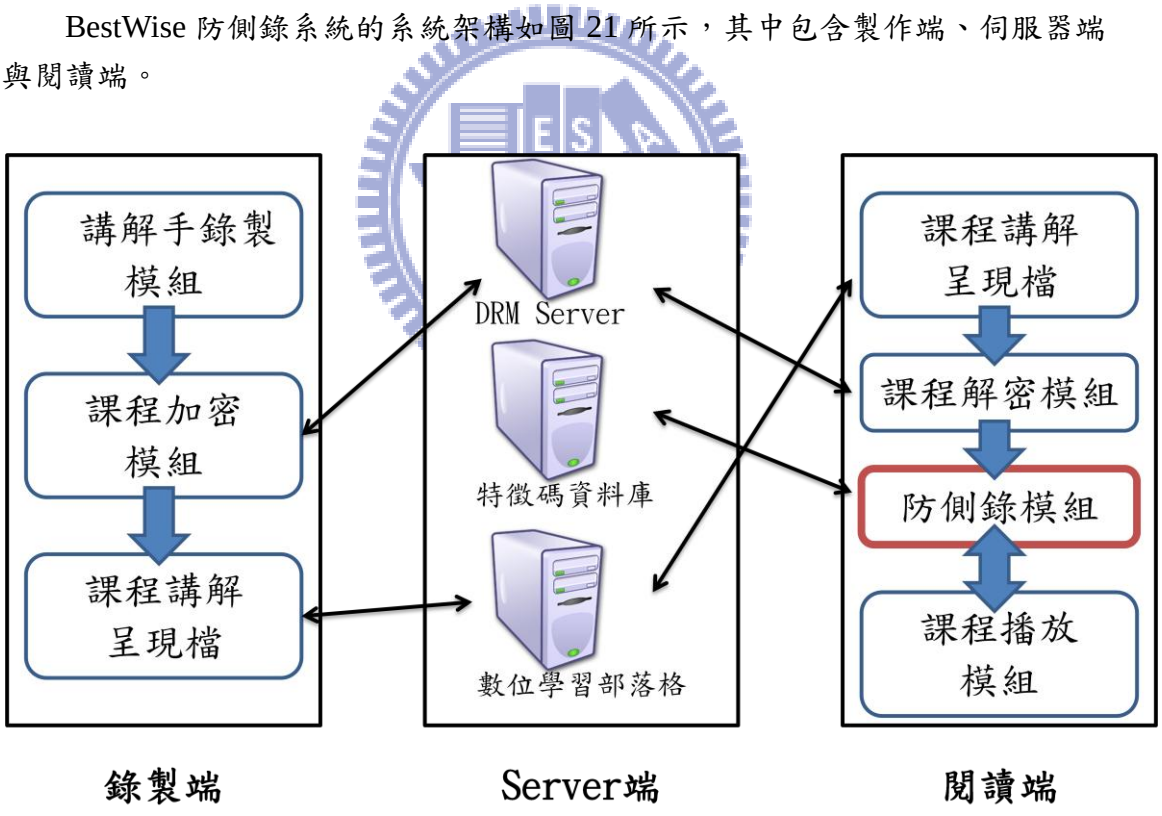


圖 21 BestWise 系統架構

- 「製作端」

使用者可使用由該公司所提供的講解手錄製工具，利用其中的錄製模組對多媒體檔案進行課程內容之錄製，並可透過其中的課程加密模組與 Server

端的 DRM Server 取得加密金鑰，利用該金鑰對課程內容進行加密。最後加密完成之後，即可產生一份受 DRM 保護的加密課程。

- 「伺服器端」

主要分為 DRM Server、特徵碼資料庫與數位學習部落格三個部分。DRM Server 主要是對課程權限作管控，例如:課程的閱讀時間、閱讀電腦的 IP Range 限制等等功能。當使用者開啟課程內容時，必須經過 DRM Server 的身分以及權限認證後，才能開始閱讀課程內容。特徵碼資料庫是用來記錄每一個桌面或視窗側錄程式所產生的唯一特徵碼，並可由 Web 使用者介面進行新增與設定，方便建立資料於特徵碼資料庫。數位學習部落格是提供使用者將課程上傳與其他使用者分享的地方。

- 「閱讀端」

閱讀端為一課程播放器，此課程播放器是配合智勝國際科技公司[8]所開發的講解手 Player，當中包含三個模組:課程解密模組、防側錄模組與課程播放模組。

當使用者從數位學習部落格中開啟一份受到保護的課程時，課程解密模組會先與 DRM Server 做連線，而 DRM Server 會要求使用者的驗證資訊，一旦身分與使用權限確認通過，DRM Server 就會傳送解密金鑰給課程解密模組，此時課程解密模組便可進行解碼之動作。而解密之後的課程內容會傳送給課程播放模組，但在傳送之前會先啟動防側錄模組。防側錄模組開始列舉使用者電腦中所有正在執行的處理程序，並加以進行比對之工作。一旦發現符合特徵碼資料庫中的項目時，要求播放棄停止播放課程內容，接著會送出警告訊息要求使用者關閉側錄程式才能繼續觀看課程內容，同實隱藏播放區域直到使用者關閉側錄程式。等待使用者關閉側錄程式之後，會通知播放器恢復播放狀態，如此已達到防側錄效果。

3.1.2 防側錄模組運作流程

在此章節將說明閱讀端的防側錄模組運作流程。為了可以即時更新特徵碼比對資料庫，因此，在防側錄模組開啟之前，會先透過網路檢查特徵碼資料庫是否有更新，若有更新，則會自動下載並儲存到 Local 端。另外，為了考慮到 Local 端可能會有離線瀏覽的狀態，因此，在防側錄模組第一次安裝的時候，也同時將 Server 端的特徵碼資料檔複製到 Local 端中，以提供 Local 端在沒有網路連線的

狀態下使用。

Local 端特徵碼資料庫的更新時機，有以下兩種情形：

1. **Local 端完全沒有任何資料庫暫存檔**：

   在沒有任何資料庫暫存檔的情況之下(可能被使用者刪除之情況)，課程開啟時，防側錄模組會要求課程播放器馬上停止，並且隱藏播放課程，直到資料庫檔被下載及安裝到 Local 端。

2. **Server 端資料庫內容有更新**：

   Local 端所下載的資料庫暫存檔中有記錄該檔案的建立時間與修改時間，因此當此記錄時間與 Server 端的記錄時間有所不同時，即代表兩者之間的資料庫內容有所差異，此時會再次下載及更新 Local 端的資料庫暫存檔。

另外，為了避免在 Local 端的資料庫暫存檔被任意開啟與修改，因此，資料庫暫存檔都會使用加密演算法進行保護。

接下來，在圖 22 中將說明當閱讀端啟動防側錄模組之後的流程，此模組會列舉 Windows 中目前所有正在執行的處理程序(Process)，並且利用上次偵測時的程式數目與當下偵測實的程式數目進行比對，檢查是否有新開啟的程式，借由此作法減少不必要的防側錄偵測。再透過模組中的特徵碼產生演算法取得每一個處理程序的唯一特徵碼，並記錄在自定義的資料結構中。接下來將利用每一個處理程序的唯一特徵碼與特徵碼資料庫中的項目一一做比對，若發現有符合的項目，則代表是側錄程式，立刻發出訊息通知課程播放器停止播放，並隱藏課程內容，直到下一次的比對程序中沒有發現任何的側錄程式，才會重新開啟播放並顯示課程內容。

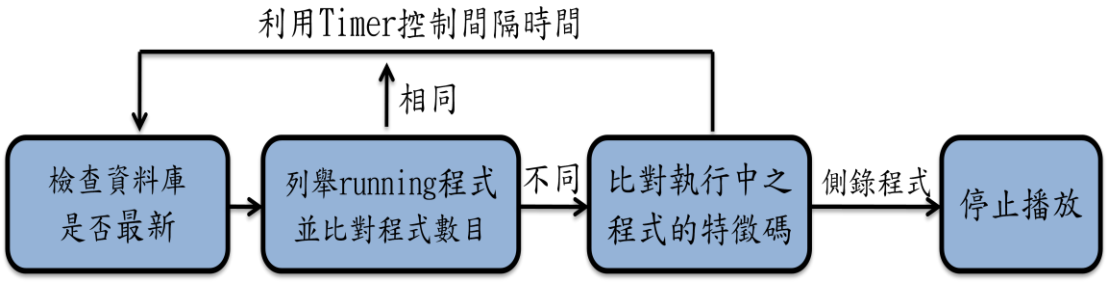


圖 22 BestWise 防側錄模組之流程

此防側錄模組存在兩個足以影響防側錄偵測之結果的問題。首先是在比對程式數目，當程式數目相同時不代表沒有新開啟的程式，可能在兩次偵測之間發生

某一個程式被關閉與另外一個程式被開啟，造成這兩次偵測的程式數目仍然相同，使得新開啟的程式躲過防側錄模組之偵測，此問題本研究稱之為數目假象。另外一個問題就是特徵碼竄改，側錄程式可以藉由竄改本身的執行檔內容，使得防側錄模組對它所產生的特徵碼與資料庫中的特徵碼不同，進而躲過防側錄模組之偵測。

3.2 防側錄模組設計考量

在前章節中提到數目假象與特徵碼竄改之問題導致防側錄模組之偵測受到影響，為了避免數目假象以及解決特徵碼竄改之問題，將重新設計防側錄模組，因此，本章節將說明本研究之防側錄模組的設計考量。

首先，為了避免數目假象之問題新增比對的項目，當比對程式數目之結果是相同時，再進一步比對其他項目，以確保真的沒有新開啟之程式，避免側錄程式利用此問題而躲過防側錄偵測。

接下來，為了解決特徵碼竄改之問題，將設計一套雙重偵測機制來補強此問題，並且提高防側錄模組之辨識能力。雙重偵測機制分成靜態偵測機制與動態偵測機制，靜態偵測機制主要是用來辨識程式之身份(一般程式、可疑程式、側錄程式)，再配合動態偵測機制對可疑程式進行分析，分析該程式是否有側錄之行為。

因為特徵碼竄改之問題主要是藉由產生不同的特徵碼來躲過特徵碼資料庫的比對，因此，在特徵碼資料庫比對的部分新增其他代表該側錄程式的相關訊息，當該側錄程式之特徵碼不同時，利用該側錄程式的相關訊息比對，進而達到辨識之目的。因為只利用側錄程式的相關訊息無法完全判定一個程式就是側錄程式，因此將原本的身分定義新增為：一般程式、可疑程式、側錄程式，可疑程式就是利用側錄程式的相關訊息比對後，相似度較高的程式。判別出可疑程式之後，為了確定該程式有側錄之行為，則利用 Windows Hook 技術[4]攔截該程式之 Message，進而分析該程式之行為，若分析出有側錄之行為，則如同處理側錄程式之做法處理之。

最後，為了使防側錄模組在 Windows 7 的 User Account Control 機制下能夠取得其它程式的相關訊息，以供防側錄偵測使用，因此，需要具備管理者權限以取得其它程式的相關訊息進而偵測其它程式之行為，所以將防側錄模組以 Windows Service 之型態運作，以滿足此需求。

3.3 雙重偵測防側錄系統

此防側錄系統是依據上章節所提出的設計考量而命名，在此章節將說明重新設計過後的系統架構，以及該防側錄模組的運作流程，與雙重偵測機制的流程，最後說明此防側錄模組之特性。

3.3.1 系統架構

重新設計過後的系統架構因為防側錄模組的運作方式不同，使得防側錄模組不在閱讀端中，而是獨立以 Windows Service 之型態安裝在使用者電腦中。因為防側錄模組獨立於閱讀端之外，因此防側錄模組與閱讀端之間的溝通則利用雙方協議好的 Pipe 管道進行溝通。當閱讀端需要防側錄模組進行防側錄偵測時，就傳送要求開始偵測的訊息給防側錄模組；當防側錄模組有偵測到側錄程式時，才會再回傳要求停止播放的訊息給閱讀端。系統架構如圖 23 所示。

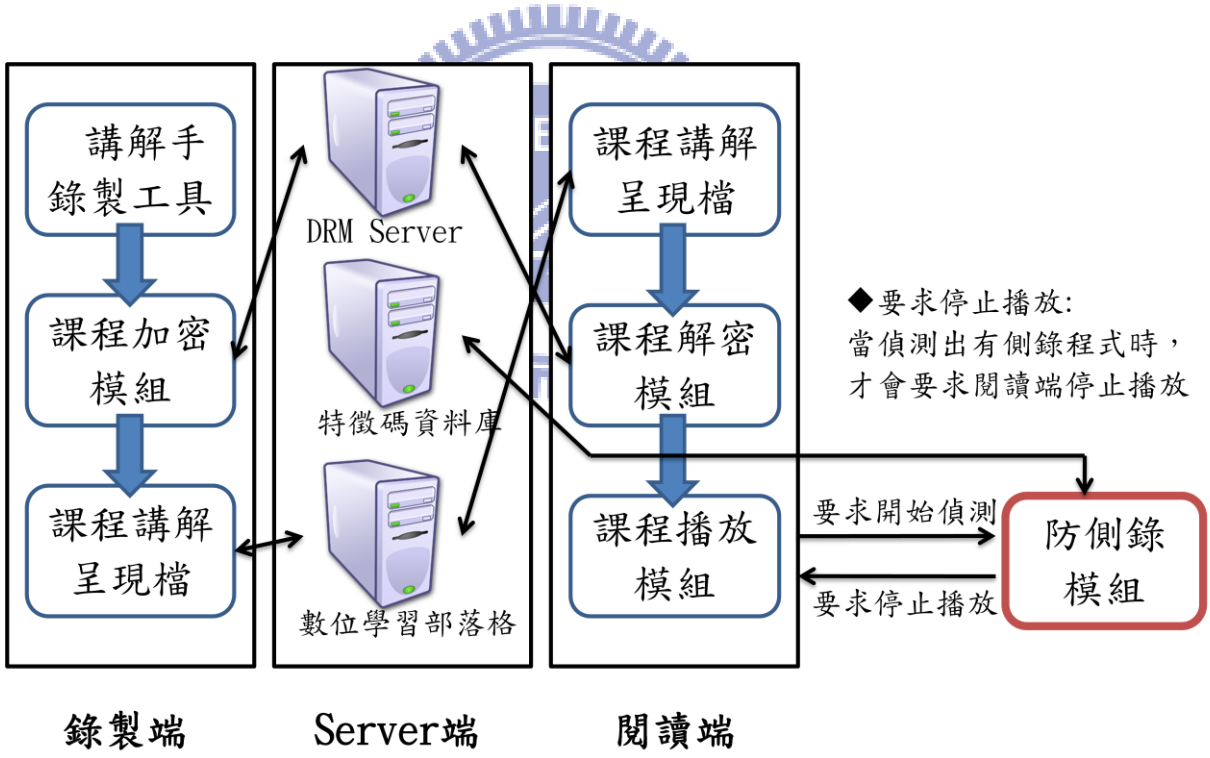


圖 23 雙重偵測防側錄系統架構圖

3.3.2 防側錄模組運作流程

在此章節將介紹防側錄模組的運作流程，因為加入了雙重偵測機制、改良數目假象之機制與 Windows Service 之型態，使得運作流程大為改變，在此章節會說明其中之不同。

閱讀端送出要求開始偵測的訊息給防側錄模組之後，防側錄模組就開始進行防側錄偵測。首先，為了減少不必要的偵測，先列舉出所有系統中正在執行的程式，並且進行比對本次偵測時的程式數目與上次偵測實的程式數目，以檢查是否有新開啟的程式，若為相同則進一步比對程式名稱以確保真的沒有新開啟的程式。若是不相同，則利用靜態偵測機制進行辨識程式之身份再加以處理，判別是一般程式、可疑程式或側錄程式之後，在作出對應之處理，對於側錄程式就要求閱讀端停止播放與隱藏播放內容，而可疑程式之處理則利用動態偵測機制作進一步的行為分析。動態偵測機制是利用 Windows Hook 技術去攔截可疑程式所收發的訊息，並由這些訊息去分析該程式是否有側錄行為，若有側錄行為代表該程式為側錄程式，再依照處理側錄程式之作法處理之。此防側錄模組之運作流程如圖 24 所示。

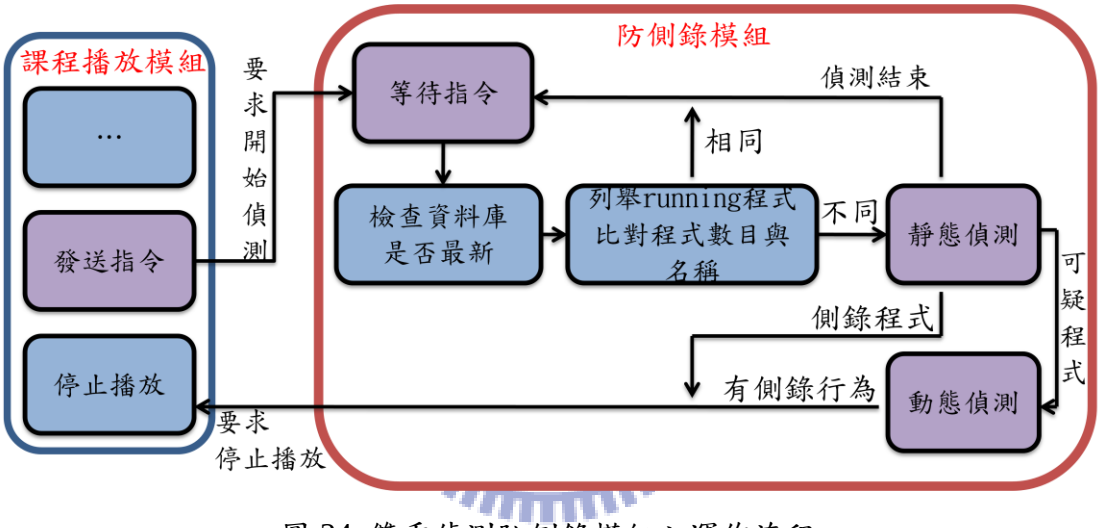


圖 24 雙重偵測防側錄模組之運作流程

3.3.3 雙重偵測機制之流程

此章節將介紹靜態偵測機制與動態偵測機制的內部流程。

首先，將介紹靜態偵測機制之流程。先製作一份 Process List 去儲存目前所有正在執行的程式，再由此 Process List 中取出目前系統中正在執行之程式，並由該程式之執行檔取出部分內容透過 MD5 雜湊演算法進而產生特徵碼，同時取出該程式其他相關訊息用來辨識該程式之身份，再依據辨識之結果作相對應之處理。

接下來，將介紹動態偵測機制之流程。借由靜態偵測機制而辨識出可疑程式，再利用 Windows Hook 技術攔截該程式之 Message，當該程式送出或接收 Message

時，都會被動態偵測機制所攔截，進而分析該程式是否有側錄之行為，若有分析出有側錄之行為則馬上更新特徵碼資料庫，並且傳送要求停止播放之訊息給閱讀端。圖 25 說明靜態偵測機制與動態偵測機制的互動流程。

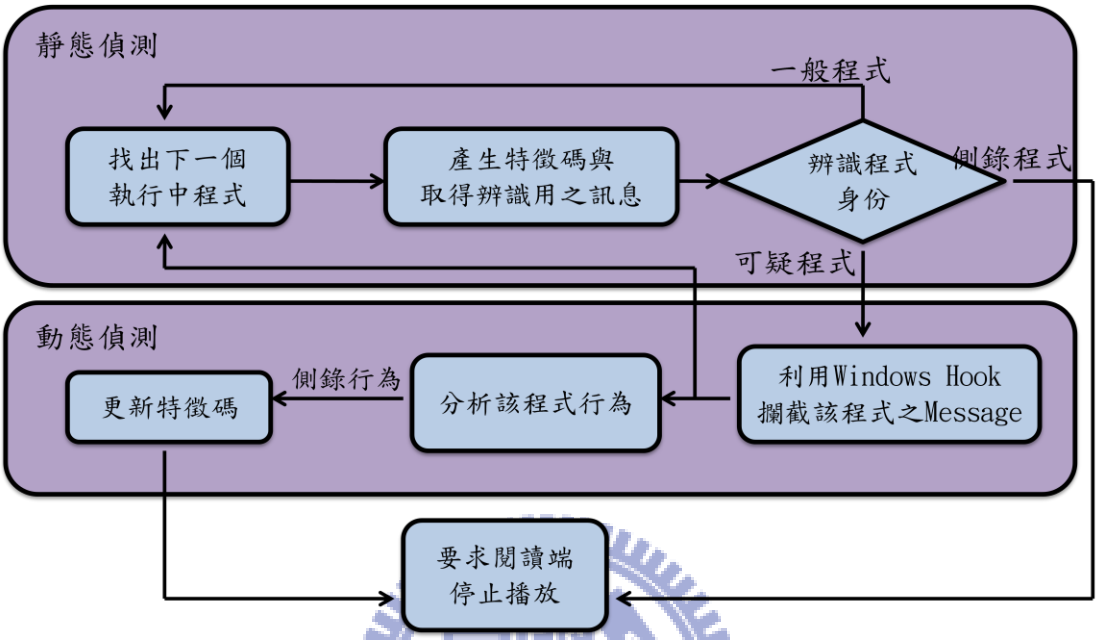


圖 25 雙重偵測機制之流程

3.3.4 雙重偵測防側錄模組之特性

此章節章介紹本研究之防測錄模組的特性，並加以說明。

- **高權限：**

  在 Windows 的 UAC 機制之下，當側錄程式取得高權限時，防測錄模組必須能夠正常運作。

- **高安全性：**

  因為防側錄模組具有高權限，可能會被那些正在伺機取得高權限的惡意程式視為攻擊目標，因此借由 Windows Service 的安全機制-Session 0 isolation 以保護防測錄模組不被攻擊。

- **即時偵測：**

  借由傳送與接收訊息的機制，使防測錄模組能配合課程播放模組在特定的時機立即偵測，例如：開始播放時、拖拉播放時間列等等...。

四、 防側錄模組實作

4.1 利用 Windows Service 實作防側錄模組

4.1.1 Windows Service 程式之開發流程

本章節將會介紹一般的 Windows Service 的開發流程，並依據此流程來實作防測錄模組。圖 26 為 Windows Service 的開發流程，步驟如下：

1. **撰寫 Windows Service 程式**
   在 Windows Service 程式中撰寫所想開發的功能，以及撰寫處理來自 Service Control Manager (SCM)之指令的 Function。

2. **撰寫 Install 程式**
   將 Windows Service 程式向 SCM 進行 Install，其中必須指定所要 Install 的 Service Name、Service Path 等等的相關設定。

3. **撰寫 Start 和 Stop 程式**
   透過 SCM 開啟或關閉 Windows Service 程式，其中必須指定所要開啟的 Service Name。

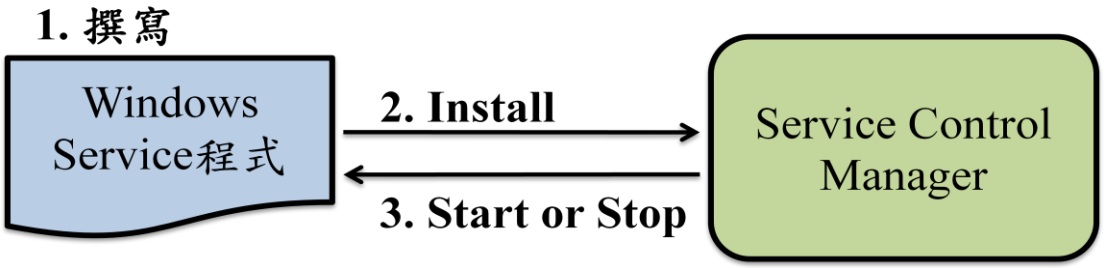


圖 26 Windows Service 之開發流程

4.1.2 防側錄模組運作說明與示意圖

本章節將在利用 Windows Service 實作防測錄模組，並且撰寫處理來自 Service Control Manager (SCM)之指令的 Function，用來控制 Windows Service 的運作，以及實作 Install、Start 和 Stop 程式。

首先在 Windows Service 的 Main Thread (在此稱之為 Service Thread)中進行 Service 的初始化設定，其中包括註冊用來處理來自 SCM 的控制命令的 Service Handle function，需要指定 Service Name 和 Service Handle function。接下來，Service Thread 會 Create 一個新的 Thread 用來執行防測錄模組之功能，稱之為 Work Thread，而 Service Thread 只要負責處理來至 SCM 的控制命令即可。Work Thread 會再 Create 一個新的 Thread 用來接收與傳送訊息給閱讀端，稱之為 Pipe Thread，Work Thread 依據 Pipe Thread 所接收到的訊息做出不同的對應處理，例如：開始偵測、處理側錄程式。最後，當 Service Thread 收到 SCM 之 Stop 控制命令則會 kill 在其底下的 Child Threads 再 Exit。利用 Windows Service 實作防側錄模組程式之示意圖如圖 27 所示。

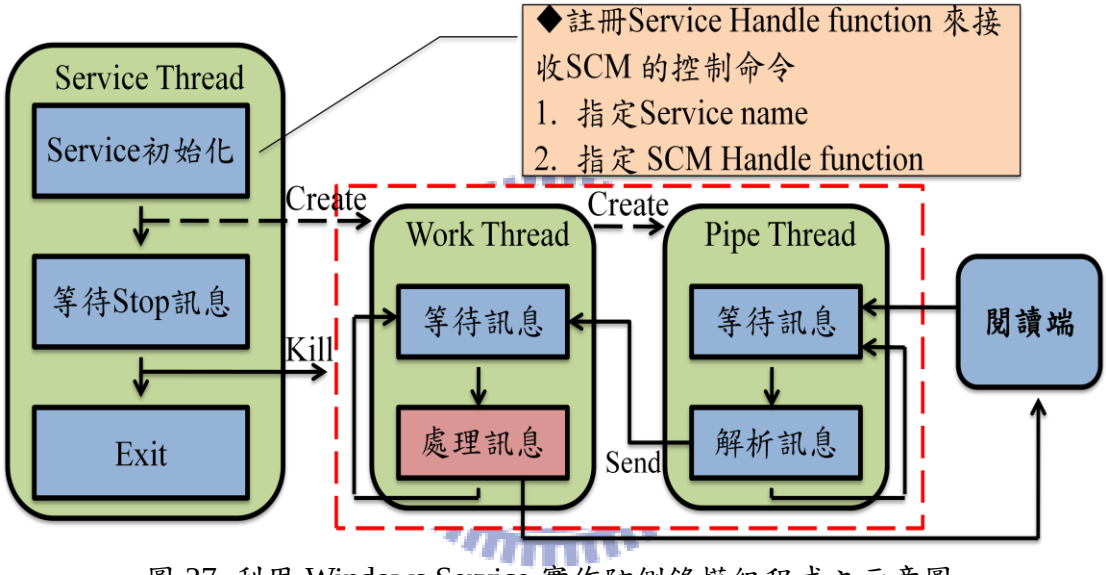


圖 27 利用 Windows Service 實作防側錄模組程式之示意圖

4.1.3 防側錄模組程式之 Installation

因為 Windows Service 程式大多不與使用者互動，與一般程式不大相同，它必須由 SCM 作管理。因此，將防側錄程式撰寫完成之後，必須將防測錄模組 Install 至 SCM 的 Database 中。作法如下:

1. **Create SCM Object**

   SCM Object 用來定義本次 SCM 命令 (Installation)的權限，因為要建立 Service，所以必須設定為可建立 Service (Insert Service Object)的權限，此權限參數為 2，並且建立 Service 需要管理者權限。

2. **Insert Service Object (防側錄模組)至 SCM**

   在此步驟中必需要設定幾個重要的參數: Service Name, SCM Object, Service 執行檔路徑。

3. **將 Service 設定為自動啟動**

   此步驟是為了使防測錄模組在作業系統開啟之後就能馬上運作，不必等到使用者登入。

4.1.4 側錄模組程式之 Start 和 Stop

當防側錄模組程式成功地Install至SCM中之後,雖然已經設定為自動啟動，但是必須等到作業系統重新開機才會開啟，所以必須先借由 Start 程式將防側錄模組開啟，也可利用 Stop 程式將防測錄模組關閉，但是以上之動作皆需要管理者權限才能正常運動。作法如下:

1. **Create SCM Object**

   SCM Object 用來定義本次 SCM 命令 (Start or Stop)的權限，因為要連上 SCM 並進行操作，所以必須設定為可連上 SCM 並操作 Service(Open Service)的權限，此權限參數為 1。

2. **Open SCM 中的 Service**

   在此步驟中必需要設定幾個重要的參數: Service Name, SCM Object。

3. **啟動 Service 或送出停止訊號給 Service**

4.2 靜態偵測之實作

在 3.3.3 節有先提過靜態偵測的運作流程，因此在此章節將介紹 Process Info List 的取得方法，以及取得之後的特徵碼實作與介紹，最後利用實作出來的特徵碼與該程式的相關資訊進行評分比對，以辨識該程式之身份。

4.2.1 Process Info List 的取得方法

目的是為了取得所有執行中程式的相關資訊，進而產生代表該程式的特徵碼，作法如下:

1. **使用 Windows API 列舉出正在執行的 Processes**

   Windows API : EnumProcesses。

2. **取出 Processes 的資訊**

   所取出的資訊有 Name, Path, Process id, Thread id 等等…。

3. **排除非屬於使用者之 Processes**

   由取出的資訊中的 Process Domain Name 可能為 SYSTEM, NETWORK SERVICE, LOCAL SERVICE 或是使用者 ID，因此，可透過 Process Domain Name 將非使用者的 Processes 排除。

4. **利用 Processes 之資訊製成 Process Info List**

   將 Processes 之資訊存在 Process Info List 中，以便特徵碼實作。

4.2.2 特徵碼介紹與實作

在實作特徵碼之前，必須先了解 Windows 的執行檔格式，才能有效的取出該程式的相關資訊。在 2.2.1 節中有先介紹過 PE File 檔案格式，在此只簡單說明其中的部分格式。

- **MS-DOS MZ Header**

  在此 Header 中可以發現一開始的前兩個位元組是“MZ”開頭，可用來檢查是否為 Windows 執行檔，並且在此 Header 中可以找到 PE Header 的起始偏移量。

- **PE File Header**

在此 Header 中可以發現一開始的前兩個位元組是“PE”開頭，代表在此兩位元組之後為該程式的真實內容，包括該程式的相關資訊與程式碼。

- **PE File Optional Header**

在此 Header 中也儲存其他相關資訊，其中最重要的是可以找到此檔案的 Entry Point，找到 Entry Point 即可找到該程式碼的真實位置。

接下來，將介紹如何利用 PE File 檔案格式來實作特徵碼，如圖 28 所示，並且作法如下:

1. **由 MS-DOS MZ Header 中找出 PE File Header 的起始偏移量**
2. **由 PE File Header 往後位移一段距離**
3. **開始取該檔案的部分內容**
4. **計算該檔案真實內容的大小**
5. **將檔案部分內容與檔案大小進行 MD5 雜湊演算法**
6. **產生唯一特徵碼**

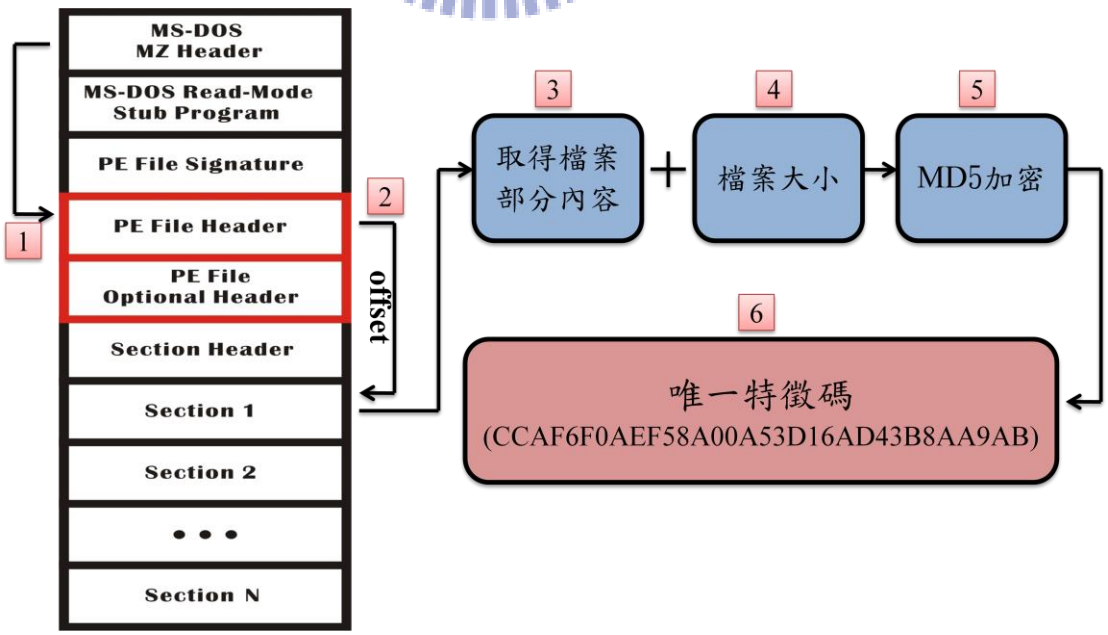


圖 28 特徵碼實作之流程圖

4.2.3 Process 身分偵測

在此章節將說明如何利用上章節所產生的特徵碼以及由 PE File 檔案格式中取的的程式資訊，進行 Process 身分偵測。Process 身分偵測主要是利用評分機制來辨識該程式之身份，由比對特徵碼與程式資訊來進行評分。

- **評分項目**
  - 檔名(5 分)
    - 檔名相同時再比對檔案相關資訊(每項 1 分)
      檔案相關資訊由 PE File Header 中取出，其中的項目例如: Number Of Sections, Size Of Code, Base Of Code, Entry Point 等等…。
  - 特徵碼(10 分)
- **評分標準**
  - 5 分以上列為**可疑程式**
    5 分以上代表該程式資訊與側錄程式資訊相似而特徵碼不同，所以將其視為可疑程式。
  - 10 分以上列為**側錄程式**
    10 分以上可能代表該程式的特徵碼與側錄程式相同，或是該程式的特徵碼雖然受過竄改，但是其程式資訊卻與側錄程式高度相似。

4.3 動態偵測之實作

動態偵測主要是希望透過分析程式行為，而判別程式是否有側錄行為，進而辨識出是否為側錄程式。所以一開始將先介紹側錄程式之原理，接下來將介紹 Windows Hook 技術以及如何使用該技術達到分析程式行為之目的。

4.3.1 側錄程式之原理

在 Windows 系統下的程式，大多都會接觸到 API 函式的使用，常用的 API 函式大約有 2000 個左右，而側錄程式要擷取桌面或視窗畫面必須借由 Windows API 才能進行擷取畫面之行為。主要的作法說明如下:

1. **側錄程式使用 Windows API**

2. **Windows API 會發送 DCI (Display Control Interface) command Message 給 System**

3. **System 收到 Message 之後，將影像複製到 Buffer 中**

4. **側錄程式再由 Buffer 取得影像**

因此，得知側錄程式的側錄行為會發送 DCI command Message 給系統，所以可以藉由攔截可疑程式的 Message 再加以分析，即可得知該可疑程式是否有側錄行為。

4.3.2 Windows Hook 之介紹

Windows 系統是利用 Message 作為其控制機制，系統程式與使用者程式可以通過 Message 為傳遞訊息給其他程式，並且系統和應用兩者都可以產生消息。而 Windows Hook 技術是 WINDOWS 提供的一種 Message 處理機制平台，是指在程式正常運行中接受或傳送 Message 之前預先啟動的 Hook 函數，用來檢查和修改傳給該程式或傳送給其他程式的 Message。

Windows Hook 技術之運作流程如下，圖 29 為利用 Windows Hook 攔截接收 Message 之示意圖。

1. **透過 System 將 Hook 函式寄生到 Target Process**

2. **Hook 函式攔截並處理 Target Process 要接收或送出的 Message**

3. **透過 System 將 Message 傳送給要接收的 Process 以完成接收或送出**

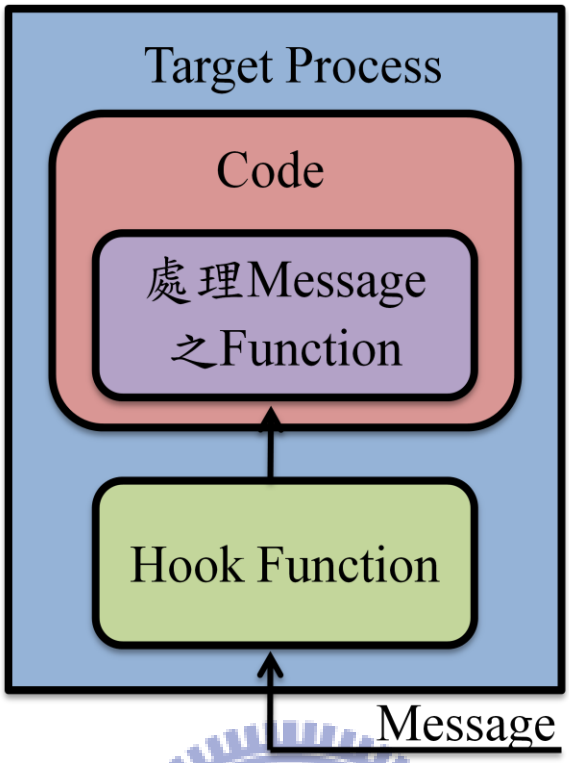


圖 29 利用 Windows Hook 攔截接收 Message 之示意圖

4.3.3 利用 Windows Hook 分析程式行為

此章節將說明如何利用 Windows Hook 攔截可疑程式之 Message，再加以分析其 Message，進而判斷是否有側錄行為。

以下將說明利用 Hook Function 攔截 DCI command Message 之作法:

1. **由 Process Info List 取得可疑程式的 Thread id。**

2. **取得 Hook Function Address。**

3. **委託 System 將 Hook Function 寄生到可疑程式。**
   並且必須告知 System 之訊息: Thread id, Function Address, Hook Module

4. **Hook Function 開始攔截可疑程式之 Message**

5. **若攔截到 DCICOMMAND 之 Message 透過 Named Pipe 通知防側錄模組有側錄行為**

五、 應用範例

在此章節將會展示如何應用雙重偵測防側錄模組對具備管理者權限之側錄程式和執行檔被竄改過之側錄程式進行偵測，並且包括說明課程保護的操作方式。

5.1 課程保護的操作方式

首先，先利用「錄製端」製作一份受 DRM 數位版權保護的課程內容。

Step 1、開啟講解手錄製工具，介面如圖 30。

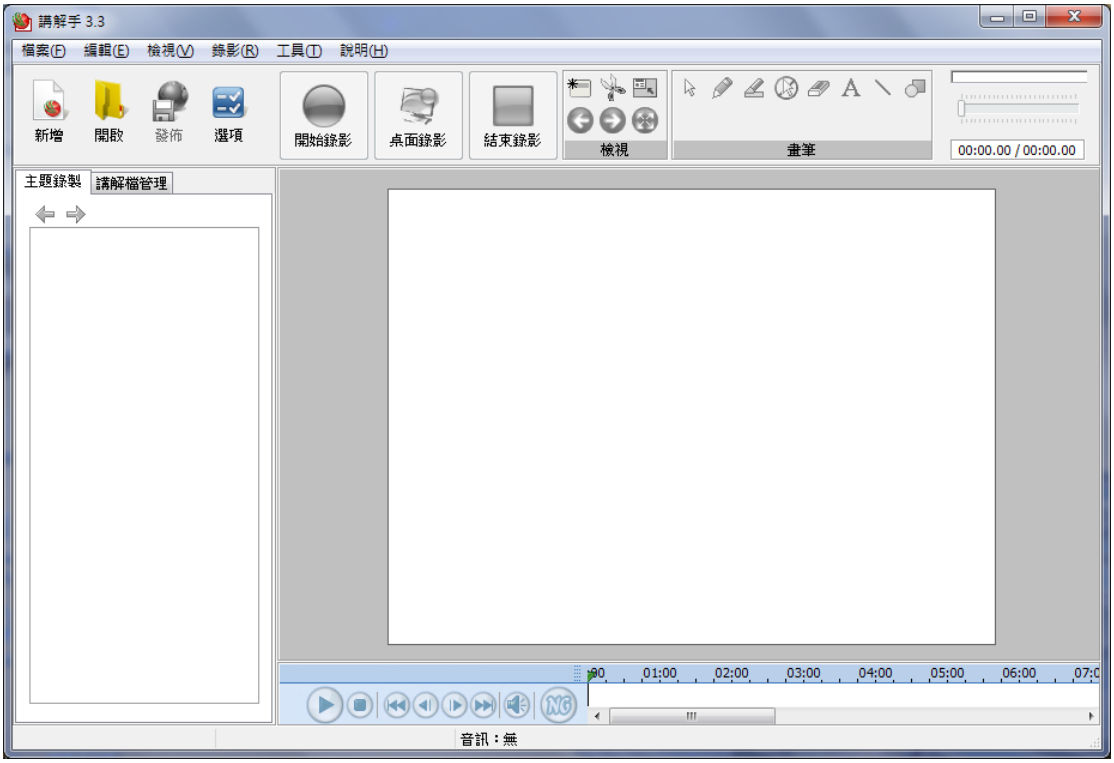


圖 30 講解手錄製工具

Step 2、點擊講解手錄製工具介面左上角的「新增」按鈕，開啟要用來錄製的教材檔案。如圖 32 所示。

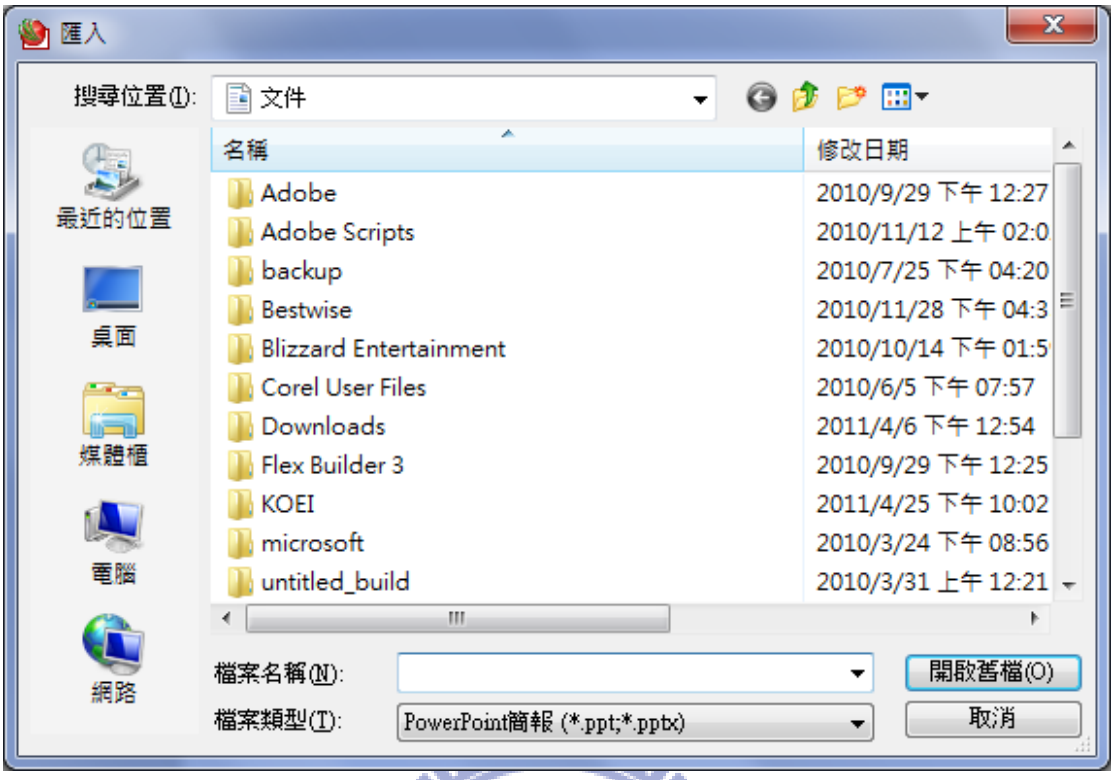


圖 31 匯入教材檔案

Step 3、完成匯入教材後，即可由介面上方的「開始錄製」按鈕開始錄製教材，如圖 32 所示。

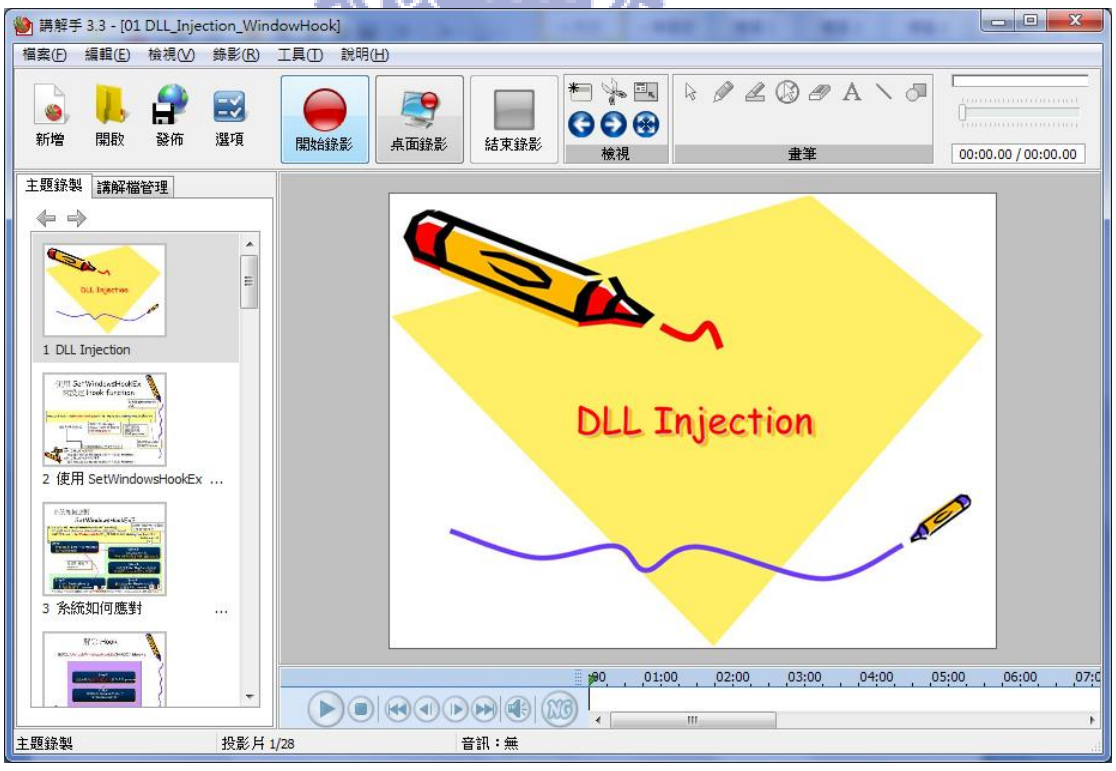


圖 32 完成匯入教材

Step 4、當錄製課程內容結束之後，再點擊介面左上角的「發佈」來產生課程檔案，並且可選擇是要上傳至部落格或是存放於本機端，如圖 33 所示。

圖 33 選擇要發佈的位置

Step 5、選擇好要發佈的位置之後，即開始設定課程內容的相關訊息，例如: 錄製者的資訊、面板格式、主題設定和 DRM 管理，如圖 34 所示。

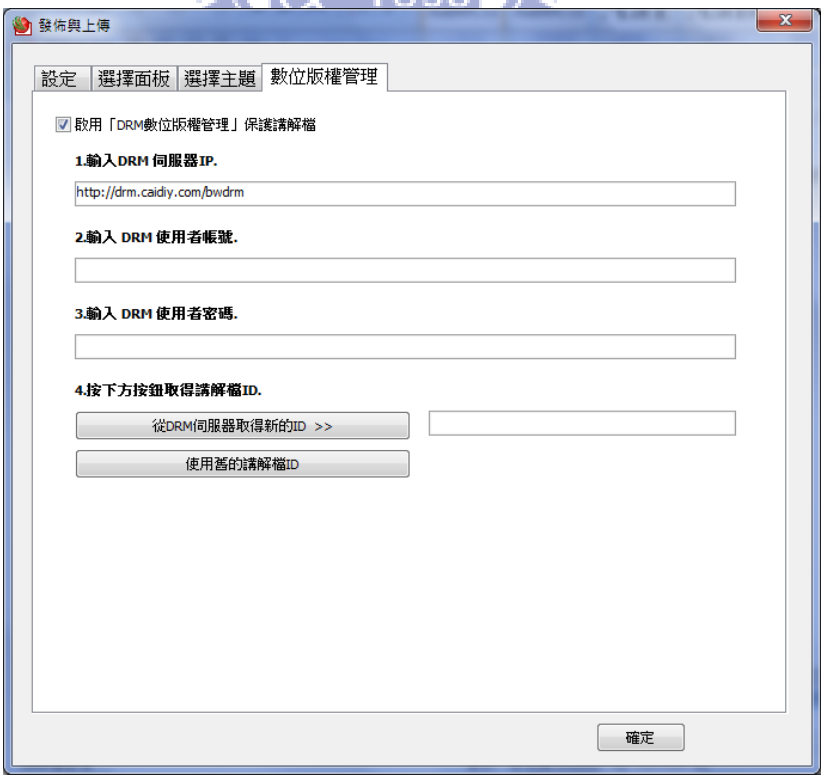


圖 34 課程檔案之相關訊息設定

Step 6、選擇透過 DRM 數位版權保護和課程檔案的儲存位置，即完成錄製課程。

Step 7、開啟剛剛錄製好的課程檔案，會自動開啟講解手瀏覽器來瀏覽課程內容，此時尚未開啟防測錄模組。如圖 35 所示。

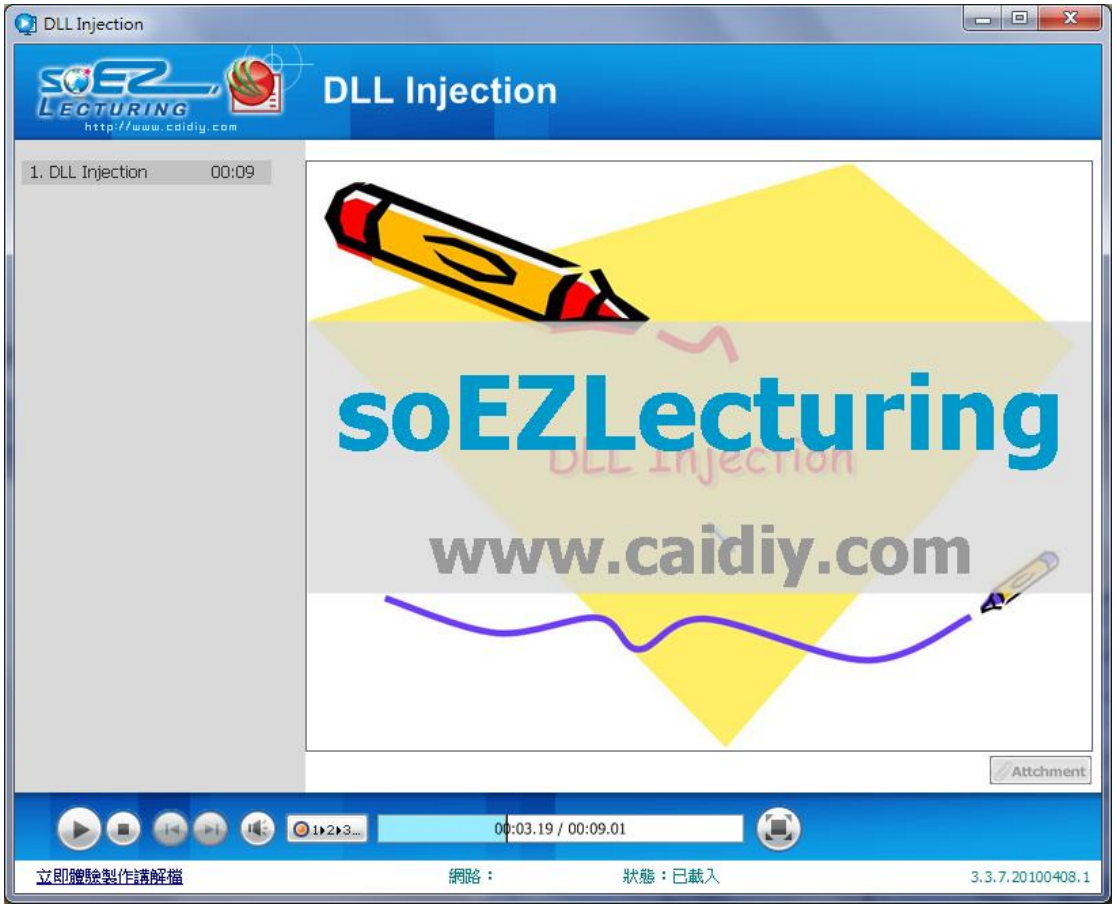


圖 35 利用講解手瀏覽器閱讀課程內容

5.2 安裝與啟動防測錄模組

完成 5.1 節的操作之後，即可產生一份受 DRM 數位版權保護的課程檔案。此章節將利用 4.1.3 節中所撰寫的防側錄模組 Install 程式，將防側錄模組 Install 至閱讀者電腦中的 SCM 中。安裝完成之後，可由工作管理員中的服務找到安裝完成的防側錄模組，如圖 36 所示。但是安裝完的防側錄模組的狀態為已停止，雖然在 Install 程式中有設定為自動啟動，但是因為尚未重新開機，所以它仍為被開啟。

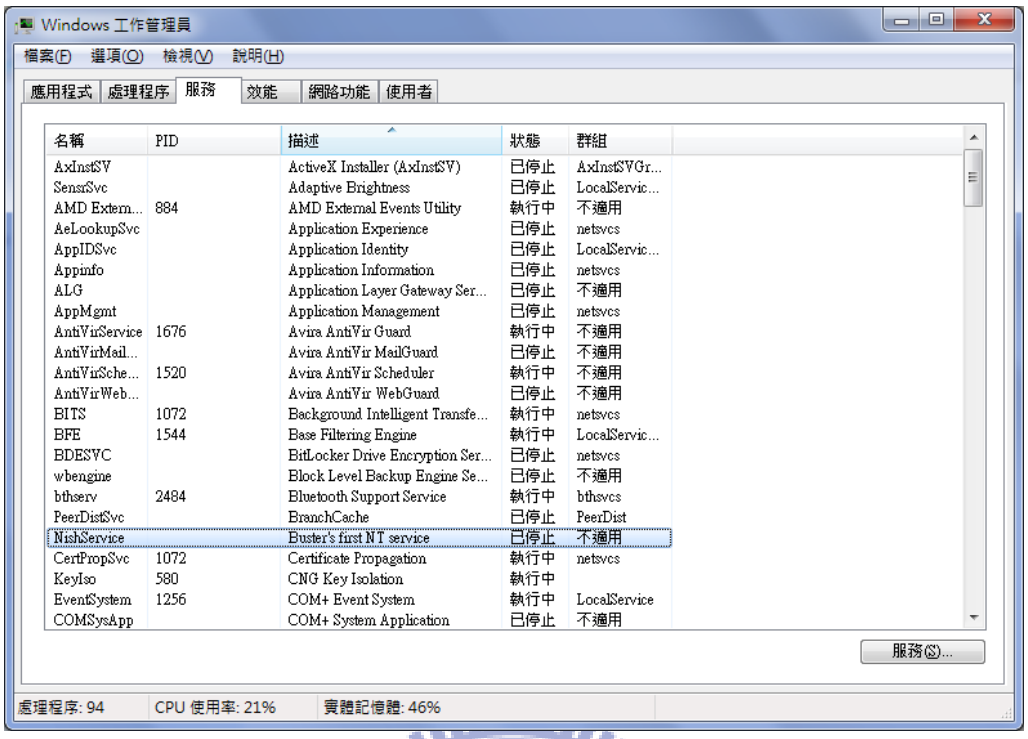


圖 36 防側錄模組已成功安裝

接下來，利用 4.1.4 節所撰寫的防側錄模組 Start 程式，來啟動防側錄模組。即可由工作管理員中的服務看到防側錄模組已啟動，其狀態已改為執行中，如圖 37 所示。

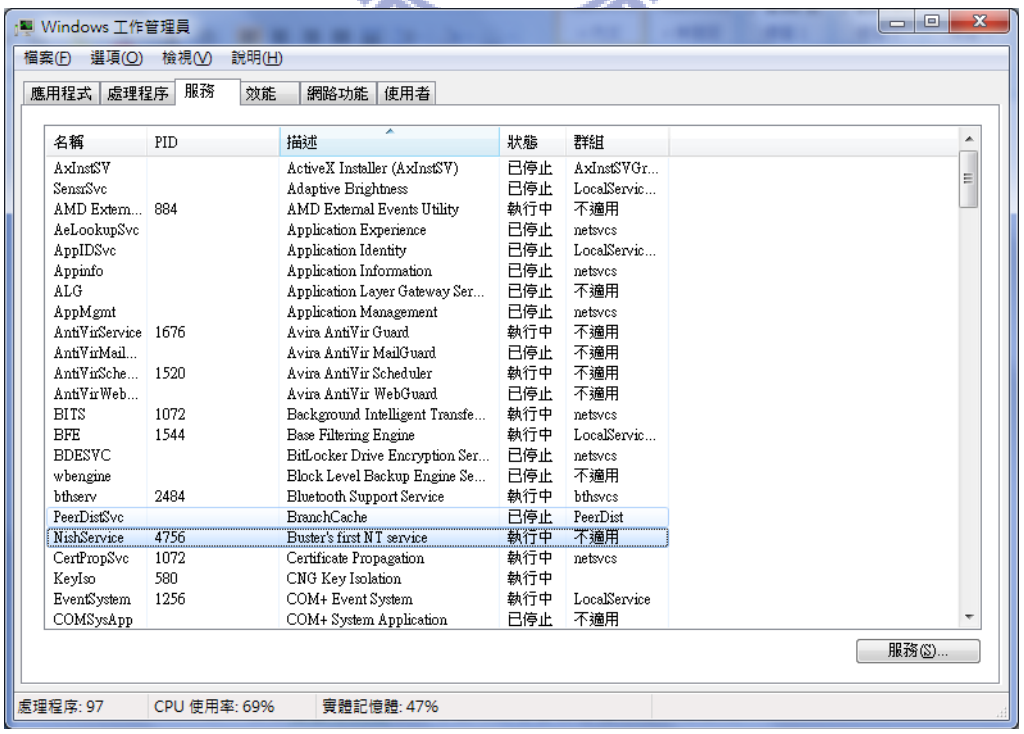


圖 37 成功啟動防側錄模組

5.3 防側錄模組的驗證

本章節中將會驗證本研究的防側錄模組的功能，例如:可偵測在 Windows 7 的 UAC 機制下的側錄程式，可偵測執行檔被竄改過之側錄程式。並且利用模擬講解手瀏覽器之程式來接收與傳送訊息給防側錄模組，以便觀看偵測之結果。

5.3.1 偵測具備管理者權限之側錄程式

在 Windows 7 的 UAC 機制之下使用者可指定側錄程式具有管理者權限來躲過以往的防側錄模組之側錄偵測。因此，此章節將驗證雙重偵測防側錄模組可偵測具有管理者權限的側錄程式。

Step 1、將側錄程式用管理者權限開啟，對側錄程式點擊滑鼠右鍵，在清單中可選擇用管理者權限開啟它，如圖 38 所示。

圖 38 使用管理者權限開啟側錄程式

Step 2、開啟模擬講解手瀏覽器之程式來接收與傳送訊息給防側錄模組。

Step 3、利用模擬講解手瀏覽器之程式接收要求開始偵測之訊息給防側錄模組。

此時防側錄模組偵測出有側錄程式正在執行，所以發送要求停止播放之訊息給講解手瀏覽器，因此模擬講解手瀏覽器之程式就接收到此訊息，如圖 39 所示。

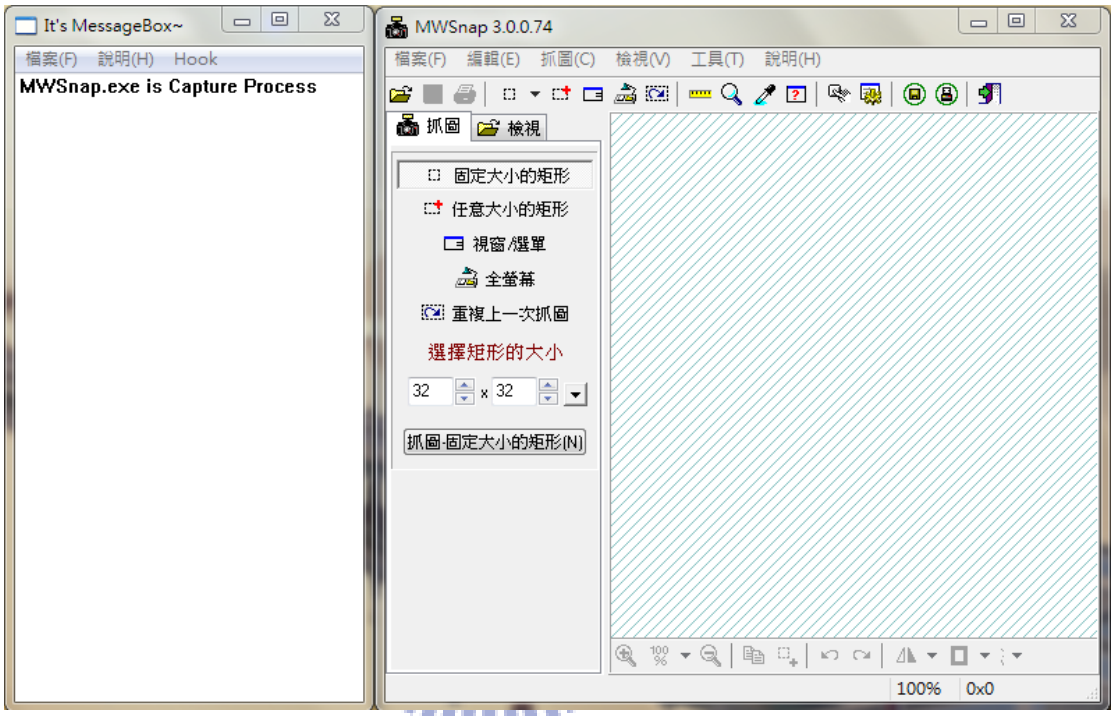


圖 39 模擬講解手瀏覽器之程式接收到有側錄程式之訊息

因此，驗證了雙重偵測防側錄模組可偵測具有管理者權限的側錄程式。

5.3.2 偵測執行檔被竄改過之側錄程式

當側錄程式被竄改其執行檔之後，所產生的特徵碼就不唯一，進而可躲過以以往的防側錄模組。因此，此章節將驗證雙重偵測防側錄模組可偵測執行檔被竄改過的側錄程式。

Step 1、利用修改 PE File 的軟體工具-PEditor，對側錄程式之執行檔進行修改，如圖 40 所示。

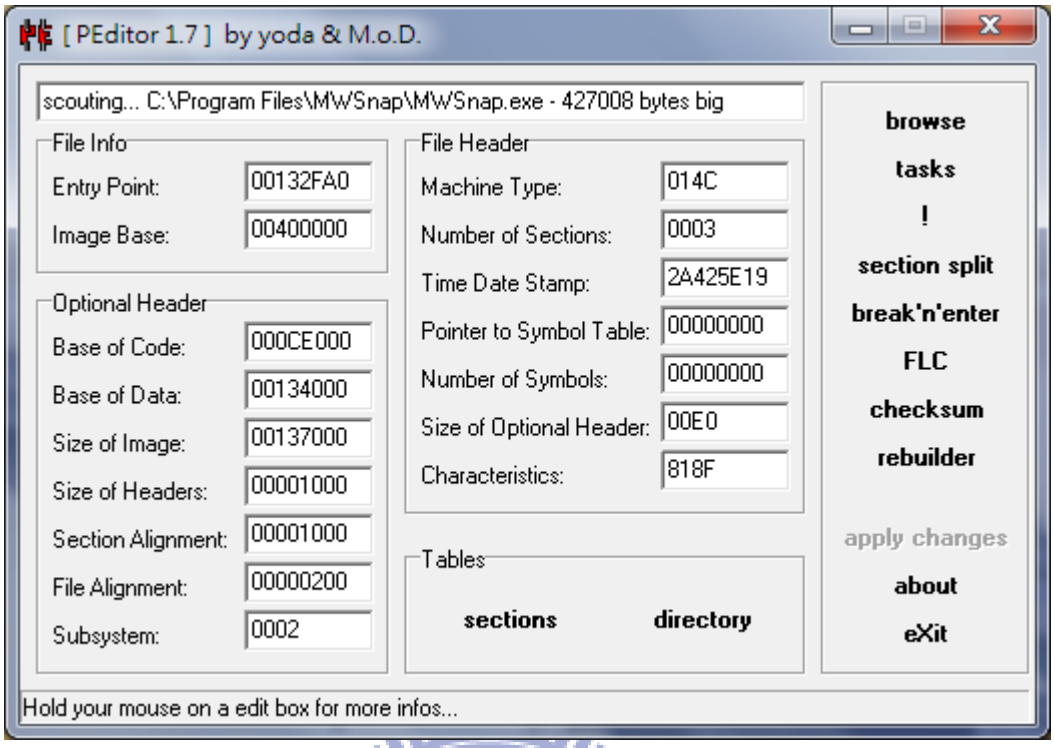


圖 40 利用 PEditor 修改側錄程式之執行檔

Step 2、開啟執行檔被竄改的側錄程式。

Step3、利用模擬講解手瀏覽器之程式接收要求開始偵測之訊息給防側錄模組。此時防側錄模組發現可疑程式，因此對該程式進行動態偵測，開始攔截該程式之 Message，如圖 41 所示。

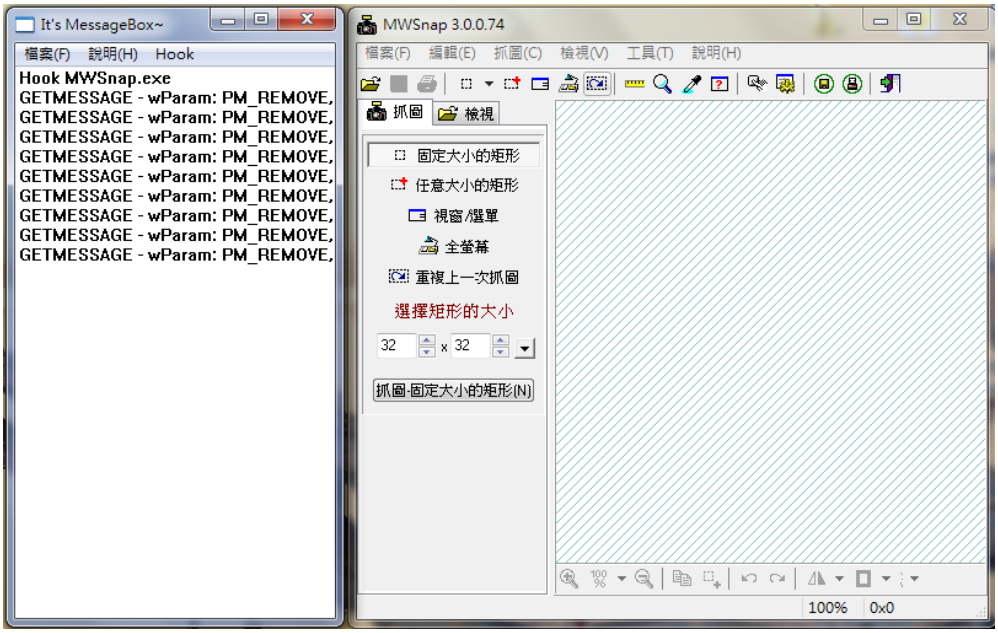


圖 41 防側錄模組利用動態偵測攔截可疑程式之 Message

Step 4、利用側錄程式對桌面進行側錄之行為。此時 Hook 函式攔截到可疑程式送出 DCI command Message，判別該程式有側錄行為，馬上通知防側錄模組並告知可疑程式之 Thread Id，如圖 42 所示。

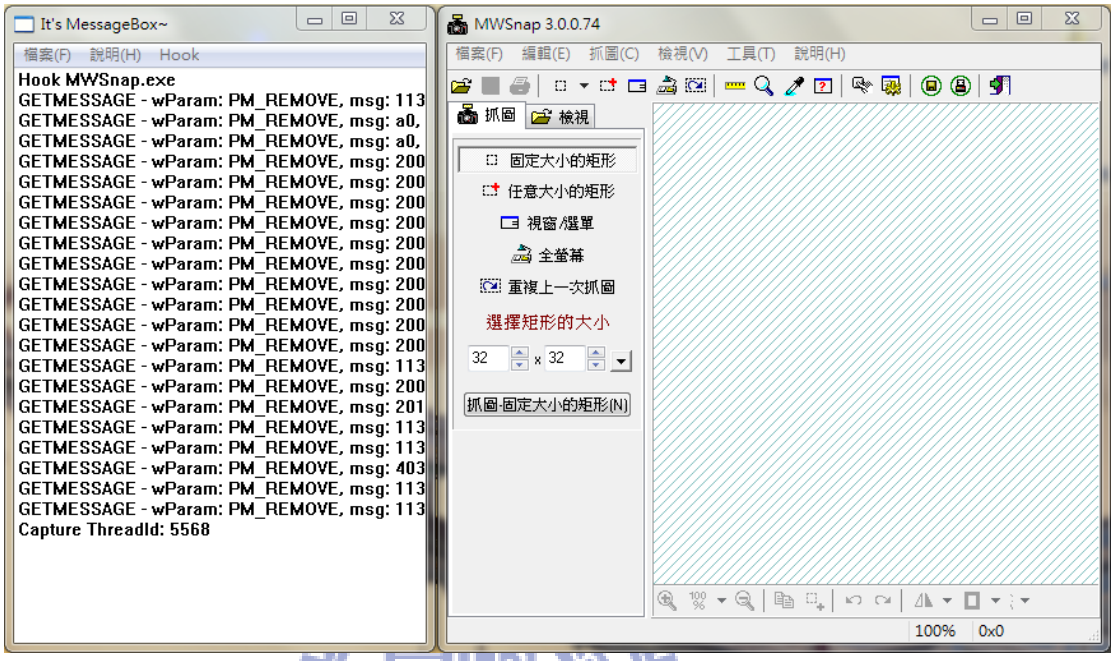


圖 42 Hook 函式判定可疑程式有側錄行為

Step 5、防側錄模組借由 Hook 函式所傳回的 Thread Id，將該可疑程式的特徵碼更新到資料庫中，並且傳送要求停止播放之訊息給講解手瀏覽器。

因此，驗證了雙重偵測防側錄模組可偵測出執行檔被竄改過的側錄程式。

六、 結論與未來展望

6.1 總結

本研究所提出的防側錄機制，目的如下:

- 將防側錄模組以 Windows Service 之型態運作，使防側錄模組可偵測其他程式之行為，解決 Windows 7 的 UAC 使側錄偵測較為困難之問題。

- 設計雙重偵測機制以補強防側錄模組的偵測能力，可偵測出執行檔被竄改之側錄程式的行為，進而阻擋側錄程式之側錄行為。

因為 Windows 7 的 UAC 機制使得可以由使用者指定側錄程式具有管理者權限，進而躲過以往的防側錄模組之偵測。主要原因是因為權限較低的防側錄程式無法取得權限較高的側錄程式之相關資訊，使得該防側錄程式無法正常運作。並且發現利用特徵碼比對技術之防側錄程式的問題，當側錄程式之執行檔被竄改過之後，該側錄程式所產生的特徵碼就不再是唯一，進而可躲過防側錄程式之偵測。

為了解決以上之問題，本研究提出結合 Windows Service 與雙重偵測機制之防測錄模組。

由於 Windows 是個多工處理的系統架構，可允許多個程式同時進行，而這些程式之資訊，均可由「Windows 工作管理員」中的處理程序來檢視。另外，Windows 系統是利用 Message 作為其控制機制，系統程式與使用者程式可以通過 Message 為傳遞訊息給其他程式，並且系統和應用兩者都可以產生消息。根據以上兩項系統特性，當本研究的防側錄模組已啟動並接收到要求開始偵測的訊息之後，便能立即列舉出所有正在執行之程式，並加以產生該執行檔之特徵碼，再與資料庫進行特徵碼比對。當發現可疑程式時，再利用 Windows Hook 技術攔截可疑程式之 Message，進而判斷該程式是否有側錄行為。

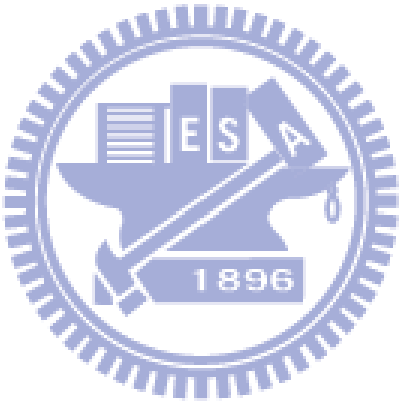
以上的作法可有效偵測其它程式之行為，並且可有效偵測是否有側錄程式正在運作。

6.2 未來展望

在此章節針對本研究所提出的防側錄機制，提出一些功能上的補強與未來發展的方向。

- 未來可評估移植到其他平台之可能性，例如：Apple OS, Android OS, …。最近幾年的智慧型手機與其他手持電腦越來越普及，數位內容課程也漸漸地應用在這些平台上，因此，可研究在這些平台上開發防側錄程式。

- 未來可開發加強特徵碼計算之演算法，因為特徵碼容易遭受到竄改，可研究一套產生特徵碼的演算法來加強特徵碼比對技術，加強特徵碼的唯一性而補強防側錄技術。



參考文獻與資料

[1] LUEVELSMEYER, PE Format, [On-line].Available:
http://webster.cs.ucr.edu/Page_TechDocs/pe.txt

[2] MSDN, Peering Inside the PE: A Tour of the Win32 Portable Executable File Format (M. Pietrek), in: Microsoft Systems Journal 3/1994, [On-line].Available:
http://msdn.microsoft.com/en-us/library/ms809762.aspx

[3] Randy Kath , The Portable Executable File Format from Top to Bottom, [On-line].Available:
http://www.csn.ul.ie/~caolan/publink/winresdump/winresdump/doc/pefile2.html

[4] Windows Hook,[On-line].Available:
http://msdn.microsoft.com/en-us/library/ms632589(v=vs.85).aspx

[5] John Robbins, Debugging Applications: Microsoft (Dv-Mps Programming) (Paperback)

[6] 史萊姆論壇, 詳談 HOOK API, [On-line].Available:
http://forum.slime.com.tw/thread167916.html

[7] 趨勢科技, 防毒入門-基本概念-認識病毒碼與掃瞄引擎, [On-line].Available:
http://www.trend.com.tw/corporate/security/virusprimer_2.htm

[8] 智勝國際科技, [On-line].Available:
http://www.caidiy.com/

[9] 網核股份有限公司, OSafe 企業內容安全方案, [On-line].Available:
http://www.iisc.com.tw/Product/Security/OsafeMirage/tabid/50/language/zh-TW/Default.aspx

53

[10] TrustView inc., TrustView DRM for Office, [On-line].Available:

http://www.trustview.com.tw/default.aspx?tab=product_office

[11] 以柔資訊, [On-line].Available:

http://www.wnjsoft.com/index.php

[12] Windows Service, [On-line].Available:

http://en.wikipedia.org/wiki/Windows_service

[13] DLL Injection, [On-line].Available:

http://en.wikipedia.org/wiki/DLL_injection

[14] VNC Mirror Driver, [On-line].Available:

http://www.tightvnc.org/driver.html

[15] Sessions, Desktops and Windows Stations, [On-line].Available:

http://www.cppblog.com/dawnbreak/articles/90278.html

[16] Advanced Windows Programming, [On-line].Available:

http://debut.cis.nctu.edu.tw/~ching/Course/AdvancedC++Course/__Page/Windows_Programming.htm#Introduce%20to%20the%20Windows%20Programming

[17] 簡單分析特徵碼修改技術, [On-line].Available:

http://big5.china-code.net/read/8/2/148484.html

[18] Session 0 Isolation, [On-line].Available:

http://iamgyg.blog.163.com/blog/static/382232572010199211892/

[19]User Account Control, [On-line].Available:

http://en.wikipedia.org/wiki/User_Account_Control