行政院國家科學委員會專題研究計畫 成果報告

行動無線網路安全與惡意程式行為分析跨國產學合作計畫（國際合作）
研究成果報告(完整版)

計 畫 類 別 ：個別型
計 畫 編 號 ：NSC 98-2218-E-009-020-
執 行 期 間 ：98 年 10 月 01 日至 99 年 10 月 31 日
執 行 單 位 ：國立交通大學資訊工程學系（所）

計 畫 主 持 人：曾文貴
共 同 主 持 人：謝續平、黃育綸
計畫參與人員：碩士級-專任助理人員：張智凱
碩士級-專任助理人員：陳柏愷
碩士班研究生-兼任助理人員：王嘉偉
碩士班研究生-兼任助理人員：江孟寰
碩士班研究生-兼任助理人員：鄭昀旻
碩士班研究生-兼任助理人員：鐘凱任
碩士班研究生-兼任助理人員：盧彥銘
碩士班研究生-兼任助理人員：顏豪緯
碩士班研究生-兼任助理人員：彭博群
碩士班研究生-兼任助理人員：黃錦銘
碩士班研究生-兼任助理人員：葉書宏
碩士班研究生-兼任助理人員：鄭偉強
碩士班研究生-兼任助理人員：許鴻生
碩士班研究生-兼任助理人員：黃奕奇
碩士班研究生-兼任助理人員：賴鈺婷
碩士班研究生-兼任助理人員：李勇叡
碩士班研究生-兼任助理人員：黃晉澤
碩士班研究生-兼任助理人員：陳玟媗
碩士班研究生-兼任助理人員：吳思穎
大專生-兼任助理人員：那西格
博士班研究生-兼任助理人員：沈宣佐
博士班研究生-兼任助理人員：陳毅睿
博士班研究生-兼任助理人員：蔡欣宜

博士班研究生-兼任助理人員：陳柏廷

報 告 附 件 ：國外研究心得報告
出席國際會議研究心得報告及發表論文
國際合作計畫研究心得報告

處 理 方 式 ：本計畫可公開查詢

中　華　民　國　100 年 01 月 26 日

行政院國家科學委員會補助專題研究計畫 **■成果報告**
□期中進度報告

行動無線網路安全與惡意程式行為分析跨國產學合作計畫

（國際合作）

計畫類別：■個別型計畫　□整合型計畫
計畫編號：NSC 98－2218－E－009－020－
執行期間： 98 年 10 月 01 日 至 99 年 10 月 31 日

執行機構及系所：國立交通大學資訊工程學系（所）

計畫主持人：曾文貴
共同主持人：謝續平、黃育綸

成果報告類型(依經費核定清單規定繳交)：□精簡報告　■完整報告

本計畫除繳交成果報告外，另須繳交以下出國心得報告：
□赴國外出差或研習心得報告
□赴大陸地區出差或研習心得報告
□出席國際學術會議心得報告
■國際合作研究計畫國外研究報告

處理方式：除列管計畫及下列情形者外，得立即公開查詢
□涉及專利或其他智慧財產權，□一年□二年後可公開查詢

中　華　民　國　100 年　1 月　5 日

摘 要

隨著智慧型手機裝置（Smart Phone）與行動小筆電（Netbook）的發展，以及無線網路與行動上網的普及，行動資訊服務將是未來資訊技術的發展趨勢。有鑑於此，本計畫提出一套完善的行動平台安全檢測方案，並選派優秀學生赴UC Berkeley交流研究達四人次，在無線異質網路模擬、惡意程式分析、程式漏洞檢測、與行動平台安全管理機制等研究主題上均提出論文及雛型系統等具體研究成果。相關研究成果亦已與中華電信及友訊科技等業界知名廠商簽訂三項產學合作計畫。

本計畫已完成無線異質網路模擬平台的開發，可支援涵蓋16節點的網路實驗。使用者可在此無線異質網路模擬平台上，進行異質性無線網路系統的安全測試，以模擬可能的異質網路拓樸與可能存在的攻擊手法。本系統亦與UC Berkeley合作，並已與其開發之有線網路測試平台DETER完成相容性整合。

惡意程式除了較常被隱藏在執行檔內之外，一般常見的文書檔案如MS Word、MS PowerPoint、PDF等亦有可能成為散播的媒介。針對此研究議題，本計畫發展出一套惡意檔案文件分析系統。本系統利用虛擬CPU及記憶體動態分析技術，可偵測出目標文件是否有夾帶惡意程式或不當獲取系統資訊的意圖。同時本系統亦可過濾各類不正常字串，並找出該惡意程式利用何種手法來達到攻擊目的。其論文亦榮獲CISC2010之Best Student Paper Award肯定。

針對程式漏洞檢測之研究，本計畫對知名自由軟體 Catchconv 進行改良，已開發完成的程式碼迴圈處理機制能偵測重複且可省略的動作，透過將計算能量集中在不重複的分析上增加其執行效率。

此外，於行動平台的安全管理機制之研究，本計畫已完成一個解決方案，其中包含一套入侵偵測系統以及一套安全雲端儲存系統。於入侵偵測系統，本計畫成功將Snort移植到Android行動平台，此為Android上的第一套入侵偵測系統，並已針對執行速度、耗電量、與封包抓取率等方面進行最佳化，以確保Snort於Android上執行仍可保有高度的可用性。本計畫亦設計出一套安全雲端儲存系統，將使用者資料加密後利用Decentralized Erasure Code的方式分散在雲端儲存上，以達到高度安全性及可靠性，雛型系統已完成，並已有論文產出。

除以上具體研發成果之外，本計畫亦遵循Light-Weight CMMI標準提出專案計畫書，同時配合由中研院所推動之「自由軟體產業推動計畫」，於OpenFoundry網站建立三個專案並開放兩項雛型系統的原始碼。可望能將安全技術研發與自由軟體社群連結，並引入自由軟體社群的研發能量以吸引產業界投入的意願、厚植國內資訊安全技術的研發潛力。

關鍵詞

雲端安全、行動資安、異質無線網路、惡意程式分析、程式漏洞檢測

Abstract

Wireless and mobile networks are widely used in modern life. With the development of smart phones and netbooks, providing mobile information services has become a trend of internet technology. This situation, however, could cause security problems.

This project proposes a security inspecting solution for these problems. In this project, four outstanding students were sent to UC Berkeley for months for joint research projects focusing on heterogeneous networks, malware analysis, program defect detection, and the security management of mobile platforms. Based on the research findings, this project also developed prototype systems with related paper published. The project also contributed to various cooperation projects with some well-known companies, such as Chunghwa Telecom and D-Link.

This project completed the development of a heterogeneous network testbed, which can support the experiments with up to 16 nodes. Users can simulate heterogeneous network topologies with the possible network attacks. This testbed can also be integrated with DETER, which is developed by UC Berkeley.

Malicious program codes can not only be hidden in executable files but document files such as Microsoft Word, PowerPoint, or PDF files. For this research area, this project developed an analyzer for the malicious document files. Using virtual CPU and dynamic memory analyzing technology, this system can detect if the target file contains malicious codes or tries to obtain sensitive system information. Meanwhile, this system can also detect various abnormal strings to determine the technics used by the malwares or malicious programs.

For the research on program defect detection, this project improved Catchconv, a famous open source tool, which is widely used for gray-box testing. The improvement is focused on a loop avoiding algorithm and its prototype system. This system can detect and skip unnecessary operations result from the loops in the program. This does improve the efficiency of Catchconv.

Moreover, this project includes an IDS (Intrusion Detection System) on Android and a secure cloud storage system to improve the security management for mobile platforms such as smart phones. The IDS is a ported version of snort based on Linux system and is the first IDS system for Android. It is also optimized for the efficiency, power consuming, and packet capturing rate. We believe that it can run on Android applicably. For the secure cloud storage system, it encrypts and distributes the data to the cloud storage through Decentralized Erasure Code to achieve the goal of security and robustness. The prototype had also been completed with related paper published.

Besides, this project also follows Light-Weight CMMI and had created four projects on OpenFoundry. Two of the prototype systems also opened their source code. This could connect the security researches and the community of open source software. Hopefully this could attract the interests of the industries and can strengthen domestic potential for information security technology.

Keywords

Cloud Security、Mobile Security、Heterogeneous Network、Malware Analysis、Program Defect Detection

目 錄

壹、 計畫內容及目的 1
一、 前言及背景說明..... 1
二、 文獻探討與相關研究..... 5
三、 計畫目的..... 12
四、 研究方法..... 15
貳、 計畫項目完成度 20
參、 計畫成果 26
一、 國外交流研究..... 26
二、 產學合作計畫..... 32
三、 學術貢獻..... 34
四、 系統建置..... 35
五、 技術方案優越性..... 84
肆、 結論與展望 88
伍、 參考文獻 90

壹、 計畫內容及目的

一、 前言及背景說明

隨著智慧型手機裝置（Smart Phone）與行動小筆電（Netbook）的發展，以及無線網路與 3G/3.5G 行動上網的普及，行動資訊服務將是未來資訊服務發展的趨勢。如何提供使用者安全、可信賴的服務環境，將是重要的關鍵議題。但現今國內資安產業之關鍵技術仍仰賴國外進口，使得資安產業離技術自主性之目標仍有差距。如能增加國際合作與人員交流，加速引進關鍵技術，除有助於國內資安產業的發展，更能提升台灣於國際資安產學界之地位。

因此，本計畫之目的在於提供一套完善的行動平台安全檢測方案，包含無線異質網路模擬平台、惡意檔案文件分析、程式漏洞檢測以及行動平台的安全管理機制；大至整體的網路環境，小至行動裝置的安全保護，期許能夠提供使用者完善、安全的行動資訊服務環境。以下各節將分別針對各研究子項之相關背景進行介紹：

- **無線異質網路模擬平台**

近年來，隨著無線網路服務與應用之迅速發展，民眾的生活已與網際網路產生密不可分之關係。在此國際趨勢下，各國莫不積極規劃具前瞻性的資通訊政策，期望以完善的無線網路基礎建設與應用服務，帶動資訊產業成長，進而提升國家競爭力。我國政府於 2004 年研擬「行動台灣計畫」 (M-Taiwan)，規劃台灣行動生活產業科技發展策略，期望能成為全球領先的行動生活國家。為配合 M-Taiwan 計畫，國內法人單位、研究學者紛紛切入異質無線網路連通、無縫架構與安全技術之研究領域，以整合國內使用中的各類無線網路環境，諸如 GPRS (General Packet Radio Service，整合分封無線服務)、 3G (the Third Generation，第三代行動通訊)、Wi-Fi (Wireless Fidelity，無線相容認證)等，以及國際間正積極推動之前瞻性技術，包含了 WiMAX (Worldwide Interoperability for Microwave Access，全球互通微波存取)、4G (the Fourth Generation，第四代行動通訊)等技術。

異質無線網路測試必需涵蓋多種不同的網路型態和特性，研究人員除了考量各基本網路型態之安全機制外，尚需解決許多存在於不同網路銜接界面之間的安全議題，例如：無線網路裝置於異質網路交替 (Handoff) 時所需之安全驗證機制等等。因此，如何建立一套適用於異質無線網路測試的實驗環境成為現今學者們面臨的一大挑戰。透過建置實體的異質無線網路測試環境，可以迅速且準確的驗證新提出之安全機制；但由於新興的無線網路設備售價昂貴，建造整合多項無線網路的實體測試環境所費不貲。另一方面，藉由軟體模擬來評估安全機制，可解決缺乏實體測試環境之困難；然而，軟體模擬往往只能擷取部分系統屬性進行分析，無法模擬因硬體設計所造成的效能瓶頸。

因此，本計畫之研發目標乃在於建立一套適用於無線異質網路之模擬平台。該平台使用實體機器模擬異質無線網路實驗中的網路設備，使該平台上進行的無線網路模擬實驗可兼具硬體設備之效能考量，提高實驗結果之真實性。本計畫的研發團隊亦發展軟體模擬之技術仿真無線網路底層訊號之傳遞，期望提供平台使用者一個可控制、可重覆之實驗環境。同時，為了使該平台具備易移植性、高彈性以及高擴充性等優點，本計畫開放使用者於平台上實做新型的無線網路技術並進行測試，以補足無線網路相關技術開發環境之不足。我們相信，基於此異質無線網路測試平台，產學界各單位可以進行符合真實環境的產品和安全機制之測試，縮短前瞻網路技術開發與測試時程，加快新興網路的布建速度。

- **行動平台的安全機制（入侵偵測系統、安全雲端儲存系統）**

行動資訊服務是未來的資訊服務發展的趨勢，如何提供使用者方便與安全的系統環境是這些服務成功的關鍵。智慧型手機和行動小筆電（netbook）將是使用者主要的行動平台，透過高速的電信與資訊網路系統，使用者將可以在任何時間任何地點取得資訊服務業者提供的服務。目前行動平台的作業系統方面，除了微軟的Windows Mobile及 XP外，Google也積極的搶進這塊新興的領域，例如手機上的的Android系統及Netbook上的Chrome系統。而智慧型手機系統近年來以Android市佔率成長最快。Android系統的手機秉持著「使用和擴展」的目標，公開內部的應用程序，並提供一個開放平台加上一些簡單的規定，讓所有人都能成為應用程式的開發員，因而激發手機上無限可能的應用。

而由於使用無線網路、Android系統的開放性以及行動上網逐漸風行等關係，行動平台特別容易遭受攻擊並引發了大量的安全問題，因此我們需要各類針對行動平台的安全機制，例如，弱點掃描系統、入侵偵測與防禦系統、身份認證系統、防毒系統等，以因應無所不在的攻擊。另一方面，像是智慧型手機和小筆電等這類的行動裝置具有體積小與移動性高的特點，使得行動裝置較容易面臨遺失或失竊的情形，一旦裝置遺失則裝置內的資料將無法復原。例如手機，如果沒有備份手機上的資料，手機一旦遺失，其中儲存的通訊錄，簡訊甚至是照片都將遺失。若是手機被有心人士取得，那麼手機上的個人資訊將暴露殆盡，這對個人隱私權造成相當大的傷害。因此行動平台也是雲端服務的對象之一，雲端儲存服務可以完備整個行動平台的功能性。透過雲端儲存服務，使用者可以將資料儲存或備份在雲端，爾後只要透過網路就可以存取這些資料。而如何保障使用者儲存在雲端的資料的安全，並進一步提供多樣性的功能，也是我們將探討的議題。個人隱私問題關係著使用者的公民權力，如何保障使用者的個人資訊不被竊取與惡意使用是值得探討的議題。

- **惡意檔案文件分析**

隨著近年來網路的快速發展，藉著快速的資訊流通，造成了零時差攻擊 (zero-day attack) 這種新興的網路攻擊手法。零時差攻擊利用了軟體漏洞資訊的洩漏，在極短的時間內利用公佈的漏洞進行攻擊。若業者未能及時發布修正檔來修補漏洞，則會造成

眾多的使用者受害，或者是龐大的財產損失。即使業者推出了安全性的更新檔，也因為無法同步世界各地的更新狀況，無法確保較慢更新的使用者的安全性。而這些攻擊手法，大多使用夾帶於檔案中的惡意程式，騙取使用者執行該檔，進一步達到攻擊的目的。如何判斷檔案的威脅性，已經是資安領域中熱門的問題。

可造成破壞行為不一定要透過可執行檔，由於軟體開發人員的疏失，連同一般常見的文書檔案，如Microsoft office Word、PowerPoint、PDF等，都可能成為散播惡意程式的媒介。檔案本身雖無破壞能力，但利用相關軟體的漏洞，可以執行攻擊者事先嵌入的指令，控制該軟體的行為。甚至，執行遠端的程式碼、或下載遠端惡意程式至本地端執行來達到攻擊的目的。以近期為例：Microsoft Security Advisory (969136) v2.0修正檔即為修補PowerPoint允許遠端程式碼執行的漏洞，但早在發布的前一個月，各大資安論壇卻已廣泛討論此漏洞，從而得知此類攻擊早已是駭客們所重視的攻擊手法。

我們希望能夠開發出一套偵測檔案安全的系統，對於任何可能隱藏惡意程式的檔案，加以分析並且產生報告。對於一般的非執行檔，能夠偵測出是否有夾帶惡意程式，或者檢測其中是否有意圖獲取更多關於系統方面的資訊，過濾不正常的字串，並且找出該惡意程式利用何種技術來達到攻擊目的。最後，利用評比顯示出該檔案的威脅度，以提醒使用者是否信任該執行檔。

- **程式漏洞檢測**

在程式的開發過程中，檢查、驗證程式本身的正確性，是一個相當重要的步驟；一般來說，約30%左右的開發成本花費在測試和除錯階段。比較輕微的程式錯誤，可能只是產生錯誤、非預期的執行結果而已；然而，重大的程式漏洞，卻可能會危害到整個系統的安全。例如：程式漏洞可能會造成系統當機，讓系統不能繼續正常運作；惡意攻擊者也可能利用此程式漏洞，去嘗試滲透、攻擊系統，進而埋入惡意程式或是搶奪系統的控制權，竊取使用者的個人隱私資料，造成使用者的損失。程式漏洞的傷害根據它所影響的範疇及程度而定，有些可能無足輕重，有些卻會造成莫大的傷害。對於應用在資訊安全領域的軟體程式，例如大數、密碼方面的程式庫以及資訊安全協定的軟體實作，其正確性及安全性更是重要，若是含有漏洞，將可被利用來攻擊。例如：在CRYPTO 2008年的研討會上，Eli Biham、Yaniv Carmeli和Adi Shamir三人發表了論文“Bug Attacks”，提出利用系統硬體的設計錯誤或是密碼軟體實作上的漏洞，去推測、擷取使用者秘密金鑰的攻擊方法。這讓我們不得不去重視該如何驗證程式本身的正確性，以及如何更有效地去驗證程式的技術。

檢查程式的方法有許多種，根據我們對於程式內部運作的了解程度，可以分為以下三類：

(1) 有程式碼，對於程式內部運作完全了解，又稱作 White-Box Testing。除了可以透過程式碼分析進行靜態檢測，來尋找可能的程式漏洞外，也可產生大量的測試資料來對程式執行檔進行動態檢測，來發現潛在的程式漏洞。然而，對於大部分的程式要取得其程式碼並不容易，尤其是用來營利的商業軟體；另外，對於較為龐大複雜的程式，其大量的程式碼更是造成了檢查上的不方便及麻煩。如果利用軟體來分析程式碼，會產生相當多的 False Warning，需

要交由人工來檢驗判定是否真的是程式漏洞；我們需要許多的人力去理解、追蹤程式碼的內容，並從所有可能的執行狀態中找出會有問題的部分。這些因素使得 White-Box Testing 在實務上並不實際。例如：2004 年 Windows 2000/NT 4 的程式原始碼被洩漏在網路上，當時大家紛紛質疑這些外洩出來的程式碼是否會暴露出 Windows 系統的安全漏洞，或是被駭客利用來研發攻擊程式，造成系統安全上的傷害。但事實證明，這些顧慮只是多餘的，外洩出來的 Windows 系統程式碼太過大量，造成實際分析上的負擔，即使這些程式碼提供了所有關於 Windows 系統的實作資訊，也因為相關資訊太多，而無法辨認出真正有用的部分。

(2) 沒有程式碼也沒有程式執行時期資訊，對於程式內部運作完全不了解，只知道程式的輸入資料及輸出結果，又稱作 Black-Box Testing。對於這種沒有程式原始碼，只有程式執行檔的狀況，通常是產生一些特定的測試資料位給程式執行，用來檢測程式的輸出結果是否正確符合預期。要產生有用的測試資料需要程式實作的相關資訊以及豐富的測試經驗，這通常只在程式開發團隊內部存在。當無法具備前述條件時，Fuzz Testing 是主要的測試方法，藉由隨機產生大量的測試資料，來對程式執行檔進行動態分析，觀察比對程式的執行結果，來發現可能的程式漏洞。然而，由於缺乏程式執行時期的資訊，Fuzz Testing 所產生的隨機測試資料，很難涵蓋到所有的程式區段，通常只能測試到程式的一小部分而已。在大量的測試資料中，僅有少部分能夠測試到比較特殊的程式區塊；大部分都是重複地執行測試相同的程式區塊，這使得 Black-Box Fuzz Testing 在實際上的功效並不如預期。例如：有一程式漏洞僅在整數（32 bits）變數 X=10000 時發生，此時 Fuzz Testing 隨機產生的測試資料能夠涵蓋到此漏洞之機率僅為 1/232，幾乎無法偵測到此漏洞。

(3) 沒有程式碼但有程式執行時期資訊，能夠知道程式執行時的變數內容，以及程式執行時的條件分支執行流程，又稱作 Gray-Box Testing。雖然沒有程式原始碼，但是利用類似 Virtual Machine 或是 Run-time Debugger 的技術，監控目標程式的執行過程，取得執行時期的變數資訊和條件分支資訊，提供檢測時所需相當有用的幫助。Gray-Box Testing 與 Black-Box Testing 相當類似，最主要的差別在於當知道程式執行時期的資訊時，Fuzz Testing 能夠根據程式執行時的變數內容、條件分支執行流程去產生有意義的測試資料，而非單純地隨機產生測試輸入；這種環境行為模式下的 Fuzz Testing 又被稱作是 Concolic Testing。例如：有一程式區塊在整數（32 bits）變數 X=10000 時才會被執行到，Concolic Testing 一開始雖是使用隨機的測試資料，但是在程式執行、動態檢測的過程中，便能利用執行時期的變數資訊和條件分支資訊，得知有一程式區塊僅在 X=10000 時才會被觸發執行，接下來產生能夠使 X=10000 的測試資料，讓 Concolic Testing 再下一輪的檢測中能夠涵蓋到此程式區塊，檢測此一區塊的安全性。Concolic Testing 在 Gray-Box Testing 的環境下相當有效，每個測試資料將帶領我們去檢查尚未檢查過的程式區塊，是目前檢查程式漏洞的主要方法。

二、 文獻探討與相關研究

本計畫所涵蓋的範圍包含無線異質網路模擬平台、惡意檔案文件分析、程式漏洞檢測以及行動平台的安全管理機制等。在此我們將目前各子項之文獻探討以及相關研究列出並說明，藉此可了解目前其他研究者的研究方向及內容。以下將就本計畫所涵蓋範圍內的各項相關研究提出分析與說明：

- **無線異質網路模擬平台**

網路服務的興起增加了開發研究人員對於網路實驗平台之需求。許多專家學者針對網路實驗的重置性、真實性和擴充性等，設計數套網路測試平台。經過研究，以下列出幾套現有之知名網路測試平台：

- **Emulab**

Emulab [1] 是美國猶他大學為了分散式系統和網路的研究所設計的有線網路仿真 (emulated) 平台。使用者以 NS (Network Simulator) 語言描述實驗網路所需之拓樸，並存放於 NS 設定檔中。Emulab 再依據 NS 設定檔中的資訊，配置實驗用的實體節點，並在節點上載入指定的可執行映像檔。Emulab 透過虛擬區網 (Virtual Local Area Network，VLAN) 來建立實驗網路拓樸與區隔該平台上所執行的多組實驗。藉由 VLAN 的區隔，Emulab 可以實現實驗節點之間的溝通性（connectivity），使位於同一虛擬區網下的實驗節點可以彼此溝通；VLAN 技術也同時確保實驗之間的隔離性，避免不同實驗彼此互相干擾。然而，Emulab 開發之初著重於實驗的隔離性、可重複性與擴充性的設計，最初的設計並未考慮到通訊網路的安全與保護。因此，美國加州大學柏克萊分校繼而研發出基於 Emulab 的 DETER 有線網路攻擊防禦研究平台。

- **DETER**

DETER [2, 3] 是由美國加州大學柏克萊分校針對有線網路開發的攻擊防禦研究平台，它以 Emulab 為基礎，提供了具可重複性的網路安全實驗環境。除了防禦機制外，DETER 亦提供了數組惡意程式，可供使用者測試其新安全機制之強健度 (robustness)。為建造適合網路安全方法、協定、機制與產品的測試環境，DETER 採用了以下安全機制以確保實驗環境的隔離性和安全性：(1)採用網際網路安全通訊協定 (Internet Protocol Security，簡稱 IPSec) 的通道模式 (tunnel mode) 連接實驗節點和控制交換器；(2)以防火牆隔離外部和內部實驗網路。防火牆的設計除了維持內部實驗網路的單純性，亦可避免在實驗中的惡意程式失去控制時危害公用網路。

- **ORBIT Radio Grid Testbed**

ORBIT [4] 是以 802.11 實體節點為基礎，針對 3G 和 802.11 網路所設計、具二層架構的無線測試平台。ORBIT 由密集的 802.11 節點組成網格，網格中的節點可以動態地連結成所指定的無線網路拓樸。ORBIT 使用者需依照定義的格式完成實驗腳本格式，以指定網路拓樸、實驗設定和過濾機制等等；透過實驗腳

本，使用者可以遠端控制和觀察實驗之結果。然而，ORBIT 同時僅能支援一個實驗，若是使用者的實驗僅需使用 10 個節點，則 ORBIT 網格中其他剩餘的節點，在該實驗尚未完成之前將無法提供測試服務。

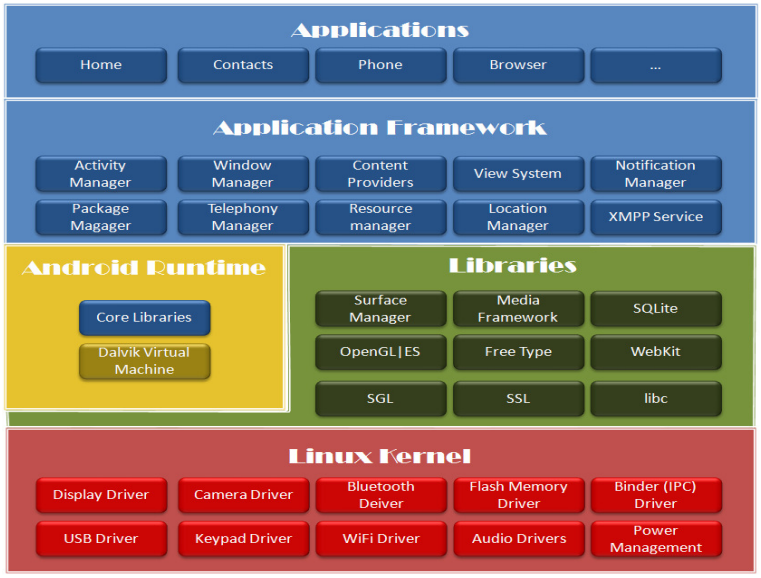
- **MobiNet**

MobiNet [5] 是一套專為無線隨建即連網路 (wireless ac-hoc network) 所設計的仿真模擬平台，其架構由二大元件組成：邊際節點 (edge node) 和核心節點 (core node)。邊際節點負責處理應用程式以及模擬各種裝置，包含筆記型電腦和 PDA (Personal Digital Assistant，個人數位助理)；核心節點則是負責架構網路拓樸、管理訊號傳輸，以及仿真各種路由協定和媒介存取控制層 (Media Access Control layer，MAC layer) 協定。透過虛擬邊際節點，一個邊際節點可以模擬數十至數百個無線裝置。此虛擬化功能，可以提升 MobiNet 中無線網路實驗的規模；然而，藉由軟體仿真無線裝置的實驗方法只能模擬部分系統之屬性，導致實驗之成果缺乏真實性。

- **行動平台的安全機制**

Android 是一個 Open Handset Alliance (OHA)所開發並由 Google 在 2007 年所推出的一個應用在智慧型手機 以及筆記型電腦上之作業系統平台。在 Android 的發展中，Google 除了提供一個 Open Source 的環境讓廠商可以節省平台授權費外，開發者也能在開發應用程式上獲取更多的自由，且開發的應用程式可以在不同平台上通用，不必再花費時間及精力將應用程式作跨平台的改寫。

Android 基本上就是一種嵌入式 Linux 系統再加上一些重要的手機應用開發軟體，其系統架構總共包含四個層次。最底層是 LINUX 系統核心，採用 LINUX Kernel 2.6 版，負責硬體的驅動程式、網路、電源系統、安全以及記憶體管理等等。第二層是 Library 的部分，由大多數開放原始碼的 Library 所構成， 例如 C 的 Library、OPENSSL、SQLite 以及 Webkit 負責 Android 網頁流覽器的運作，另外還有 OPENGL 和 SGL 圖形與多媒體 Library 負責支援各種影音和圖形檔的播放。緊接著是和第二層 Library 並行的 Android Runtime，提供 Android 特有的 JAVA 核心 Library 以及可轉換 JAVA bytecode 成 .dex 檔案格式的 Dalvik 虛擬機器。第三層則是應用軟體架構，為所有 Android 核心應用程式 Framework API 的總集合，可讓程式開發者方便取用這些常用的應用程式設計架構，以便增加應用軟體發展速度。在最上層的即是 JAVA 應用程式，包含檔案管理、通訊錄、電話撥號程式等。

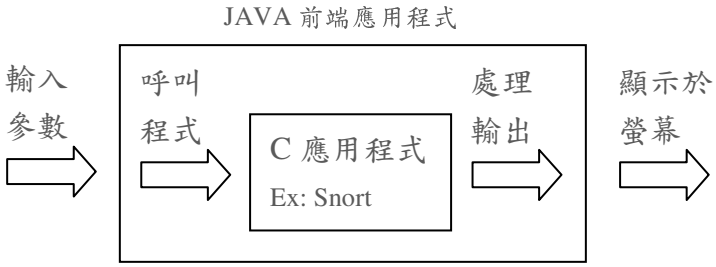


圖一：Android 架構圖

目前限定所有應用程式都必須以 JAVA 語言來撰寫，再以 Dalvik Virtual Machine 來轉換成 DX bytecode，因此程式開發和 JAVA ME 類似，應用程式的介面部分則是由 XML 程式來設計。簡單來說，即是系統由 LINUX 來執行，應用程式則透過 JAVA 語言來開發以及執行。要在 Android 上執行以 C 語言所構成的程式，有兩種方法：

(1) 利用 Android NDK，在 JAVA 應用程式中去呼叫以 C 語言所構成的 Library 以執行我們要的功能。

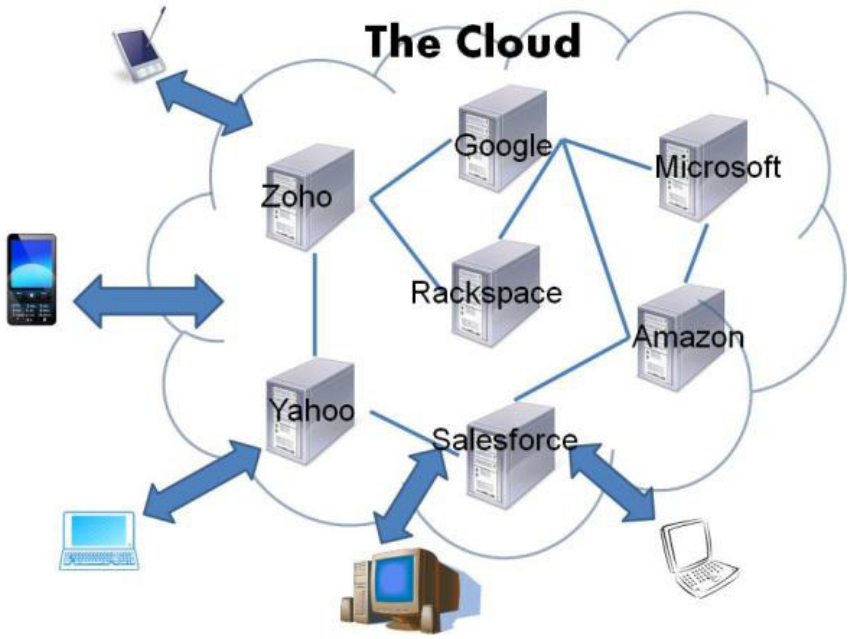
(2) 將 C 語言寫成的程式直接編譯成執行檔，然後再從 JAVA 應用程式中，去直接執行該執行檔(如圖二所示)。



圖二：JAVA 應用程式端呼叫 C 應用程式示意圖

由於 Android 等行動平台的計算能力以及儲存空間通常是受限的，故利用雲端運算技術取得額外的儲存空間以及計算能量是一個趨勢。雲端計算近年來成為大家討論的話題，而且逐漸變成下一代人們使用電腦的架構，雲端計算基本上含有一個以上大型的數據中心，裡面包含了大規模的軟體以及大量的 database，用戶們只需要一個可以連到網路的可攜式硬體，即可以透過網路享有高品質的服務(如圖三所示)，而這種獨特模式打破傳統的儲存系統均存放在自己的硬碟裡，隨著網路速度的增加以及現代人需求愈來愈多樣化的情況下，儲存技術廠商紛紛推出雲端儲存的想法，加上現在的智慧型行動裝置愈來愈普及的狀況下，許多行動裝置業者紛紛加入雲端服務的功能，

讓使用者能隨意地欣賞到最熱門的音樂、視訊、數位課程、新聞、遊戲、電子書等服務，不再像是以往的儲存系統只能在限定的機器上才能享受到應有的服務。



圖三： 雲端計算架構圖

雲端這個概念在 2007 年 10 月 IBM 與 Google 在美國校園裡展開，之所以會推廣雲端計畫，最主要的是希望能降低分散式運算在學術研究計畫裡所花費的巨大資本，所以 IBM 及 Google 提供了相關的軟硬體支援，讓每個學生都可以透過網路進行大規模運算為基礎的學術研究。此外 2008 年 1 月 Google 也宣布在台灣的雲端計畫，並且跟台灣的台大、交大等學校合作，把這個先進的雲端概念迅速的吹進了校園裡頭。

但是雲端的概念充滿了許多問題，包括安全、管理以及效率等方面逐漸浮出檯面，尤其最近幾年環保意識高漲，許多電器都需加上節能標章，所以在雲端的概念裡不斷加入了節能的方法，讓每台機器都能充分的運用到。不過在節省能源的同時也突顯了一些管理上以及安全上的問題，例如當一台沒有用到機器如何關掉其電源，而又或者忽然間大量服務湧現時，如何讓電源迅速開啟，陸陸續續許多人在管理這方面的研究，此外管理方面還會出現安全上的問題，例如當雲端計算的使用者逐漸減少，為了要節省能源將某些沒用到的電腦逐一關閉，然而在關閉的電腦裡或許還有一些使用者正在運算，那如何讓正在運算的電腦移到其另一台電腦裡，或許是管理上的問題，但是在移後不會讓目前的使用者不被移到這台的使用者所存取或是相反過來，都是我們不願意看到的結果，所以近年來雲端發展計畫在安全性這方面逐漸受到重視。

- **惡意檔案文件分析系統**

對於本研究的議題，分為國內外相關研究、與偵測動態取得堆疊記憶體技術兩部分來說明。本研究將會先簡介國內外的相關研究與開發系統。接下來介紹本系統將利用偵測動態取得堆疊技術，來辨識內嵌 Shell Code 的前置手續，如何使用虛擬 CPU 來得知是否有此行為，即可判斷該程式有惡意行為的可疑性。

靜態與動態分析

靜態分析比對惡意程式特徵碼[50]，一直是市面上防毒軟體普遍使用的分析手法。對已知惡意程式檢測來說，這種靜態的比對方式是快速且有效的方式。但由於缺乏彈性，卻對於新型的惡意程式攻擊手法無法即時招架。資安人員致力於分析新型惡意程式行為特徵，盡力縮短惡意特徵碼更新與新型惡意程式釋出的時間差，以減輕其所造成的危害。但惡意程式發展技術日新月異。常使用加密、加殼的方式保護其原始碼，防止被分析知曉其攻擊行為與抵抗惡意特徵值檢測，增加資安人員分析的困難度。

市面上已開發出檔案分析軟體，例如：PEiD，此工具將其特徵值資料庫與可疑程式執行檔進行比對，判斷此可疑程式是經由何種工具加殼加密，以提供 相對應之解碼工具讓使用者參考。尚有另一款軟體 PolyUnpack[54]，採取將可疑程式反組譯方式，但經過一連串靜態與動態分析流程，造成負擔過重。

利用動態分析方式解析惡意程式[52,53,55]，是目前的趨勢。由於無論惡意程式如何包裝自身型態，在執行期間，必定會將其對系統真正有害的程式原始碼與資料還原，存放於記憶體中。當資安人員以動態分析出惡意程式原始碼執行入口位址後，即可對惡意程式原始碼採取進一步的行為分析與特徵值取得的工作。

由於動態分析與靜態分析各有優缺點，本系統希望能夠運用各部份所長，來互補其分析的不足。以動態分析方式取得程式原始碼，因為無預先突破惡意程式的加殼加密防護牆，加速了對新型惡意程式分析的速度，成功縮短惡意特徵碼更新與新型惡意程式釋出的時間差，增加靜態分析比對特徵碼的可用性，讓資安人員於檢測可疑惡意程式的準確性大幅提升。

偵測動態取得堆疊記憶體技術

現在常見的惡意程式攻擊手法，會將惡意程式碼內嵌在檔案中，來逃避偵測，但它們在進行攻擊的時候，無法事先預測自己會被分配到哪一塊記憶體區塊，對於一些與記憶體位址有關的變數或要 Jump 的目標位址，如果還是按照最初編譯時的位址來尋址，必將導致尋址錯誤，使得惡意程式無法正常的運行 。

因此，惡意程式在攻擊的時候，必預先得到自己這隻程式現在在記憶體中的位置，來重新定位那些跟記憶體位址有關的變數或要 Jump 的目標位址。大 多數隱藏於文件檔案中的惡意程式碼，為了得知自身程式碼所在的記憶體位置，必預先將目前的 Program Counter (eip值) 壓入堆疊後，再從堆疊中讀出，這種動作稱之為 Call/Pop 序列。

```
Call/Pop Example1：
        call getDelta
getDelta：
        pop ebp
        sub ebp，offset getDelta
.....
```

如上面的範例，惡意程式碼就能利用利用 Call/Pop 序列，先用 Call 將去取 得 Program Counter(eip 值)，將其壓入堆疊後，再從堆疊中讀出。

在得到現在的記憶體位址之後，惡意程式就會去呼叫 Win32 的API 來對系統進行破壞的動作，此動作稱為 Hook API。惡意程式在做 hook API 的時候，通常會去推算 kernel32.dll 的位置，此位置可利用 FS 暫存器中的SEH，TEB，PEB 資料結構來獲得 kernel32.dll 的記憶體位置。

得到 Win32 API 的記憶體位址後，惡意程式所需使用的 API 的位址可以 透過解析該 dll 的導出地址表(Export section)和導入地址表(Import Address Table，IAT)等方法來取得，也可先在 kernel32.dll 中搜索 GetProcAddress 的地址， 再用 GetProcAddress("Win32 API 名字"，"函數名字")得到其他的 API 地址。所需的 API 也可透過 LoadLibrary 來加載。如此一來，內嵌在普通文件裡的惡意程式，就可以對系統造成破壞。

- **程式漏洞檢測**

目前，國內外的研究方向主要在Gray-Box Fuzz Testing上面，比較有代表性的研究成果有：

- Koushik Sen, Darko Marinov, Gul Agha. CUTE: A Concolic Unit Testing Engine for C. ESECFSE 2005.
- Cristian Cadar, Vijay Ganesh, Peter M. Pawlowski, David L. Dill, Dawson R. Engler. EXE: Automatically Generating Inputs of Death. CCS 2006.
- David Molnar, David Wagner. Catchconv: Symbolic execution and run-time type inference for integer conversion errors. Electrical Engineering and Computer Sciences, University of California at Berkeley, Technique Report No. UCB/EECS-2007-23.
  http://www.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-23.html
- Kuen-Han Huang, Shin-Kun Huang. Detecting Buffer Overflow Vulnerabilities by Search-based Testing. CISC 2010.

CUTE（Concolic Unit Test Engine）是一套 C/C++程式檢測系統，其檢測 Java 程式的版本為 jCUTE。CUTE 利用 Concolic Testing 技術去檢查目標程式，其特點在於能夠支援追蹤複雜的資料結構，例如：pointer、struct、class，對於經常大量使用 pointer 及自行定義 class 的 C/C++程式，檢測上有極大的幫助。EXE（EXecution generated Executions）則是由美國史丹佛（Stanford）大學提出的系統，也是使用 Concolic Testing 的技術，利用程式執行時期的資訊，去嘗試產生會觸發程式漏洞的測試資料，比起一般的 Run-time checking tool 更為有效，同時產生出來的測試輸入也提供程式設計員除錯上的幫助，能夠更快速地找出有漏洞的程式碼。Catchconv 則是由美國加州大學柏克萊分校（UCB）EECS 的教授 David Wagner 和他的博士生 David Molnar 一起研發的系統，以 Valgrind 為 instrumented framework，在上面開發的 Concolic Testing plug-ins，除了使用程式執行時期的資訊，產生觸發程式漏洞的測試

資料外，還合併了動態的型別轉換檢查，對於不同型別變數之間的互相轉換，或是有號數無號數轉換所造成的數值溢位，都能夠做檢查；除此之外，Catchconv 特別著重於尋找記憶體方面的程式漏洞，例如：記憶體緩衝區溢位，以避免這類型的程式漏洞成為系統上的安全弱點，造成重大的傷害。

三、 計畫目的

本計畫之目的在於提供一套完善的行動平台安全檢測方案，包含無線異質網路模擬平台、惡意檔案文件分析、程式漏洞檢測以及行動平台的安全管理機制，依相關文獻與研究做為參考，提出針對此四項研究子項的可行方案，以下將針對各研究子項所包含的具體研究內容進行說明：

- **無線異質網路模擬平台**

近年來，無線網路傳輸技術的蓬勃發展，使市場迫切需要一個能快速且正確的驗證新型網路技術與產品之實驗與測試環境。此外，目前市面上之產品逐漸以整合許多無線網路型態的異質無線網路服務為主，針對此類異質無線網路之安全議題，除了必須考量各基本網路型態之固有安全機制外，許多存在於不同網路銜接界面之間的安全問題亦極待解決。為建立一套能測試異質無線網路服務之實驗與測試環境，本計畫研究人員將建置一套能支援 Wi-Fi、WiMAX 等無線網路技術之異質無線網路模擬平台。

此模擬平台的開發將以 EmuLab、DETER 等有線網路測試平台為基礎，提供實體機器讓使用者自行配置所需實驗之網路拓樸，使運行於平台上之模擬實驗可兼具硬體設備之效能考量與實驗結果之真實性。為了支援 Wi-Fi、WiMAX 等無線網路技術，本平台使用模擬的方式仿真無線網路底層的訊號傳遞，實驗中的無線網路節點可藉由平台底層的有線網路將封包廣播至其他節點，來模擬無線網路在空氣介質中的傳輸行為。本模擬平台亦提供多套軟體工具，諸如使用者操作界面、無線網路攻防實驗模組等等，以協助研究人員利用本系統動態規劃無線網路實驗環境，並進行異質無線網路的安全測試及模擬可能的異質無線網路攻擊手法，藉此評估目前網路環境的安全性、可靠性，與可能存在的弱點。

本計畫也將延續和美國加州大學柏克萊分校 (UCB) 的合作，期望整合本計畫開發之異質無線網路模擬平台與 UCB 開發的 DETER 實驗平台。透過共享兩平台擁有的實驗節點，本平台可望提供更大規模(至數百節點規模)之異質無線網路實驗。最後，為推廣本平台之使用，本計畫也將撰寫詳細的開發文件和使用者操作手冊，將其以維基百科 (Wikipedia) 之方式公開在網路上，以降低本平台之使用門檻，並鼓勵學界與企業界於本平台上進行無線網路實驗，增加更多合作機會和產學效益。

- **行動平台的安全機制**

在探討行動平台安全機制上，為了預防可能的安全威脅，本研究之目的在於開發一套行動安全平台機制，包含以下兩項子系統:

  - **行動平台入侵偵測系統**

    在考量行動平台隨時隨地可利用無線網路上網的訴求，以及面對網路上層出不窮的攻擊，我們希望能在行動平台上建立一個有效率且可實用的防範系統，其中一種最有效也最多人使用的方式，就是使用 Intrusion Detection

System(IDS)，也就是所謂的入侵偵測系統，利用每種攻擊的特徵，對系統以及網路中的各項資料去進行比對。IDS 從偵測方式上可分為兩種，分別為 Anomaly-based detection 以及 Signature-based detection，前者是利用監控作業系統中的各項資源，去找出資源異常使用的地方，若有某項資源的使用量超出一般情況很多，則判定為異常，可能是受到攻擊或是使用者有異常行為，便會發出警告給系統管理員；後者則是利用字串比對的方式，去對要檢查的內容進行比對，以判斷是否含有會對系統造成危害之字串，從定義上防毒軟體可以視為被包含在這個分類裡面。從偵測目標上則可分為 Host-based IDS(HIDS)以及 Network-based IDS(NIDS)，顧名思義，前者是對主機去進行監控，而後者則是對網路封包去進行監控。其中 NIDS 大多都使用 Signature-based 的偵測方式。

本計畫預定將 Snort 移植到 Android 行動平台，Snort 為 Signature- based 的 NIDS，其為網路上很受歡迎的一款 IDS，不但功能強大而且還是共享軟體，其良好的軟體設計理念，使得 Snort 到現在還是非常多人使用，並且仍然有在進行功能更新。我們可利用 Snort 對行動無線網路傳輸做安全檢測與控管。無可諱言現有 PC 及 NB 平台上的安全機制可以移植到行動平台上，但是行動平台有其限制(受限於 CPU 計算速度、主記憶體容量、網路頻寬及電池容量等)，所以在設計相關的安全機制時，我們必須考量特殊的通訊環境與硬體規格，不能全盤照收，佔用太多資源的系統就不太適合，例如完整的防火牆。

- **安全雲端儲存系統**

本計畫預定設計出一套安全雲端儲存系統，可同時達到高度的安全性以及資料可靠度。OceanStore 在 2000 年就提出了分散式儲存系統的構想，目的是希望提供行動用戶們可以隨時隨地透過行動裝置去存取 OceanStore 中的資料。不過它不像是傳統的系統一樣將資料存放在固定的伺服器中，且每台伺服器都有防火牆保護。雲端的概念使得使用者能透過網路在任意時間任意地點，以及任意設備上獲取資料，而且會依伺服器不同的需要在任意時間點將資料搬動到其他伺服器上。假設資料是以明文儲存的話，這樣安全性上會有很大問題，另外如果伺服器上的資料有發生損毀的問題，那麼使用者下載完成的檔案也會發生問題。所以目前的儲存伺服器都必頇要具有檔案的完整性及保密性的功能。

儲存系統的目的主要是為了長期保存資料，會有系統資料損毀或是遭人惡意串改的情況，儲存上如有這些情況都是長期保存資料時所不允許的，早期的方式是將一份資料複製好幾份存到每台 storage server，或是一台 storage server，會有一台相對應的 mirror storage server(簡稱 MSS)，當資料毀損或被串改時，就可以從 MSS 上還原回來，但是以上種種方法的 overhead 都太高，所以 Erasure code 是較早提出的方法中具體達到容錯以及低 overhead 的方法。在 2006 年 Decentralized Erasure code 被提出，並用在分散式網路儲存

系統的架構中，緊接著在 2009 年 Secure Decentralized Erasure code 被提出，用以加強系統中資料的保密性。

事實上，在雲端系統上所提供的服務分為計算與儲存，本研究分的重點在安全雲端儲存系統，也就是分散式的儲存系統。當使用智者將資料上傳至雲端時，被以加密分散的方式放在許多雲端伺服器上，當使用者要需要時，只要有一定數量的雲端伺服器回應，就可以將資料取回，除此之外，使用者也可以將資料轉加密傳送給第三者。

- **惡意檔案文件分析系統**

本計畫亦開發一套惡意程式自動檢測系統，目的是讓檢測可疑惡意程式過程自動化並能綜合多項鑑識程序以提供完整的掃描報告利分析人員判讀。檔案首先將分成可執行檔與普通文件檔兩類。其次，系統將針對其檔案類型進行適當的二進位內容(Binary Analysis)靜態分析比對與檔案格式內容判定，標註程式中可供鑑識惡意行為的相關資訊，例如:含有可疑字串、檔案格式與副檔名的比對。動態的行為分析資訊為利用虛擬 CPU 技術來偵測動態取得記憶體位置之技術，來偵測內嵌程式碼的行為資訊。藉由蒐集以上靜態與動態的分析資訊，詳盡列出該程式所觸發的惡意行為。

- **程式漏洞檢測**

要有效地檢測出程式的正確性、安全性實為一困難的目標，即使有專業的人才、專門的團隊進行測試，也難以保證百分之百沒有漏洞。為了提升檢測工作的效率，減少檢測所花費的人力、時間成本，同時避免因程式漏洞所造成的系統安全問題，本計畫的目的包含一套程式漏洞檢測系統，能夠替使用者檢查所使用的軟體中是否有程式漏洞存在。如果有漏洞存在的話，將產生對應此漏洞的測試資料給予使用者，讓使用者可將此漏洞回報給原始程式的開發人員，進行除錯修補。

除了注重漏洞檢測的功效外，也必須要考量到檢測所花費的時間，才能夠有效率地完成目標。現行的 Concolic Testing 技術，都需要 Instrumented Framework 來監控程式執行，同時插入漏洞測試的動作進來；另一方面，Concolic Testing 會對檢測中遇到的每一個條件分支都做 Query，以產生對應的測試輸入。礙於這些原因，Concolic Testing 在檢測上需要不少的時間，在實際應用上也較不適合用在大型程式上。本計畫將考量到以上的因素，著重於 Concolic Testing 的效率最佳化，希望能更有效率的執行程式漏洞檢測，使 Concolic Testing 應用的範疇更廣。

四、 研究方法

本計畫包含了異質無線網路模擬平台、行動平台的安全機制、惡意檔案文件分析系統、與程式漏洞檢測等四個研究子項，對於本計畫的各子項，我們將分別採用以下之研究方法：

- **異質無線網路模擬平台**

本計畫所建置之異質無線網路模擬平台必須具備以下之功能：(1) 實驗之可控性與可重複性、(2) 無線網路模擬之真實性、(3)無線網路通訊技術之可擴充性與可移植性。因此，本計畫先以研究現有之網路測試平台為目標，探討現有測試平台架構，以及將其套用到本計畫之異質無線網路模擬環境的可行性。

在分析各平台之特性及優缺點後，我們歸納出以下幾點：(1) 採用實體無線網路裝置的實驗平台 (如：ORBIT Radio Grid Testbed) 雖具有較高之模擬真實性，其模擬之結果最接近真實世界中異質無線網路之傳輸行為。然而，此類平台擴充性低、建置成本高，尤其是新興的網路設備往往售價昂貴，欲於平台上測試新的無線網路技術所費不貲。而且，實驗易受到環境與訊號之干擾，缺乏可控性與可重複性；(2) 模擬無線網路裝置進行測試的實驗平台 (如 MobiNet) 的優點是具備高擴充性，使用者可於平台上開發新的模組來模擬新的網路協定與技術，但是藉由軟體模擬的實驗往往因只擷取部分系統屬性進行模擬而缺乏真實性，且無法考量因硬體設備所造成的效能瓶頸；(3) 現行之有線網路測試平台 (如 EmuLab、DETER 等) 採用一對一的方式，利用實體主機仿真有線網中的每個節點，於平台上建立之實驗具備可控性與可重複性，但缺乏無線網路之支援。

於上述三類之平台中，本計畫之研究人員認為於 EmuLab 等有線網路測試平台上擴充無線網路的訊號模擬，能達到所欲建立之異質無線網路模擬平台之所有功能。因此本計畫提出了虛擬驅動程式和虛擬天線的架構模擬無線網路的訊號傳送方式來補足有線平台之不足，其中：(1) 虛擬驅動程式負責與作業系統核心溝通，模擬無線裝置的網路控制層 (MAC layer) 的所有控制以及資料封包的處理工作；(2) 虛擬天線則用以仿真無線網路底層 (硬體傳輸層) 之介質傳送和資訊交換的工作，將來自虛擬驅動程式的無線網路封包 (802.11、802.16d) 封裝成有線網路 (802.3) 之 UDP 廣播封包，再傳送給所有實驗節點。使用虛擬驅動程式和虛擬天線的架構模擬無線網路，可確保進行中之無線網路實驗不受外界訊號之干擾。底層的有線網路平台使用之防火牆與 VLAN 技術，除了能保護進行中之實驗不互相干擾外，更可以確保在內部進行的攻擊實驗，不會擴散至外部公用網路。此外，本計畫模擬無線網路裝置 MAC 層與硬體傳輸層之設計，也可以方便使用者將新型無線網路技術開發成虛擬驅動程式或虛擬天線，移植到平台上進行協定之驗證與安全機制之測試，以提高異質無線網路測試平台之移植性與可擴充性。

本計畫延續和美國加州大學柏克萊分校 (UCB) 之合作，持續發展可提供異質無線網路模擬之實驗環境。為進一步擴展兩校之合作關係，本計畫透過加入 DETER

Federation (DETER 聯盟) 的方式共享 DETER 平台與本計畫之間的實驗機器，擴大本平台所能提供之實驗節點數量。加入 DETER Federation 之有線網路測試平台為一系列使用猶他大學所提出的 EmuLab 架構為基礎進行開發之網路實驗平台(例如：柏克萊大學的 DETER 平台、威斯康辛大學的 WAIL 平台等)，加入 DETER Federation 平台之使用者可透過 Federation 機制配置其他平台的實驗機器到實驗中，彌補單一平台實驗機器數量之不足。本計畫目前已完成加入 DETER Federation 所需之技術分析與測試，包含運行於實驗主機之作業系統的移植，802.11、802.16d 等無線網路協定的虛擬驅動程式和虛擬天線的驗證等等，本計畫所能運行之異質無線網路實驗皆可於 DETER 平台上進行重置；待本計畫通過 DETER Federation 之審查後即可正式與其他平台共享實驗節點，並將本計畫之無線模擬機制推廣到其他平台上，促進與其他各校之學術交流。

最後，為有系統地推廣本平台之使用，本計畫主持人與研究人員亦鼓勵台灣學界與企業界於本平台上進行無線網路模擬實驗。本計畫已完成詳細的開發文件和使用者手冊，並以維基百科 (Wikipedia) 之形式公開於本計畫之官方網站上 (http://www.swoon.cs.nctu.edu.tw)。目前已公開的文件包含 (1) 平台硬體設備安裝與設定文件、(2) 平台軟體架構設計和安裝步驟、(3) 無線網路模擬實驗的使用教學等等。除此之外，本計畫也公開研究人員所發展之網路攻防模組及其原始碼，協助使用者快速學習如何在本平台上進行無線網路模擬實驗，期望吸引更多使用者到本平台上開發或測試新的無線網路協定或安全機制，以擴充本計畫所能支援之異質網路測試之廣度與深度。

- **行動平台的安全機制**

針對探討行動平台的安全問題、安全雲端儲存系統以及行動平台的入侵偵測系統分別提出我們的研究方法:

(1) 探討行動平台的安全問題：

研究現行各種行動平台，如 Android、 Windows Mobile、 XP 以及 Netbook 上的 Chrome 等，因為行動平台使用到許多開放的系統與資料，如何保障系統資料安全是重要的課題，而密碼及通訊編碼是目前較可行的作法。目前這方面的成果不少，本研究將綜合我們所收集的資料，仔細研讀並密集的討論相關的問題。透過相互之間的腦力激盪，使我能夠更清楚及深入的瞭解論文及所要解決的問題。本計畫從研讀相關的論文中，整理出重要的部分做成 survey 報告，每週由參與計畫的學生舉行論文研討，讓學生仔細研讀最新的論文，並做正式的報告。另外像是多參加國內外相關會議，與外界交流來擴大我們的視野，並收集最新的資料。透過和其他學者的交流可以增進我們瞭解短處，及增進自己的研究能力。

(2) 安全雲端儲存系統：

本研究將探討如何將訊息明文安全地儲存在各自獨立運作的儲存伺服器之中，當我們需要時可以隨時取出來加以解密，儲存的資料還可以抵擋攻擊者的攻擊，保障訊息的隱私。本研究的重點在於如何將訊息明文加密成分散式密文，然後存入分散式系統的儲存伺服器中，當需要時，只要取出適當伺服器數內的

分散式密文，就可以結合解回原來的訊息明文。我們將先了解相關密碼工具 homomorphic signature 與 proxy re-encryption scheme 等的最新發展，再針對我們的系統需求進行客製化設計，最後進行正規的正確性與安全性分析。最後研究成果將撰寫為技術報告。而當我們有了理論及技術之後，需要建置一個雛形系統來測試我們的理論及開發相關軟體系統，以便和實際應證，如有不符合的地方，當檢討模型，然後修正理論，使系統的實際功能及效能達到預定的目標。

(3) 行動平台的入侵偵測系統：

在硬體規格上，一般 Android 手機的行動平台受限於 CPU 計算速度、主記憶體容量、網路頻寬及電池容量等限制，同時其強調隨時隨地上網，除了非常的耗費電力外，攻擊者可以隨時發動攻擊導致個人資料的外洩或成為攻擊的跳板，因此理論上在行動平台上的防護必須時時開啟。本研究的主要任務是如何將各類的防護系統整合放到行動平台上，並以其受限的硬體規格為考量，進而設計與系統整度高的安全機制。關於防護系統，如防火牆、入侵偵測、弱點掃描、惡意程式檢測等系統，我們打算用自由及共享軟體為基礎，將其輕量化後放到行動平台上。

- **惡意檔案文件分析系統**

本子研究將分成以下三個部分：(1)可疑字串檢驗、(2)檔案格式與副檔名檢驗、(3)虛擬 CPU 以偵測動態取得記憶體位址。以下將對各項做深入的說明：

(1) 可疑字串檢驗

在一般文件檔案或可執行檔中，字串會以 ASCI 碼方式保留在檔案內容。透過每個位元組位移的掃描方式，可以確定該檔是否有符合的可疑字串。利用正規表示法的特徵來找出事先設定的字串。為了往後的擴充性，設定檔是獨立出來的，供使用者往後可以針對不同的檔案內容特徵做檢查。以下為目前檢驗的字串格式：

a. IP 格式
b. 網路連結
c. windows 檔案路徑
d. unix-like 檔案路徑
e. iframe tag
f. embed tag
g. windows 可執行檔

由於程式行為很難直接判定是否為惡意。但若檢驗檔案的行為理應簡單而可預知，例如：影片、Office 文件、小遊戲…等，但卻檢查出有包含可疑的字串格式，雖有可能本身沒有包含惡意行為，但也有可能為首頁綁架、惡意軟體下載器等間接的木馬程式。我們實作了可疑字串檢查模組，將能找出類似可疑的字串，協助我們對檔案的判斷。

(2) 檔案格式與副檔名檢驗

微軟的檔案系統都會包含副檔名以得知和哪個軟體有關聯性，而該副檔名也成了騙取使用者執行的幫兇。若一個檔案格式與其副檔名不同，將有欺騙使用者之嫌。攻擊者可能把惡意執行檔替換成一般文件，以降低使用者或防毒軟體對該檔的警戒心，但必要的時候讀取該檔內容將成為攻擊程式。整合 Liunx 的 file 指令來協助檔案判斷，file 可以判斷數千種檔案格式，其特徵碼達數萬行。我們找出較敏感的若干檔案格式如下：

a. MS Windows (DLL) : dll
b. MS Windows (GUI) : exe
c. ASCII text : inf
d. MS Windows (console) : exe
e. Macromedia Flash data : swf
f. Zip archive data : zip
g. ISO-8859 text : htm
h. PDF document : pdf
i. Microsoft Office Document : msi， doc， ppt， xls， shs
j. RAR archive data : rar
k. troff or preprocessor input text : py
l. python 2.5 byte-compiled : pyc
m. MPEG sequence : ico

由於上述的檔案格式較多人使用，也常見於檔案系統之中。我們實作以找出是否屬於上述檔案，而副檔名卻正確對應的檔案。此處檔案格式與副檔名的對應可由分析者自行增加，方法如後所述。

(3) 虛擬 CPU 以偵測動態取得記憶體位址

許多惡意程式會運用緩衝區溢位(Buffer Overflow)來插入 shell code，為了控制 program counter 的位址，惡意程式需要知道目標惡意程式碼的位址。而大部分的資料都會存放在堆疊當中，利用 CPU 指令 call、pop 可以得知運行時的堆疊資訊。為了能夠偵測該程式會不會刻意取出堆疊的回傳位址(return address)，此分析模組會包含一個模擬的 x86 CPU，來模擬堆疊與 CPU 暫存器的狀況。我們會在每個 CPU 指令 call 執行之後，在堆疊寫入一特定值，若該數值馬上被取出放入到暫存器內，則可以確定該程式具有 call、pop 取出堆疊位址的技術。不同於虛擬機器，此分析模組不用模擬所有的硬體裝置，只要簡單的模擬 CPU 暫存器與堆疊，故能減少其他較不必要的運算開銷。由於 CPU 模擬器的實作較複雜冗長，我們將會結合利用 python 開發的 pyemu 來縮短此分析模組的開發時間。藉由著修改 pyemu 的系統，觀測模擬運算時記憶體的狀態，來實作出偵測取得堆疊位址資訊的可疑檔案。

此部分將會檢查出一般文件檔案是否具有堆疊位址取得的技術。給予一個欲檢測的一般檔案，本分析模組能夠判斷該程式是否會刻意的存取堆疊的內容。如果該執行檔試圖取出堆疊的內容，將會被此系統偵測出。

- **程式漏洞檢測**

我們計畫以美國加州大學柏克萊分校（UCB）研發的 Catchconv 為基礎平台，進行關鍵技術的研發與檢測系統的開發。Catchconv 是目前相當優秀、技術先進的程式檢測系統，同時它也是公開原始碼的，在上面進行技術研發，除了能夠方便取得大量有益的資源外，也可習得許多先進的檢測技術；更重要的是，能夠推廣公開原始碼軟體與提升台灣資訊軟體在公開原始碼上的參與度與貢獻度。

在 Catchconv 的檢測過程中，我們須先提供一個初始的測試資料，用來當作被檢測程式的輸入。Catchconv 將會監控被檢測程式的執行運作，檢查是否有潛在的程式漏洞，並產生可引發此程式漏洞的測試資料；Catchconv 也會根據目前程式執行過程中的變數內容與條件執行流程，產生能夠執行到尚未執行過的程式區塊的測試資料。對於被檢測程式中的程式碼敘述（例如：Buffer[i]=0;），Catchconv 將會嘗試產生可能的測試資料，使得索引 i 超過 Buffer[ ]的大小，企圖引發記憶體緩衝區溢位；另一方面，對於程式執行中的條件執行流程（例如：if (i>0) OOO else XXX），如果目前程式執行的是 i>0 這一條路徑，Catchconv 將會嘗試產生可能的測試資料，使得變數 i<=0，企圖讓被檢測程式下次執行另一條路徑。

我們發現到在 Catchconv 的檢測過程中，迴圈將會造成許多重複的條件執行流程，這會讓 Catchconv 的檢測變得較無效率。例如：對於迴圈 for (i=0;i<10000;++i)，Catchconv 將會面對 i<10000 這樣的條件執行流程大約一萬次（i=0、i=1、…、i=10000），然而這些條件執行流程所產生的測試資料，事實上卻只是影響了迴圈的執行次數，並沒有帶領我們發現、檢測新的程式區塊。根據以上的觀察與發現，我們希望能夠偵測出被檢測程式中，哪些條件執行流程是由迴圈造成的，這樣我們便可讓 Catchconv 在檢測程式的過程中，不會陷於迴圈的陷阱。

我們首先進行 Catchconv 的程式碼修改，使其能夠輸出檢測過程中所遇到的條件分支資訊，包含目前的條件分支是否被執行以及此條件分支的目標位置。接下來開發迴圈偵測程式，利用條件分支的資訊，判斷哪些條件分支是由迴圈所造成的，並輸出可以被省略的 Query 清單供 Catchconv 使用。之後我們修改 Catchconv 的主程式部分，使其在進行檢測的過程中，能夠利用可省略的 Query 清單，不用對所有的條件分支進行 Query，節省 Query Solving 的時間，增進整體效率。

除了迴圈偵測及 Query 最佳化的部分之外，我們注意到 Catchconv 在不同回合的檢測過程中，有許多是之前的回合中已經被 Query 過，又再度被重複 Query 到的條件分支，因此我們同樣希望可以辨識出這些重複的 Query，以省略再一次 Query 所造成的浪費。

我們同樣利用條件分支的資訊，紀錄哪些條件分支是已經被 Query 過的。我們修改 Catchconv 的主程式，使其利用 Prefix Tree 來紀錄目前為止已經被 Query 過的條件分支，如此便能對目前所遇到的條件分支進行檢查，判斷是否已經被 Query 過而可以被省略，以達到我們的目的。

貳、 計畫項目完成度

以下依照本計畫所涵蓋的無線異質網路模擬平台、行動平台安全機制、惡意檔案文件分析系統、程式漏洞檢測系統等四個子項，回報其工作執行狀況與其完程度。

- **無線異質網路測試平台 (完成度：100%)**

本子研究之工作項目完成度如下：

(1) 現有網路測試平台之分析 (100%)

本計畫依據：(1)實驗之可控性與可重複性、(2)無線網路模擬之真實性、(3)無線技術之可擴充性與可移植性等條件，分析現有之有線/無線網路測試平台之優缺點，尋找適用於本計畫所欲建立之異質無線網路模擬平台之技術與架構。根據分析的結果，本計畫之研究人員認為於 EmuLab 有線網路測試平台上擴充無線網路的訊號模擬，能符合本計畫所欲建立之異質無線模擬平台的所有要求： (1) Emulab 利用實體機器仿真實驗節點的方式可以確保模擬結果之真實性與實驗之可重複性；(2)有線網路平台的封閉測試環境可保障實驗之安全性與隔離性，避免實驗失控危害一般網路之使用； (3) 利用虛擬驅動程式和虛擬天線的架構仿真無線網路訊號之傳遞的方式，可彌補有線網路測試平台缺乏無線傳輸設備的缺點，並提高異質無線網路測試平台之移植性與可擴充性。

(2) 異質無線網路模擬平台架構設計與建置 (100%)

本計畫之異質無線網路模擬平台採用 EmuLab 所提出之有線網路測試平台架構，以輔助實驗機器與系統安全之管理。本平台之管理功能集中在指揮伺服器與使用者伺服器： (1) 指揮伺服器負責實驗之建置與管理，並利用虛擬區網技術控制網路交換器來區隔不同之網路實驗，達到實驗之隔離性； (2) 使用者伺服器主要之功能為管理使用者帳號以及使用者所建立的實驗，使用者可經由使用者伺服器利用 SSH 連線至其所建立之網路實驗節點。同時，本計畫亦採納 UCB 的 DETER 開發團隊之建議，使用防火牆隔離內部與外部網路，避免駭客入侵實驗中之主機或是內部網路之惡意程式失控擴散至公有網路。

(3) 異質無線網路模擬平台軟體開發 (100%)

本計畫提出了虛擬驅動程式和虛擬天線的架構仿真無線網路底層訊號之傳遞，以補足有線網路測試平台缺乏無線模擬功能之不足： (1) 虛擬驅動程式負責與作業系統核心溝通，模擬無線裝置的網路控制層 (MAC layer) 的所有控制以及資料封包的處理工作； (2) 虛擬天線則用以仿真無線網路底層 (硬體傳輸層) 之介質傳送和資訊交換的工作，將來自虛擬驅動程式的無線網路封包 (802.11、802.16d) 封裝成有線網路 (802.3) 之 UDP 廣播封包，傳送給所有實驗節點。本計畫亦開發了完善的使用者介面與基本的網路攻防模組，以協助使用者於本平台上進行無線網路模擬實驗。

(4) 整合本計畫平台與 DETER 平台之實驗資源 (100%)

本平台採納 UCB 的 DETER 團隊所提出 DETER 聯盟架構 (DETER Federation Architecture)，與 DETER 平台分享彼此之實驗資源。本計畫目前已完成加入 DETER Federation 所需之技術分析與測試，已及運行於實驗主機上之作業系統/虛擬驅動程式/虛擬天線之移植和驗證，確認本計畫所能運行之異質無線網路實驗皆可於 DETER 平台上進行重置，待本計畫通過 DETER Federation 之審查後即可正式與其他平台共享實驗資源。

(5) 異質無線網路模擬平台開發文件和使用者手冊撰寫 (100%)

為加強推廣本異質無線網路模擬平台予國內產學各單位，本計畫之研究人員亦著手撰寫本異質無線網路模擬平台之說明文件，包含詳細的軟硬體架構設計與使用者手冊，並已維基百科的方式公開至本平台之官方網站 (http://www.swoon.cs.nctu.edu.tw)，期望吸引國內外之研究學者於本平台上開發新的網路協定與驗證其安全機制。

- **行動平台的安全機制 (完成度：100%)**

本子研究之工作項目之完成度如下：

(1) 探討行動平台的安全問題 (100%)

研究現行各種行動平台，並研讀相關的安全議題及相關論文。目前已完成相關背景研究。

(2) 安全雲端儲存系統 (100%)

已完成一套兼顧資料安全性與可靠度的雲端儲存雛型系統，並有相關期刊及會議論文產出。

在系統設計理論上的研究中，我們的重點在於如何將訊息明文加密成分散式密文，然後存入分散式系統的儲存伺服器中，當需要時，只要取出適當伺服器數內的分散式密文，就可以結合解回原來的訊息明文。更進一步來看，如果解密金鑰是由一些安全伺服器所共同持有，每一金鑰伺服器擁有一把金鑰持份。當要取出分散式密文時，我們希望金鑰伺服器利用其金鑰持份作分散式的解密動作得到訊息明文持份，然後使用者再將訊息明文持份結合成訊息明文。多個訊息明文經過加密後，送到獨立運作的儲存伺服器中儲存，儲存伺服器將這些資料作一些特殊的運算後儲存起來，類似網路編碼的功能，以節省儲存的空間，當使用者要取出資料時，命令她的金鑰伺服器從儲存伺服器取出資料作部分解密，然後由使用者作最後的計算，得到多個訊息明文。要特別注意的是，儲存伺服器是獨立運作的，沒有相互聯繫，金鑰伺服器也是獨立運作，每一個金鑰伺服器隨機從數個儲存伺服器取出儲存的加密資料，然後利用其金鑰持份對資料作部分解密的動作，得到部分訊息明文。

在系統的實現的研究中，我們加入控制伺服器來管理使用者存取的權限以及儲存伺服器的數量，另外還幫忙使用者將一份資料傳給數個儲存伺服器，因為我們假設使用者端的網路頻寬是相當有限的，不像是 server 端的網路那麼好，因此我們控制伺服器將幫我們決定傳送哪幾個儲存伺服器，我們這樣的方法不僅僅只有節省了用戶傳輸的流量更是節省遠端儲存伺服器的儲存成本，另外安全方

面我們更是確保了資料的完整性及隱私的問題。而我們所建立的雛型系統也具有以下優點:

(a) 系統能有效率地將檔案加解密以及儲存，同時也提供網頁介面讓使用者能輕易管理檔案。

(b) 系統位於應用層，可包裝成 Android 的安裝檔.apk，使用者可簡單地安裝、卸載、及維護。

(c) 本系統可支援多種作業系統與多種無線裝置，亦有利於移植至不同平台。

(3) 行動平台的入侵偵測系統 (100%)

研讀現行 Android 系統的文件及研究其相關的安全議題，而我們了解到手機在安全性議題上有很多要注意的地方，諸如：手機上的資料是否安全儲存，是否有外洩的可能性、以及手機上的MMS攻擊或Bluetooth攻擊、在無線網路上傳輸的資料是否安全，是否有接收到具有攻擊字串的封包，等等...。大致可以分類為儲存資料的安全問題，以及是否有受到外來攻擊之安全問題，經過我們的研究發現，在其它智慧型手機上已經有類似防火牆以及防毒軟體之類的應用軟體出現，但是 Android 上卻只有看到 iptables 之防火牆軟體，以及 tcpdump 之封包擷取軟體，而沒有能夠檢查封包並且發出回報之網路安全軟體，故此，我們才決定將 Linux 上許多人所使用的 Snort 移植到 Android，期望能利用 Snort 強大的功能，去確保我們手機使用上的安全性，也作為 Android 上第一款封包檢測安全軟體，並開發各種應用。

(a) 而在此工作項目中我們成功將 Snort 移植到 Android 行動平台上，此系統為 Android 上第一套具有高度實用價值之入侵偵測系統，同時並有相關會議論文產出。而此項成果具有以下優點:

(b) 為目前已知文獻中首度將入侵偵測系統(IDS)移植至 Android 手機，可加強行動裝置無線上網的安全性。

(c) 電源消耗量低，低耗電量有利於常駐使用。

系統設計上許多方面的可擴彈性大，像是支援入侵行為特徵資料庫的擴增以增強其安全性、前後端系統易於抽換或獨立升以及記憶體使用量依據載入之特徵資料庫大小決定等。

- **惡意檔案文件分析系統 (完成度：100%)**

惡意檔案文件分析系統於今年度開發完畢，符合預期進度。本系統主要以分析非執行文件的檔案安全性為主，而採用數種靜態與動態分析的方式，來達到檔案安全性的驗證。而本子研究之工作項目之完成度如下：

(1) 可疑字串分析 (100%)

找尋包含在檔案內的字串，檢查是否有包含較敏感的資訊。由於檔案行為很難直接判定是否為惡意。但若檔案的行為理應簡單而可預知，例如：影片、Office 文件、小遊戲...等，但卻檢查出有包含可疑的字串格式，雖有可能本身沒有包含惡意行為，但也有可能為首頁綁架、惡意軟體下載器等間接的木馬程式。我

們實作了可疑字串檢查模組，將能找出類似可疑的字串，協助我們對檔案的判斷。透過檔案指標傳遞，讓此分析模組取得欲檢測之檔案。經過每位元組位移的掃描方式，可以確定該檔是否有符合的可疑字串。利用正規表示法的特徵來找出事先設定的字串。為了往後的擴充性，設定檔是獨立出來的，供使用者往後可以針對不同的檔案內容特徵做檢查。

此部分將以正規表示式(Regular Expression)檢查檔案中是否含有可疑的字串片段，檢查項目包括：

(a) IP Address - 如 140.113.1.1、72.65.11.101

(b) URI - 如 http://www.hinet.net、ftp://fadsaf.dsafdsaf.dsafdsa

(c) Windows UNC Path - 如 c:\windows\、d:

(d) Linux Path - 如/usr/bin、/bin/sh

(e) iframe tag - 如<iframe……</iframe>

(f) embed tag - 如<embed……</embed>

(g) Windows 可執行檔名稱 - 如 winword.exe、command.com、autoexec.bat

(2) 蒐集各類檔案格式資料 (100%)

目前很多惡意檔案利用使用者對副檔名的誤判，來達到騙取使用者點選或是不適當的操作。本分析模組將會檢查該檔案的格式，是否和副檔名相同。由於正常的應用軟體產生檔案，並不會刻意的隱藏或改變該檔案的副檔名，故可利用此現象來過濾出副檔名與檔案內容格式不符的可疑檔案。對於欲檢測的檔案，我們整合Linux 的file 指令來協助檔案判斷，file 可以判斷數千種檔案格式，其特徵碼達數萬行。由於微軟的檔案系統都會包含副檔名以得知和哪個軟體有關聯性，而該副檔名也成了騙取使用者執行的幫兇。若一個檔案格式與其副檔名不同，將有欺騙使用者之嫌。攻擊者可能把惡意執行檔替換成一般文件，以降低使用者或防毒軟體對該檔的警戒心，但必要的時候讀取該檔內容將成為攻擊程式。本模組已完成，此系統可檢查該檔案之副檔名是否符合其檔案格式，且檔案格式與副檔名的對應可由分析者自行增加。

(3) 模擬 x86 CPU 及虛擬記憶體動態分析模組實作 (100%)

由於內嵌惡意程式開始對使用者的系統進行攻擊前必頇要先得到它現在在記憶體中的位址，而目前最常用的方法就是利用 Call/Pop 序列來取得此記憶體位址。我們所設計的分析系統首先會掃描目標檔案，利用 x86 反組譯器，將目標中每個位元組當作 x86 的指令碼解析，將其反組譯成 x86 程式碼，分析指令的類型，若為 Call/Jump 這種類型的指令，就停止繼續反組譯。找到第一個 Call 的指令後，我們會運行一個 x86 cpu 模擬器來去實際執行，第一個先後執行剛剛的 Call 指令，再將剛剛 push 進堆疊的 eip 值存起來，然後從他跳到的目標在檔案中的位址開始一直執行。最後再檢查是否曾經去讀取剛剛存的那個 eip 值，若有，則表示有發生 Call/Pop 這樣去取得自己現在在記憶體中的位址的情況，就極有可能是惡意程式。在模擬執行的過程中，分析程式將監測所有的記憶體讀取動作，當從記憶體中讀取出的值符合當初壓入堆疊的特殊值時，即代表偵測出一個 Call/Pop 序列。

以上為一簡單的 shellcode 程式碼中一開始的部分，在 01 行時進行 call 跳到 12 行的 code4 的位置，真正做 Call/Pop 的地方則是從 13 行的 call code2 跳回 02 行後接下來 03 行馬上執行 pop ebx，就可把剛剛 13 行的 call 所推進堆疊中的記憶體位址讀取出來，我們的分析程式能夠抓到這樣的行為。

```
Example:
01 jmp code4
02 code2:   ;call 到這邊繼續執行
03 pop ebx:     ;彈出位址，從此位址開始加密
04 xor ecx, ecx:     ;ecx 清成零
05 mov eax, 090909090h     ;eax 放入密鑰
06 code1:
07 xor dword ptr [ebx+ecx*4]，  eax ;開始加密
08 inc cx ;計數
09 cmp cx，  03ah ;計數比較 ebh/4=03ah
10 jl code1 ;没有計算完繼續計算
11 jmp code3 ;計算完則跳到 shellcode english
12 code4：
13 call code2
14 code3：
```

- **程式漏洞檢測系統（完成度：100%）**

本子研究之工作項目之完成度如下：

(1) 修改 Catchconv 系統與開發程式碼漏洞檢測最佳化技術 （100%）

本計畫針對 Catchconv 開發了程式區塊迴圈處理的演算法，能偵測重複且可省略的 Query 動作，以增加其執行效率與漏洞檢測準確率。本計畫也針對各個回合之間，之前已經處理過而重複可以省略的 Query 動作，修改了 Catchconv 的主程式，使其能夠紀錄、省略這些 Query 動作，以增加檢測效率。

本計畫所開發之迴圈偵測程式，由於只以條件分支是否被執行以及條件分支目標位置來分類，因此能夠支援 for、while、do-while、goto、function call 等各種型態之迴圈，也能夠支援偵測複雜的巢狀迴圈結構。開發之迴圈偵測程式也支援可調整之參數設定，使用者可以依照自身的需求，調整不同的迴圈偵測強度，以適應各式各樣的受測程式；在可省略之 Query 清單中，也同樣支援可調整之參數設定，使用者可以設定迴圈所產生的 Query 中，哪些部分要省略，哪些部分要保留，提供了豐富的彈性及自由度。

本計畫修改 Catchconv 主程式，使其利用 Prefix Tree 紀錄已經 Query 過的條件分支，當以後再次遇到此條件分支時，即可省略此 Query 動作。然而，當一程式長

度（步數）為 n 時，其對應之 Execution tree 大小最大可能為 2n，此時將無法在系統記憶體中儲存此 Prefix Tree。所幸，我們通常不會也無法去花費 Exponential Time 的時間去檢測程式，因此，我們只考量在 Polynomial Time 內的漏洞檢測所能夠達到的效用，而不考慮花費 Exponential Time 去將所有可能的執行狀態做檢查。

(2) 開發程式自動檢測雛型系統 (100%)

以上述程式區塊迴圈處理演算法為主體，本計畫已修改 Catchconv 主程式，使其能夠輸出所需之條件分支資訊，並且完成迴圈處理演算法之實作，以 tools 的方式與 Catchconv 完成整合，Catchconv 可於檢測過程中呼叫迴圈偵測程式，偵測迴圈所產生的條件分支，並省略其對應之 Query。本計畫也已完成於 Catchconv 主程式中，實作了 Prefix Tree 結構，以便紀錄已經 Query 過的條件分支，Catchconv 將會針對目前遇到的條件分支進行檢查，以決定是否該 Query 此條件分支。

此程式漏洞自動檢測雛型系統已完成開發，並與 Catchconv 專案進行整合，放置於 Source Forge 網站上：http://sourceforge.net/projects/catchconv/

使用者可利用 CVS（Concurrent Versions System）下載最新版本之 Valgrind-Catchconv：

在欲下載 Valgrind-Catchconv 的目錄下輸入以下指令：

cvs -d:pserver:anonymous@catchconv.cvs.sourceforge.net:/cvsroot/catchconv login

按下 Enter 鍵，使用空白密碼登入

輸入以下指令下載 Valgrind-Catchconv：

cvs -z3 -d:pserver:anonymous@catchconv.cvs.sourceforge.net:/cvsroot/catchconv co -P valgrind-catchconv

參、 計畫成果

本計畫執行所獲得之成果可分為國外交流研究、產學合作計畫、學術貢獻、系統建置等部分，最後並分析本計畫所提出技術方案之優越性。

一、 國外交流研究

本計畫於本年度執行期間已派遣四位優秀學生赴美國 UC Berkeley 進行交流研究，國外指導學者以及學生研究期間如下表：

學生姓名	訪問學校	國外指導學者	交流時間
彭博群	UC Berkeley	Prof. Doug Tygar Keith Sklower	2010 年春季
陳毅睿	UC Berkeley	Prof. Doug Tygar	2010 年春季
陳柏愷	UC Berkeley	Prof. Doug Tygar	2010 年秋季
王嘉偉	UC Berkeley	Prof. Doug Tygar	2010 年秋季

表一：交流研究學者及時間表

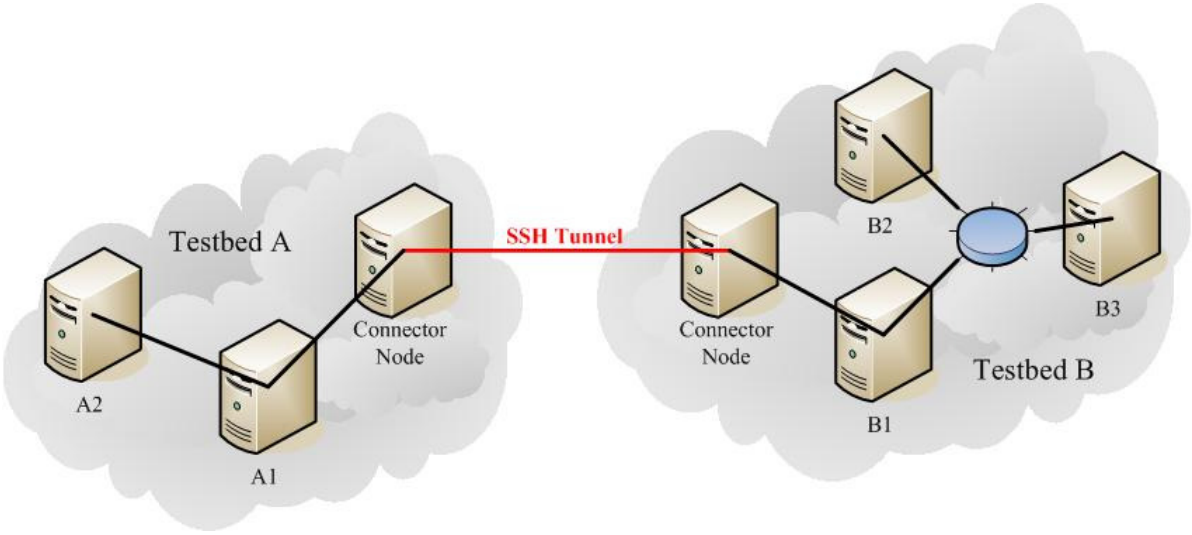
交流研究之題目除涵蓋本計畫所規劃之無線異質網路模擬平台、惡意檔案文件分析、程式漏洞檢測、及行動平台安全管理機制之外，亦對其他相關領域研究進行意見交流，其具體的交流研究項目如下：

- 安全雲端儲存系統設計與實作。
- 無線異質網路模擬平台與 DETER 之相容性分析與技術整合。
- 惡意文件檔案與內嵌程式碼分析。
- 程式迴圈偵測機制。
- Air Signature – The Finger Gesture for Security AccessGroup Key Management
- Group Key Management
- DNSsec Security Management
- Malware Detection and Behavior Analysis

以下將針對本計畫所包含各子項之交流研究內容進行說明：

- **異質無線網路模擬平台與 DETER 之相容性分析與技術整合**

DETER 聯盟架構 (DETER Federation Architecture，簡稱 DFA) 為 UCB 的 DETER 開發團隊所提出的有線網路測試平台之間的資源共享方式，使研究人員能夠同時使用多個基於 Emulab 所開發之網路測試平台的實驗機器，以進行大規模的網路模擬實驗。DFA 使用兩種不同的控制器來協助跨平台實驗之建立與管理： (1) 實驗控制器 (Experiment Controller) 提供使用者一個建置與管理跨平台實驗之介面，協助使用者取得與控制其他網路測試平台之實驗資源；(2) 存取控制器 (Access Controller) 負責處理遠端實驗控制器所發出之請求與協調本地測試平台之資源，並建立於跨平台實驗時節點與節點之間的溝通管道。存取控制器會為跨平台之網路模擬實驗在參與實驗之測試平台上建立連線節點 (Connector Node) ，連線節點間使用 SSH 通道的方式傳輸實驗的網路封包，如下圖四 所示。圖四 中的 A1 節點傳送封包給 B1 節點時，網路封包會透過 A 平台的連線節點的 SSH 通道傳送給 B 平台的連線節點，B 平台的連線節點再將封包送達目的節點 B1。



圖四：DETER Federation Architecture 與其 SSH 通道

本計畫所設計之異質無線網路模擬平台之底層使用 EmuLab 之有線網路架構，符合 DETER Federation Architecture 之基本需求。本計畫目前已完成加入 DETER Federation 所需之技術分析，並於本計畫平台移植與測試實驗控制器與存取控制器之運作。此外，為了加快雙方合作之腳步，本計畫研究人員亦將本平台無線網路實驗節點執行之作業系統移植到 DETER 平台，測試 802.11、 802.16d 的虛擬驅動程式和虛擬天線，確認本計畫所能運行之異質無線網路實驗皆可於 DETER 平台上進行重置；待本計畫通過 DETER Federation 之審查後即可正式與其他平台共享實驗節點，並將本計畫之無線模擬機制推廣到其他平台上，促進與其他各校之學術交流。

- **應用於空氣簽名安全驗證之影像追蹤演算法**

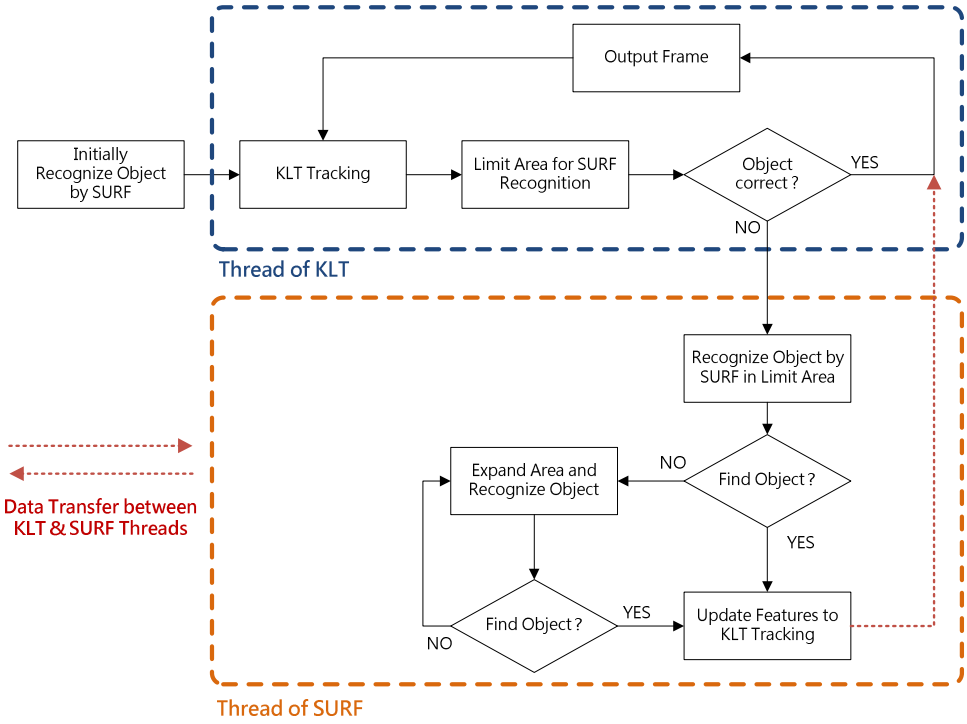
  (Air Signature - The finger gesture for security access)

本計畫之交流研究人員針對使用空氣簽名的門禁安全管制系統，開發了一套可即時追蹤並分析使用者手指動作的影像追蹤演算法，以增強系統之安全性。空氣簽名為一套結合影像處理之身分驗證機制，使用者可直接用手指在空中簽名以辨識身分；影像處理系統會追蹤受驗證者的手指動作，分析簽名的軌跡特性，並與資料庫中的簽名資料進行比對，驗證簽名者的合法性。

空氣簽名通常採用光流追蹤 (Optical Flow Tracking) 的影像處理技術來追蹤使用者的手指運動軌跡。光流追蹤為一種可以兼顧速度及效率之物體追蹤技術，但其追蹤準確率往往受限於環境明亮度、目標物遮蔽程度、目標物移動速度等因素。例如：KLT (Kanade-Lucas-Tomasi) 演算法雖然可以即時追蹤目標之影像，但若目標被部分遮蔽或移動速度過快，則 KLT 演算法容易因誤判而遺失追蹤目標；SURF (Speeded Up Robust Features) 演算法使用敘述子 (descriptor) 來增強物體辨識功能之穩健性，即使目標短暫消失或環境明亮度改變，亦能保持較高之追蹤準確率，然而 SURF 演算法的缺點是必須處理整張畫面，其計算量取決於影像之解析度，無法應用在即時系統上。

為了提升空氣簽名之辨識率，本計畫之研究人員開發了多執行緒的光流追蹤技術以提升影像處理系統之穩健性和準確率。多執行緒的光流追蹤技術始使用多核心處理器平行處理物體的追蹤與辨識，並交換兩者間的追蹤目標資訊，達到即時追蹤的目的。在本計畫中，研究人員以 KLT 作為基本的物體追蹤演算法，以 SURF 作為基本的物體辨識演算法。

圖五說明了本計畫追蹤與辨識運算之間的資訊交換流程： (1) 執行 KLT 演算法的執行緒負責目標追蹤以及顯示畫面更新，因 KLT 演算法的計算複雜度較低，能維持穩定之畫面更新率；KLT 演算法也負責縮小 SURF 演算法的影像處理範圍，降低 SURF 演算法的計算量；(2) 執行 SURF 演算法的執行緒負責在 KLT 演算法指定的畫面範圍內更新追蹤物體的特徵敘述，協助 KLT 演算法修正追蹤目標的軌跡，降低追蹤物體時產生的誤差；當追蹤之目標消失於畫面上時，SURF 演算法會藉由處理整張畫面的方式重新尋找追蹤之目標，確保追蹤目標回到畫面內時影像處理系統能快速的重新定位與追蹤。



圖五：平行運算之物體追蹤與辨識運算流程

- **行動平台的安全管理機制 – 安全雲端儲存系統設計與實作**

本計畫之交流研究人員已完成一套兼顧資料安全性與可靠度的雲端儲存雛型系統，並有相關期刊及會議論文產出。在系統設計理論上的研究中，我們的重點在於如何將訊息明文加密成分散式密文，然後存入分散式系統的儲存伺服器中，當需要時，只要取出適當伺服器數內的分散式密文，就可以結合解回原來的訊息明文。更進一步來看，如果解密金鑰是由一些安全伺服器所共同持有，每一金鑰伺服器擁有一把金鑰持份。當要取出分散式密文時，我們希望金鑰伺服器利用其金鑰持份作分散式的解密動作得到訊息明文持份，然後使用者再將訊息明文持份結合成訊息明文。以下圖為例，多個訊息明文經過加密後，送到獨立運作的儲存伺服器中儲存，儲存伺服器將這些資料作一些特殊的運算後儲存起來，類似網路編碼的功能，以節省儲存的空間，當使用者要取出資料時，命令她的金鑰伺服器從儲存伺服器取出資料作部分解密，然後由使用者作最後的計算，得到多個訊息明文。要特別注意的是，儲存伺服器是獨立運作的，沒有相互聯繫，金鑰伺服器也是獨立運作，每一個金鑰伺服器隨機從數個儲存伺服器取出儲存的加密資料，然後利用其金鑰持份對資料作部分解密的動作，得到部分訊息明文。已發表相關論文如下：

Yi-Ruei Chen, J. D. Tygar, and Wen-Guey Tzeng. “Secure group key management using uni-directional proxy re-encryption schemes”. in Proceedings of the 30th IEEE International Conference on Computer Communications (IEEE INFOCOM 2011)

在系統的實現的研究中，我們加入控制伺服器來管理使用者存取的權限以及儲存伺服器的數量，另外還幫忙使用者將一份資料傳給數個儲存伺服器，因為我們假設使用者端的網路頻寬是相當有限的，不像是 server 端的網路那麼好，因此我們控制伺服

器將幫我們決定傳送哪幾個儲存伺服器，我們這樣的方法不僅僅只有節省了用戶傳輸的流量更是節省遠端儲存伺服器的儲存成本，另外安全方面我們更是確保了資料的完整性及隱私的問題。而我們所建立的雛型系統也具有以下優點:

(a) 系統能有效率地將檔案加解密以及儲存，同時也提供網頁介面讓使用者能輕易管理檔案。
(b) 系統位於應用層，可包裝成 Android 的安裝檔.apk，使用者可簡單地安裝、卸載、及維護。
(c) 本系統可支援多種作業系統與多種無線裝置，亦有利於移植至不同平台。

- **惡意程式分析系統**

交流研究人員已針對惡意程式分析與國外指導教授進行意見交流，並已產生以下構想：任何程式都需要硬體資源來運行，惡意程式也不例外，利用虛擬機器技術可以輕易的掌控所有的虛擬硬體，從中觀察系統的狀態可觀察到所有需要硬體去運行的軟體，其中當然也包括了惡意程式。因此，不管隱匿式惡意程式用任何手法矇騙系統上層的使用者，在硬體層仍需暴露自己的資料，故基於虛擬機器內外比對的方法來偵測隱匿式惡意程式，將為一有效可行的手段。將實際運行於硬體上的資料跟系統上層使用者對系統的觀察兩相比較，若存在差異，表示惡意程式介入其中並隱藏部份系統訊息,為目前偵測隱匿式惡意程式的主流方法之一。本研究所產出之論文亦榮獲 CISC2010 之 Best Student Paper Award 肯定。

- **程式漏洞檢測系統**

Catchconv 是一套仍在開發中的開放原始碼程式漏洞檢測系統，有許多的開發子計畫正在進行中，目前已知的開發子計畫有：測試自動化、測試雲端化（Amazon EC2）、檢測狀態回報系統、程式漏洞回報系統、記憶體使用最佳化、Query 輸出格式轉換（SMT-LIB）、支援多樣化的漏洞類型、程式檢測最佳化（迴圈偵測、重複的條件分支）…等。

本團隊以與美國加州大學柏克萊分校（UCB）的合作研究為基礎，於今年度開發並實做了迴圈偵測元件以及重複條件分支偵測元件，並已整合進 Catchconv 母系統。藉由省略重複已偵測的程式區塊，以提升偵測過程中的 coverage rate 和效率。迴圈偵測的元件，能夠偵測 for、do、do-while、goto、function call 等各種迴圈種類，同時也支援巢狀迴圈結構，能夠自動地將各個迴圈部分辨識、擷取出來。偵測重複條件分支的演算法，利用 Prefix Tree 紀錄目前已經展開的程式執行樹節點，藉此得知哪些條件分支之前已經 query 過，哪些還沒被 query 過，使 Catchconv 能夠略過不去處理已經 query 過的條件分支。

本計畫今年更進一步進行了偵測演算法的測試與參數調校的工作，使 Catchconv 在程式檢測的過程中，能夠達到良好的效用。將偵測演算法實作於 Catchconv 之中，

須考量到實作上的效率，而在實作方法與偵測準確度上有所妥協；本計畫也透過實驗，進行偵測演算法的參數調整，找出適合的參數設定。

本計畫針對 Catchconv 所進行的程式檢測最佳化以及參數調整部分，皆已完成開發及實作，並整合於 Catchconv 計畫中，放置於 Source Forge 網站上，使用者可利用 CVS 下載經過最佳化的最新版本之 Catchconv：

1. 輸入以下指令登入 Catchconv 的 cvs 系統：

   cvs -d:pserver:anonymous@catchconv.cvs.sourceforge.net:/cvsroot/catchconv login

2. 按下 Enter 鍵，使用空白密碼登入
3. 輸入以下指令下載 Catchconv：

   cvs -z3 -d:pserver:anonymous@catchconv.cvs.sourceforge.net:/cvsroot/catchconv co -P valgrind-catchconv

- **DNSsec Security Management**

此部分主要的研究成果為論文-“New Threat comes with DNSSEC”。此論文探討的是 DNSSEC 技術當中，雖然加強了 DNS 的安全性，並有許多先進的安全性技術，然而其中的一個特性- NSEC3，卻隱含一個可能的安全漏洞。這個 NSEC3 的漏洞，可能造成資訊洩漏的問題，導致使用者的姓名、E-mail 等資訊被揭露，以致有 spam、social engineering attack 等後續的問題發生，本論文證明了這個漏洞確實存在，並提出解決方法。

此論文在出國前僅有基本構想，然而經過兩個月的交流研究，已完成了架構設計、程式實作、實驗數據等等。此論文目前草稿已完成，現階段與 Prof. Tygar 討論修改當中，可望在近期內定稿，並擬投稿於國際期刊。

二、 產學合作計畫

本計畫所涵蓋的研究範圍以及所發展的技術深具產業應用價值，本年度已陸續與友訊科技與中華電信等國內業界大廠簽訂產學合作計畫，其計畫名稱與合約金額如下表二：

合作計畫名稱	合作對象	合約金額
Open D-Link Routers Forum 建置、維護與測試技術與諮詢服務	友訊科技	1,500,000
惡意程式與檔案分析檢測系統研發計畫	中華電信	984,120
行動平台資通訊安全問題的研究	中華電信	940,000

表二：產學合作計畫總表

其計畫內容分別簡述如下列各表：

合作計畫名稱	合作對象	合約金額
Open D-Link Routers Forum 建置、維護與測試技術與諮詢服務	友訊科技	1,500,000
合作計畫內容簡述: 本計畫團隊於本計畫執行中獲得相當多的跨網路軟硬體開發與控制經驗，故 D-Link 委託本團隊針對其產品，建置並維護一個開放性的網路論壇，以期利用本研究團隊在網路與嵌入式系統方面所累積的研究經驗與能量，提供以下服務： ● 專注於討論與維護常見 Open Source Router Projects 移植到 D-Link Router 產品上所產生的相關議題。 ● 非官方性針對產品緊急問題提供即時解決方案、以及其 hot-fix firmware。 ● D-Link Router 產品相關技術分析與說明，以提升使用者接受度與信賴度。 ● 開創更多用於 D-Link Router 產品之新興應用。		

表三：產學合作-Open D-Link Routers Forum

合作計畫名稱	合作對象	合約金額
惡意程式與檔案分析檢測系統研發計畫	中華電信	984,120
合作計畫內容簡述: 惡意程式的攻擊行為越來越複雜，且具有各類多型和變種的能力，故目前特徵式(pattern-based)的偵測方式已經不堪使用。 本團隊與中華電信的合作計畫將以本計畫所研發出的惡意程式檢測技術為基礎，協助中華電信針對其實務需求開發一個惡意程式與檔案分析檢測系統。此研發計畫的目標是針對目前的惡意程式樣本分析惡意程式的行為模式，並利用行為模式分析配合動態觸發式(trigger-based)偵測法，此種檢測方式亦為未來趨勢。		

表四：產學合作-惡意程式與檔案分析檢測系統研發計畫

合作計畫名稱	合作對象	合約金額
行動平台資通訊安全問題的研究	中華電信	940,000
合作計畫內容簡述: 智慧型手機和行動小筆電是使用者主要的行動平台，透過高速的電信與資訊網路系統，使用者將可以在任何時間任何地點取得資訊服務業者提供的服務。 本合作研究計畫的主要目地是利用本團隊在本計畫中所累積的行動服務平台相關背景知識，進行針對行動平台的安全議題探討與研究，並協助中華電信在弱點掃描系統、入侵偵測與防禦系統、身份認證系統、防毒系統、安全雲端儲存等系統上的開發。 整體來說，本計畫將與中華電信合作，以本團隊的研究能量在三年的期間內與中華電信合作研發行動服務平台的安全及隱私機制與系統。		

表五：產學合作-行動平台資通訊安全問題的研究

三、 學術貢獻

- **期刊論文:**

1. Shih-I Huang, Shiuhpyng Shieh, "Secure Encrypted-Data Aggregation for Wireless Sensor Networks," accepted for publication, *ACM Journal of Wireless Networks*.
2. Chi-Wei Wang, Shiuhpyng Shieh, "The Evolution of Fine-Grain Malware Behavior Analysis -From Static to Dynamic," *IEEE ATR*, 2010.
3. Shih-I Huang, Shiuhpyng Shieh, "Secret Search Mechanism for Wireless Sensor Networks with Passive RFIDs," accepted for publication, *International Journal of Security and Networks*
4. Hsiao-Ying Lin, Wen-Guey Tzeng, "A Secure Decentralized Erasure Code for Distributed Network Storage," *IEEE Transaction on Parallel and Distributed Systems,* 2010.

- **會議論文:**

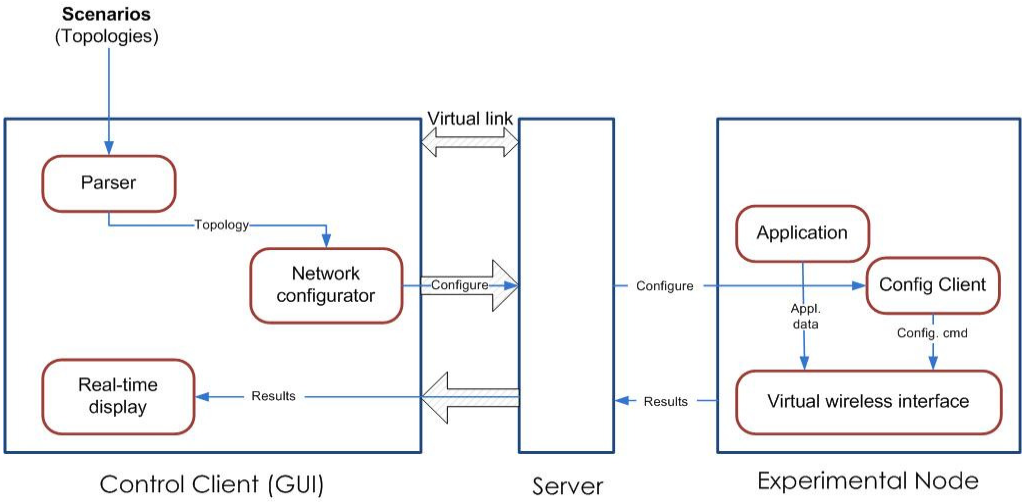
1. Chia-Wei Hsu, Shiuhpyng Shieh, "FREE: A Fine-grain Replaying Executions by Using Emulation," *The 20th Cryptology and Information Security Conference (CISC 2010),* Taiwan, 2010. **(Best Student Paper Award)**
2. 王繼偉、王嘉偉、許家維、謝續平，"基於虛擬機器外部觀察與映像檔比對的惡意程式分析，" *The 20th Cryptology and Information Security Conference (CISC 2010)*, Taiwan, 2010.
3. 蔡欣宜、王繼偉、陳柏廷、黃育綸、謝續平，"基於虛擬裝置之無線網路安全測試平台，" *The 20th Cryptology and Information Security Conference (CISC 2010)*, Taiwan, 2010.
4. Bo-Ting Chen, Yu-Lun Huang, "The Design and Implementation of a Multi-core Supported Network Intrusion Detection System," *The 20th Cryptology and Information Security Conference (CISC 2010)*, Taiwan, 2010.
5. 羅日宏、曾文貴、陳彥仲，"移植至 Android 上的入侵偵測系統，" *The 20th Cryptology and Information Security Conference (CISC 2010)*, Taiwan, 2010.
6. 顏豪緯、曾文貴，"以智慧型手機為使用者端裝置之安全分散式儲存系統實作，" *The 20th Cryptology and Information Security Conference (CISC 2010)*, Taiwan, 2010.
7. Yi-Ruei Chen, J. D. Tygar, and Wen-Guey Tzeng. Secure group key management using uni-directional proxy re-encryption schemes. in Proceedings of the 30th IEEE International Conference on Computer Communications (IEEE INFOCOM 2011)

四、 系統建置

本計畫所提出的各項技術為具體可行，且本計畫團隊亦按照計畫書之規畫實做出四套具高度實用性之系統，以下將分別就異質無線網路模擬平台、行動平台的安全機制、惡意檔案文件分析系統、與程式漏洞檢測等系統敘述其設計理念、實作方法、以及其實際執行畫面：

- **無線異質網路測試平台**

本計畫之異質無線網路模擬平台由四大元件組成：(1) 控制客戶端 (control client)、(2) 伺服器 (servers)、(3) 實驗節點 (experimental node) 以及(4) 虛擬鏈結 (virtual link)。使用者可藉由控制客戶端設計所需之無線網路實驗拓樸並指定實驗參數，控制客戶端將實驗網路之拓樸轉為 NS (Network Simulator，簡稱 NS) 設定檔後，傳送給伺服器。伺服器則依該 NS 設定檔之內容配置實驗節點並建立對應的網路拓樸。控制客戶端可以透過伺服器傳送實驗參數、指令至實驗節點，實驗節點會依使用者設定的指令進行實驗。詳細的系統架構如圖六所示：

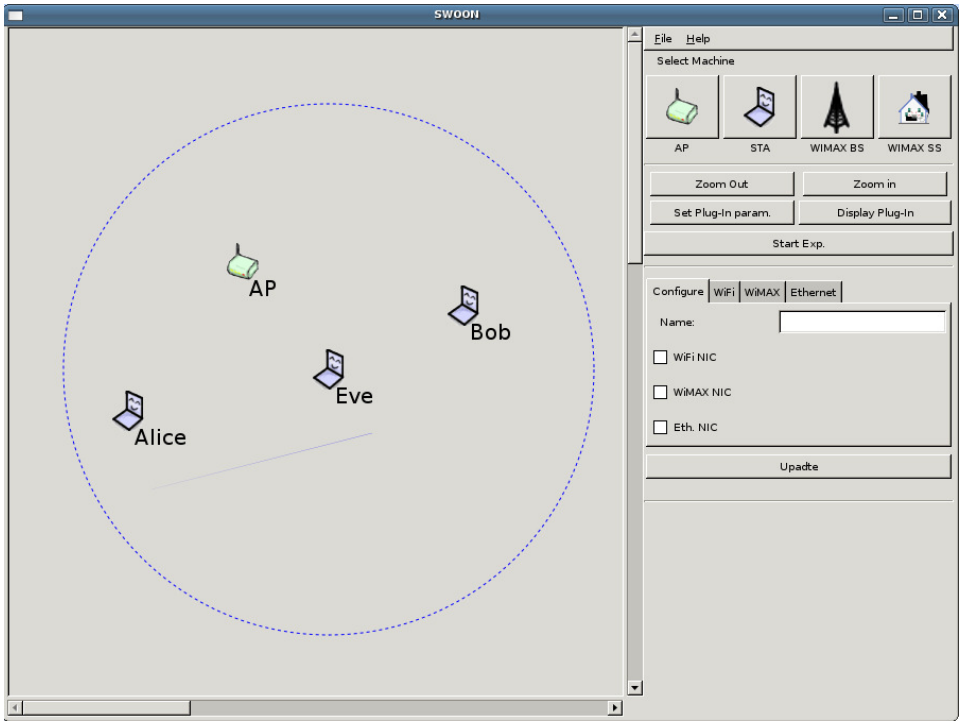


圖六：異質無線網路模擬平台架構

控制客戶端：

控制客戶端提供使用者一套圖形化的設定工具，用以設置實驗環境、開啟攻防程式以及觀測實驗結果，其畫面如下圖七所示。使用者可藉由控制客戶端的圖形化介面規劃所需之無線網路實驗拓樸和指定實驗參數；控制客戶端的分析器 (parser) 負責解析無線拓樸圖形產生網路參數設定；網路設定程式 (network configurator) 處理網路參數設定，並產出 NS 設定檔。即時顯示畫布 (real-time display) 則負責接收及動態呈現實驗結果。其中，NS 設定檔為 EmuLab 用來配置實驗機器的設定檔；伺服器接收到 NS 設定檔後，會依 NS 設定檔所提供之資訊，將有線網路節點配置為所要求的無線網路架構。NS 設定檔內存放之無線網路設定包含了：(1) 實驗節點在無線環境中的空間配置、(2) 支援之無線網路介面及其網路卡實體層位址 (MAC address)、(3) 無線裝置之訊號覆蓋範圍等等。待伺服器將實驗網路建置

完成後，控制客戶端會將每台實驗節點的設定資訊轉換成訊號覆蓋表 (coverage table)，並將其傳送至各個實驗節點。訊號覆蓋表紀錄了節點於虛擬空間中之距離，可以用來模擬傳輸距離與電磁訊號的衰退關係。



圖七、控制客戶端的圖形化設定工具可用於設定無線網路拓樸與實驗參數

伺服器：

在此異質無線網路模擬平台上，控制實驗用的伺服器主要可分為指揮伺服器 (Boss server) 和使用者伺服器 (User server)。

指揮伺服器負責解析使用者上傳的 NS 設定檔，建立新的網路實驗。指揮伺服器藉由控制網路交換器和電源控制器進行實驗節點之配置、創建實驗者設計的網路拓樸，並載入指定的可執行映像檔至選定的實驗節點。為了使同時進行中的網路實驗不會互相影響，指揮伺服器利用虛擬區網 (Virtual Local Area Network，VLAN) 技術控制網路交換器來區隔不同之網路實驗。透過將同一實驗之節點配置在相同虛擬區網的方式，同一虛擬區網內的實驗節點可以彼此溝通，如同這些節點都連接至同一實體線路；節點之間是否能夠溝通取決於虛擬區網的規劃，與節點的實際位置無關。

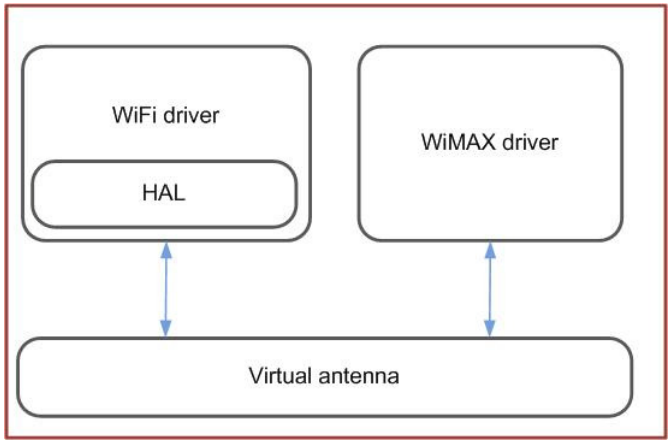
使用者伺服器的主要工作為管理使用者帳號以及使用者所建立的實驗，並作為使用者連線至其所建立之網路實驗節點之跳板：使用者必須透過 SSH 協定連至使用者伺服器，再由使用者伺服器登入實驗節點存取資料以及控制實驗。為了確保實驗之隔離性和安全性，本計畫以防火牆隔離外部和內部實驗網路，避免實驗中的惡意程式失控時，危害到外部公有網路。

實驗節點：

實驗節點包含三個主要元件：應用程式 (application)、設定客戶端 (config client) 和虛擬無線界面 (virturl wireless interface)。

應用程式包含了本計畫研究人員開發之網路攻擊、防禦程式，如 Wireshark 等網路封包分析工具、洪水攻擊 (Authentication Flood Attack) 等模組。使用者可於使用者伺服器上取得所有攻防模組進行無線網路實驗。

設定客戶端則負責處理控制客戶端傳送之無線網路設定和控制訊號，包含訊號覆蓋表、無線網路卡實體層位址 (MAC) 等等，設定客戶端亦可於實驗中即時接收使用者的動態設定，例如使用者在虛擬空間中移動無線裝置所造成之位置變化與訊號覆蓋範圍更新。



圖八：虛擬驅動程式介面由底層的虛擬天線和上層的虛擬驅動程式所組成

虛擬無線界面提供仿真模擬真實的 Wi-Fi 和 WiMAX 無線網路界面卡之方式，其設計主要可分為底層的虛擬天線 (virtual antenna) 和上層的虛擬驅動程式 (virtual driver)。圖八為虛擬無線介面之架構設計。以下茲簡介各元件之設計與功能。

a、 虛擬驅動程式 (virtual driver)

虛擬驅動程式可直接與實驗節點的作業系統核心溝通，負責網路實體層 (Media Access Control Layer，簡稱 MAC Layer) 的所有控制以及封包的處理工作。虛擬驅動程式接收到來自系統核心的資料後，會將資料以無線協定表頭 (header) 進行封裝 (encapsulate)，使其成為無線網路封包後，再交由虛擬天線進行無線訊號傳遞之模擬。當虛擬驅動程式接收到來自虛擬天線的封包後，它先解封裝 (decapsulate) 802.3 封包表頭，再將資料轉傳給作業系統核心及應用程式。本計畫實作了支援 Wi-Fi (IEEE 802.11 協定) 和 WiMAX (IEEE 802.16 協定) 的虛擬驅動程式。Wi-Fi 驅動程式的實作參考 Atheros 公司所生產的 Linux 無線網卡驅動程式修改而成，並重新實做了硬體抽象層 (Hardware Abstraction Layer，簡稱為 HAL)以提供上層統一的界面。WiMAX 虛擬驅動程式則為本計畫之研究人員遵循 IEEE 802.16 之標準，自行開發一套適用於 Linux 作業系統的 WiMAX 虛擬驅動程式。

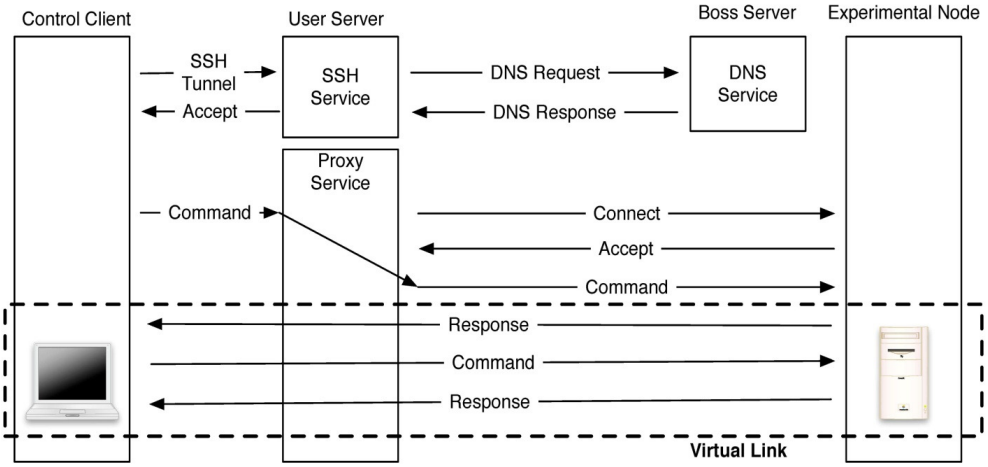
b、 虛擬天線

本計畫是基於有線網路測試平台而設計的無線網路模擬平台，因此原本的實驗節點僅支援有線網路 (802.3) 之傳輸協定，因此本計畫透過在實驗節點上運行「使用者資料包通訊協定」 (User Datagram Protocol，簡稱 UDP)

廣播封包來模擬無線網路封包於空氣介質中傳遞時的廣播特性。虛擬天線之主要功能為模擬無線網路底層之介質傳送和資訊交換等工作，將來自上層虛擬驅動程式的 802.11 或 802.16 封包封裝為符合 UDP 的廣播封包，再將 UDP 廣播封包交由實體乙太 (ethernet) 網路卡傳送至其他實驗節點。同理，當虛擬天線接收來自其他節點的 UDP 廣播封包後，會先解開 UDP 表頭，依據訊號覆蓋表過濾不在節點訊號傳輸範圍內之封包，再將剩餘之封包傳送至上層虛擬驅動程式。

虛擬鏈結

虛擬鏈結旨在建立控制客戶端和使用者伺服器之間的連線，這是由於在原本 EmuLab 的設計架構中，使用者只能經由指揮伺服器或使用者伺服器存取內部的實驗節點。因此，為了使位於實驗網路之外的控制客戶端能與內部之實驗結點進行溝通，本平台提出藉由在使用者伺服器上建立虛擬鏈結之方式，提供轉傳客戶端封包至實驗節點之代理服務。



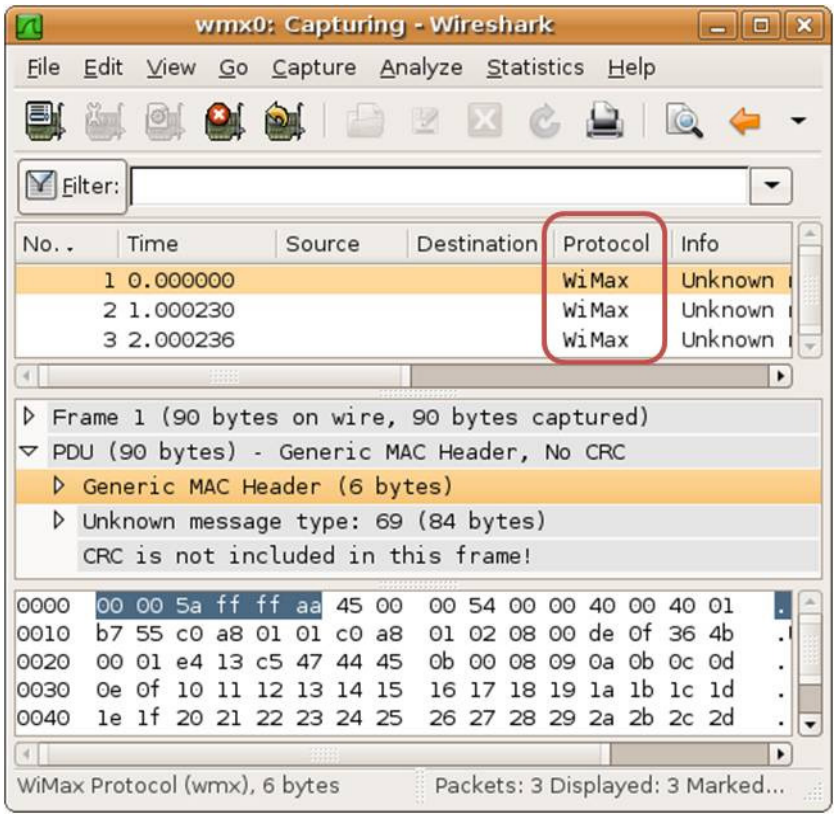
圖九：虛擬鏈結的訊息流程。

圖九說明了建立虛擬鏈結的訊息流程。使用者先於使用者伺服器上建立可觀測實驗結果和節點狀態的 SSH 通道 (tunnel)。SSH 通道使實驗節點可通過防火牆，透過代理伺服器 (proxy server) 與客戶控制端進行資料交換。控制客戶端透過代理伺服器將控制訊號傳送給實驗節點的設定客戶端，設定客戶端則負責將實驗結果透過代理伺服器回傳到控制客戶端的即時顯示畫布呈現。由於控制客戶端送出的指令嵌有節點名稱，指揮伺服器的網域名稱系統 (domain name system, 簡稱 DNS) 服務也會被要求協助轉送各種服務所需的指令。同理，DNS 服務的回應也將經由代理伺服器回傳至控制客戶端。

為協助使用者快速學習如何在本平台上進行無線網路模擬實驗，本計畫之研究人員開發了許多基本的網路攻防模組；使用者可在其進行之無線網路模擬實驗中直接套用這些工具觀察實驗結果或是驗證所提出之網路協定或安全機制。以下舉出兩個本計畫研究人員開發之攻防模組作為說明：

1. 無線網路封包分析工具：

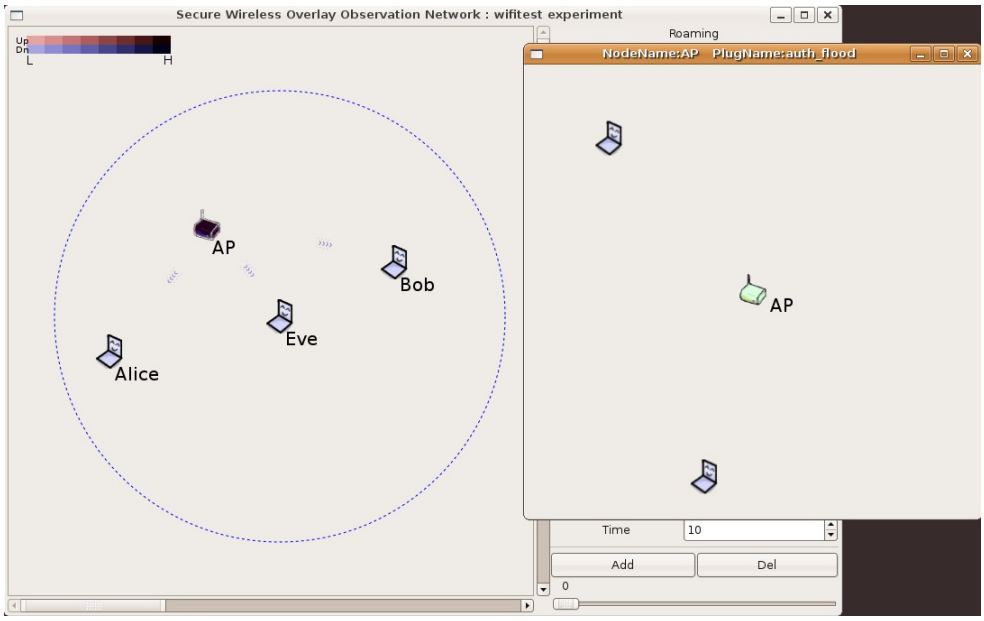
本計畫之研究人員移植知名的網路封包分析工具 Wireshark [6]至本計畫發展之無線模擬平台上，讓使用者能夠藉由 SSH X forwarding 的方式將 Wireshark 的畫面回傳至使用者的電腦端顯示。 Wireshark 能夠擷取無線網路裝置在空氣介質中傳輸的封包，並協助使用者了解網路之行為或是驗證新的通訊協定。此外，本計畫之開發人員亦修改了 Wireshark 的程式碼，使其能夠支援本計畫所模擬之 802.16d 無線網路協定之判別，協助使用者發展 WiMAX 之通訊協定，其畫面如圖十所示。



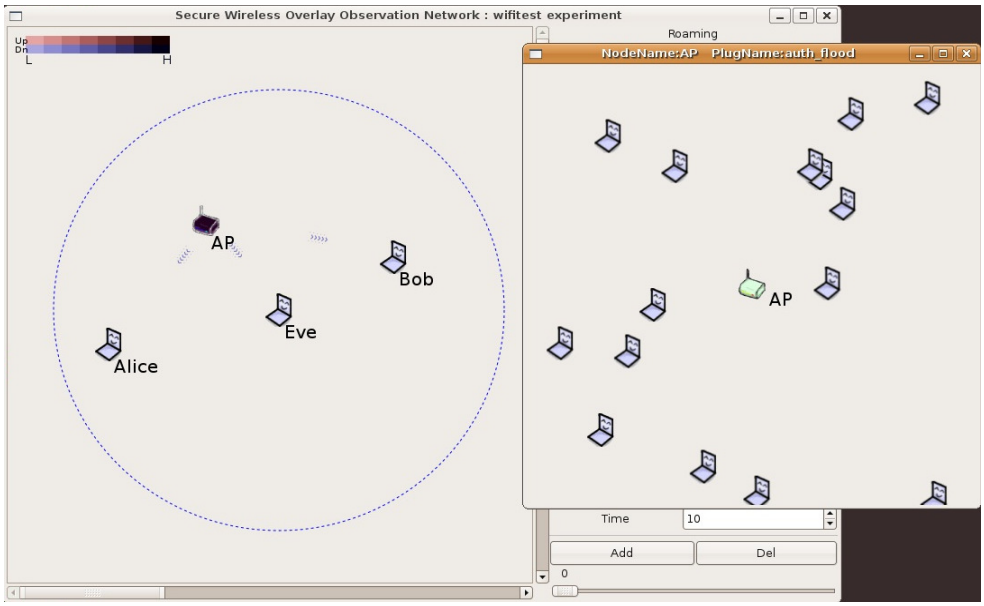
圖十：Wireshark 協助使用者分析 WiMAX 網路封包

2. 分散式阻斷服務攻擊模組

本計畫之研究人員為了驗證 802.11 的無線存取點 (AP) 是否能夠抵擋分散式阻斷服務攻擊，開發了洪水攻擊 (Authentication Flood Attack) 模組，企圖使 AP 拒絕正常使用者的連線請求。圖十一-(a) 顯示在洪水攻擊開始前與 AP 連線的節點數量 (也就是圖中的 Alice 和 Bob 兩個節點)，當 Eve 節點開始對 AP 發動洪水攻擊後，Eve 會不斷的更換網路卡實體層位址，並偽裝成新的節點和 AP 進行連線，企圖填滿 AP 的連線紀錄表，如圖十一-(b) 所示。當連線數量超過 AP 所能容許的上限後，正常的使用者及無法和 AP 進行連線，形成分散式阻斷服務攻擊。



(a)



(b)

圖十一：使用洪水攻擊模組驗證 AP 是否能抵擋分散式阻斷服務攻擊

最後，本計畫為將加強推廣本異質無線網路模擬平台予國內產學各單位，本計畫之研究人員亦著手撰寫本異質無線網路模擬平台之軟硬體設計文件與使用說明，其內容可分為開發者手冊與使用者手冊兩部分：

1. 開發者手冊

開發者手冊分為硬體架構及軟體設計兩部份。硬體架構包含異質無線網路平台所使用的設備規格、伺服器/防火牆/網路控制器/實驗機器之間的網路拓樸、網路控制器的設定和伺服器的安裝說明文件等等。軟體設計部分則包含了控制客戶端等程式之設計文件與原始碼，以及虛擬驅動程式/虛擬天線/攻防模組等軟體之架構設計、原始碼與安裝說明。

SWOON 的 source tree 包含以下幾個部份：

- GUI：控制客戶端的圖形化設定工具原始碼
- attack_monitor：本研究團隊開發之網路攻防模組原始碼
- config_client：設定客戶端的原始碼
- service_server：虛擬鏈結使用的代理伺服器原始碼
- wifi_driver：Wi-Fi 虛擬驅動程式原始碼
- wimax_driver：WiMAX 虛擬驅動程式原始碼

2. 使用者手冊

使用者手冊則提供完整的異質無線網路平台的使用教學，包含帳號申請與實驗操作兩部份。

帳號申請：

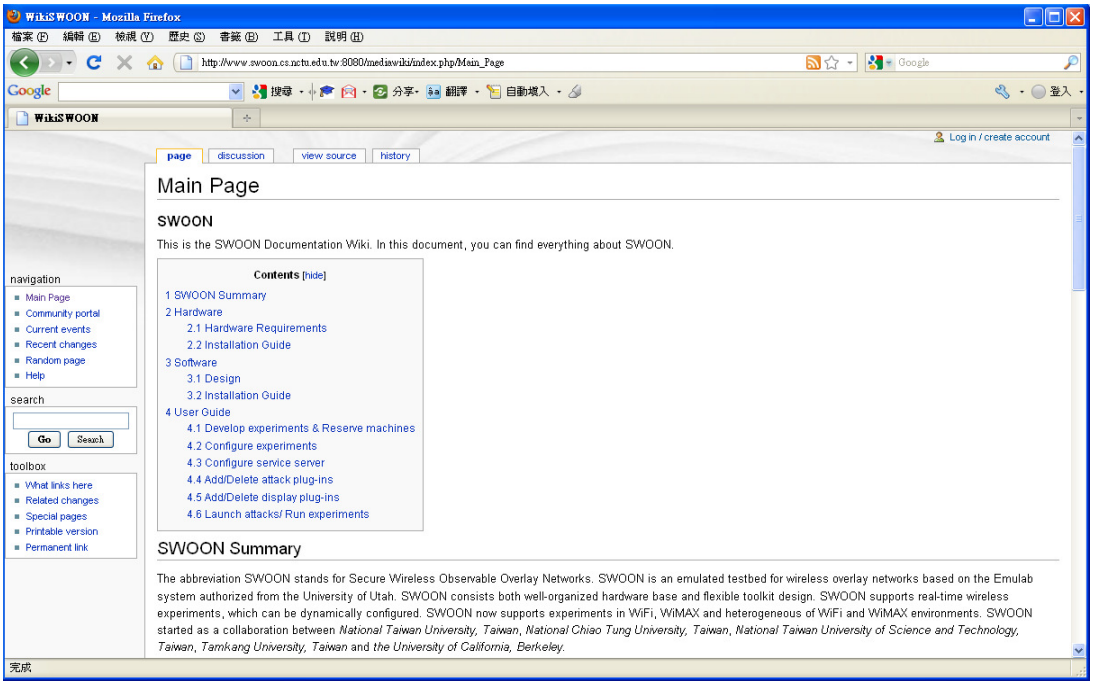
為加強帳號之管理，本計劃採用 EmuLab 的建議，對帳號申請採用嚴格控管的方式，以免本平台成為駭客網路攻擊之道具。申請新 Project 之使用者必須為學術計畫或研究之主持人，且須經過本計劃研究團隊之審查方可建立新的 Project，並成為該 Project 之主持人 (Project Leader)；一般使用者只能選擇加入現有之 Project，且須經由該 Project 的 Leader 之認證才能於該 Project 下開始建立網路實驗。

實驗操作：

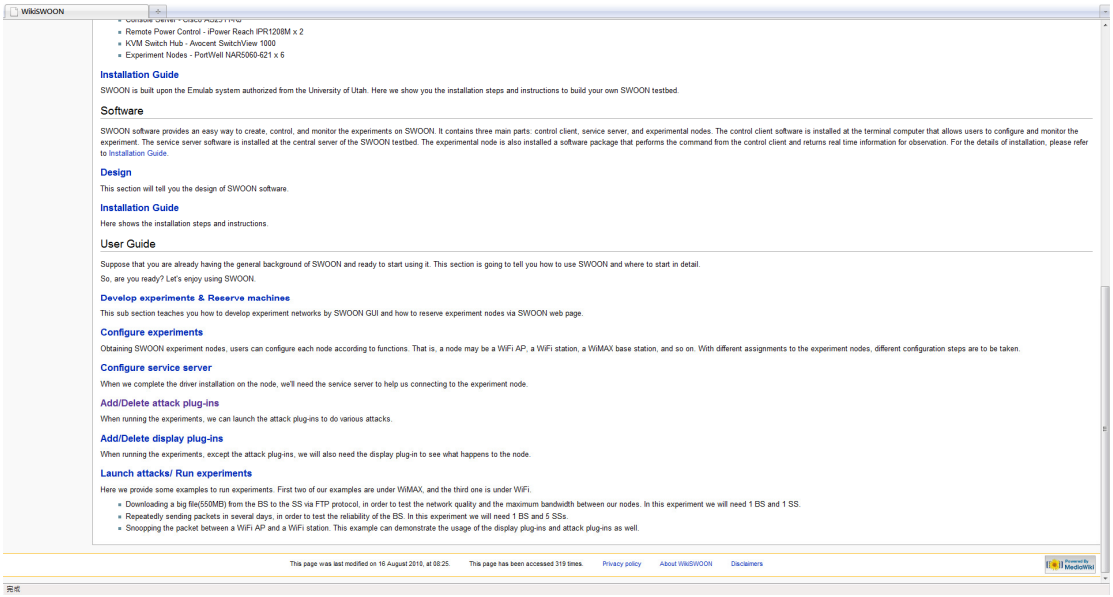
實驗操作的教學分為以下幾個部份：

- 實驗環境建置
- 設定無線網路實驗拓樸
- 設定代理伺服器 (service server)
- 加入攻防模組和顯示模組
- 使用者介面操作

本計畫之研發團隊已將上述之文件以維基百科的方式公開至本平台所架設之官方網站上(http://www.swoon.cs.nctu.edu.tw、網站畫面如圖十二與圖十三所示)，期望藉由拋磚引玉的方式吸引國內外之研究學者於本平台上開發新的安全機制和網路協定，搶得引領安全技術之先機，並將此平台推廣給更多國內產學單位、尋求更多合作機會。



圖十二：異質無線網路模擬平台的維基百科



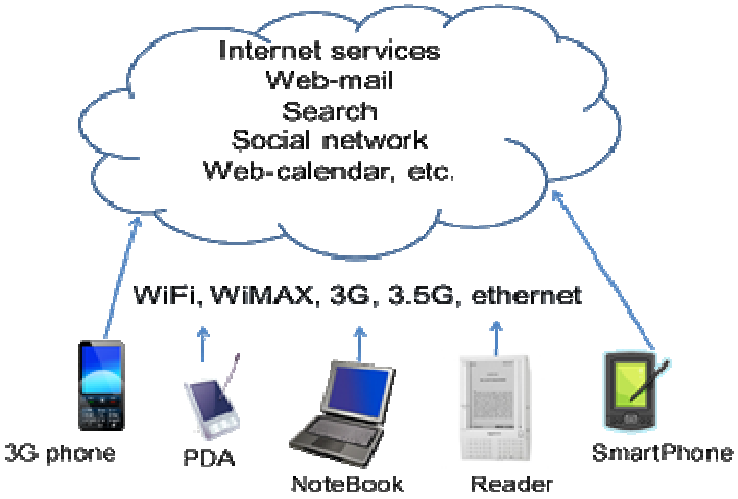
圖十三：異質無線網路模擬平台的維基百科

- **行動平台的安全管理機制**

為兼顧行動平台的網路安全防護與資料安全儲存，本計畫針對本研究子項共研發出兩套系統，分別為「安全雲端儲存系統」與「行動平台的入侵偵測系統」，以下將分別針對此兩套系統提出說明：

1. 安全雲端儲存系統

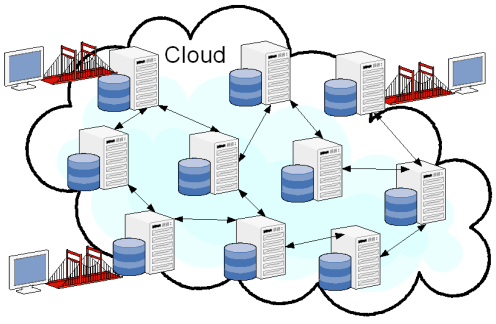
本系統之設計與理論研究內容共分為「問題背景」、「應用情景」、「主要貢獻」、「論文簡述」與「系統實做」等部分，以下將分別說明：



圖十四：行動上網裝置可透過各種網路媒介使用 Internet 上的各種服務

a. 問題背景

由於高速網路與多樣隨身上網裝置的普及化，許多服務都透過網路來傳遞(如圖十四所示)。除了有線網路的基礎建設完備與傳輸速率倍增，無線網路技術的完善與商業化造就了一個無所不在的網路存取環境。另一方面，隨身上網裝置的多樣性也相輔相成地使得社會大眾能隨時隨地上網。方便的網路環境，使得許多服務都透過網路來傳遞。較為常見的有網路信箱，搜尋引擎，網路聊天室，網路文件編輯器，等等。不僅僅是商務人士或年輕學子，一般民眾透過網路使用搜尋服務或者在咖啡店收發 email 已融入大都會的生活中。由於網路在許多文件中是由一個雲朵圖案來表示，所以這些透過網路來傳遞的服務，在近幾年被稱為雲端服務。而提供這些服務的硬體軟體與網路則整體被稱之為雲端(Cloud)。



圖十五：分散式雲端架構

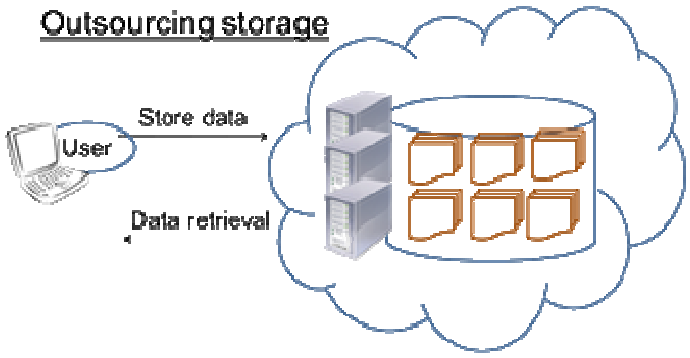
在眾多雲端服務中，我們考慮雲端儲存這項基礎服務並專注於非集中式的雲架構環境 (如圖十五所示)。一個非集中式的雲架構環境並沒有一個長駐的中央控制管理單位，這使得整體系統較為彈性且不會在中央控制管理單位造成系統效能瓶頸，任何雲端中設備的新增或刪除都不需要透過中央控制單位處理，但是相對地，沒有中央控制管理單位的存在，整體系統行為的掌握就仰賴系統對每個伺服器成員的設定與規範。

將資料儲存在雲端首先會面臨的使用者疑慮是資料是否能夠正常取回，這點在各界都已有完善的解決方案，主要是透過容錯機制來防止任何系統內部的意外錯誤，另外一個新興的使用者疑慮是資料隱私性的問題，資料存放在雲端之後是否會被惡意人士竊取再利用造成更大的社會動亂，在雲端服務四起的現在，我們非常需要一個能夠同時處理系統容錯與資料隱私性的雲端儲存系統。

在我們針對非集中式雲端儲存系統的研究中，我們提供了一個具有容錯能力與高度資料隱私性的系統，除了儲存服務之外，我們也新增了金鑰管理服務以降低使用者管理金鑰上的風險。

b. 應用情景

在非集中式雲端儲存系統中，圖十六為一個雲端儲存應用情境，當使用者將資料都儲存到雲端時，系統必須保障使用者可以再將資料取回。我們不僅考慮雲端儲存的功能性(高度容錯能力)亦強調維護使用者資料的隱私性。



圖十六：雲端儲存服務

在功能性上，我們透過 Redundant 儲存來因應雲端中伺服器可能意外地斷線或儲存設備的毀損，使得系統在發生意外狀況時仍能夠提供服務。在資料隱私性上，我們則是考慮一個高度隱私性的要求，使用者的資料不僅僅是其他系統中使用者無法接觸，負責提供服務的雲端儲存伺服器本身亦無法得知資料的內容。有此要求主要是希望能夠提供給使用者一個高度可信賴的雲端儲存系統，消除對於儲存個人資料的隱私性疑慮。

在提供了具有容錯能力與高度資料隱私性的雲端儲存系統後，最基本的可以成為一個提高資料隱私性的儲存加值服務，讓使用者自行分流需要加強隱私性的資料與其他資料，在可以應用的雲端服務上除了先前提到的 data outsourcing 之外，更可以形

成一個雲端服務的基礎平台，在平台上可以建構網路信箱，網路相簿，網路社交平台等多種雲端應用服務。

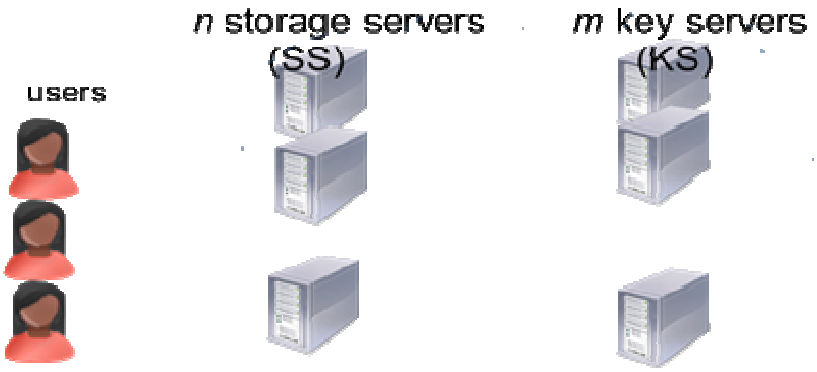
c. 主要貢獻

研究成果的主要貢獻可以從兩個角度來說明。從學術理論上來看，我們提供了一個結合了容錯技術與公開金鑰加密系統的密碼學工具，這個工具能夠在一個非集中式的儲存系統環境中被使用，使得系統同時具有資料可信賴與高度隱私性並且兼顧了分散式的優點，另外針對系統中資料儲存的取回正確率上，我們亦提供了一個完整的分析方式並建議了一組通用的系統參數。從儲存系統發展與應用上來看，我們強調了資料隱私性在雲端儲存系統上的重要性與一個強度上的分野，早期網路儲存系統的隱私性是建立在完全信任儲存伺服器的假設下，僅對登入的使用者進行身分認證，我們則是強調資料隱私性的強度應該要能夠消除對儲存伺服器的信任的假設條件。

d. 論文簡述

隨著高速網路與行動通訊的普及，雲端儲存服務已融入日常生活中，例如網路信箱，網路相簿等。使用者可以隨時隨地遠端透過行動裝置存取資料。除了可信賴的儲存機制之外，雲端儲存系統中的資料隱私性問題已日益被重視。將資料儲存在雲端系統中意味著將資料放置在第三者的環境中。如何同時保障使用者資料隱私性與儲存系統功能性是我們研究的主題。我們考慮一個沒有中央控制單位的雲端儲存系統，結合了公開金鑰加密系統與容錯編碼技術來設計一個同時具有高度隱私性與容錯能力的雲端儲存系統。我們的儲存系統保障了使用者資料的隱私性，即使是所有的儲存伺服器都被攻擊者控制，也無法破壞。系統同時具有容錯能力，當儲存系統中的儲存伺服器無預警離線或關閉，系統服務仍能正常運作。為了非集中式的系統架構，我們的公開金鑰加密系統經過特殊設計，使得編碼的程序與解密的程序可以平行地在各伺服器中運作，無須中央控制單位的協助。整體儲存系統除了基本的容錯能例外，使用者可以享有高度的資料隱私安全。

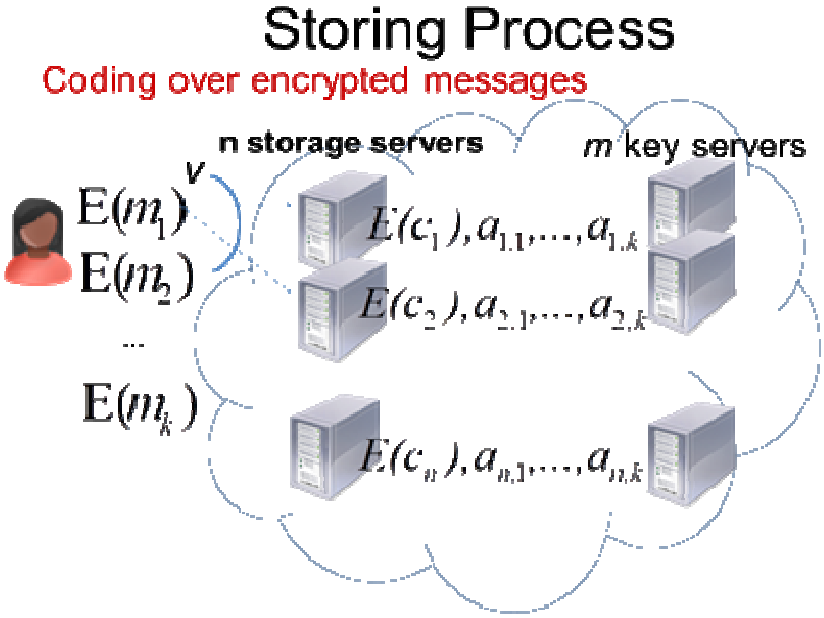
我們的系統包含了三種身分，儲存伺服器與金鑰管理伺服器形成雲端系統，使用者則是可以存取雲端系統。另外我們設定儲存伺服器的數量為 n，以及金鑰管理系統的數量為 m。系統概況請參考圖十七。



圖十七： 儲存系統架構

簡略地來說，使用者的資料將在被加密後透過隨機容錯編碼技術分散到各儲存伺服器中，各儲存伺服器儲存的是加密狀態下的一個編碼片段，爾後，使用者要取回資料的時候，透過金鑰管理伺服器的協助，可以將加密狀態下的一個編碼片段先行解密再進行解碼解析回原來的資料。在這期間，由於儲存伺服器與金鑰伺服器都是獨立進行編碼與協助解密的程序，所以不需要一個中央控制單位的協助。

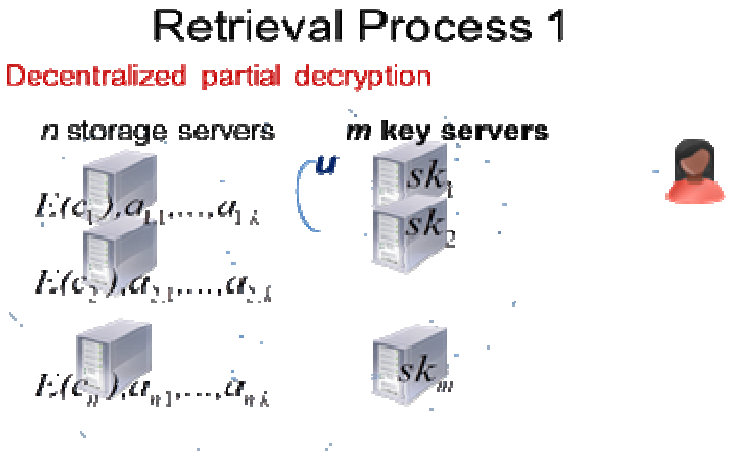
從整體系統流程來看，我們將系統分為資料儲存與資料取回兩個功能。在一開始系統架設後，使用者加入系統將會獲得一對金鑰分別是加密金鑰與解密金鑰。使用者可以選擇一組金鑰管理伺服器並將自己的解密金鑰透過秘密分享機制分散給這些金鑰管理伺服器使得其中只要有超過 t 個伺服器合作，可以將解密金鑰解回。



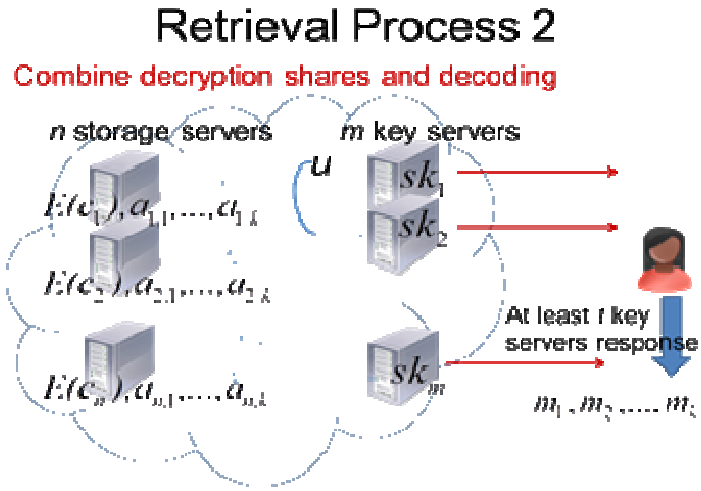
圖十八：資料儲存流程圖

資料儲存流程有兩個步驟，圖十八為儲存流程圖。第一步，使用者加密自己要儲存的 k 筆資料，並且針對每筆資料隨機地傳送給一個儲存伺服器並重覆這個隨機

散布的程序 v 次。第二步，每個儲存伺服器針對每筆收到的密文資料隨機挑選一個係數並且對所有收到的密文資料透過選定的係數進行編碼。最後在單一儲存伺服器中儲存的資料為一個編碼後的結果與選定的所有係數。資料取回流程有兩大步驟。第一步(如圖十九所示)，使用者先向金鑰儲存伺服器提出要求，每個金鑰管理伺服器再隨機向 u 台儲存伺服器詢問資料。第二步(如圖二十所示)，當儲存伺服器回傳資料給金鑰管理伺服器後，金鑰管理伺服器使用使用者所給予的部分解密金鑰進行一個部分解密程序， 並將解密結果與係數一併回傳給使用者。使用者在收集來自於超過 t 個金鑰管理伺服器的回覆之後，即可合併這些回傳值得到編碼區段，再透過解碼程序得到原始資料。



圖十九：資料取回第一步驟之流程圖



圖二十：資料取回第二步驟之流程圖

由於資料儲存與取回都使用了隨機程序，使用者在儲存資料進入系統之後，正確取回資料的事件有一個發生的機率。我們將儲存系統表示為一個隨機圖形，透過隨機圖學中既有的定理，我們分析出這個正確取回資料事件的發生機率值，並提供一個適當的 u 值與 v 值設定，以保障足夠好的資料正確取回機率。這個設定表示如下:

$$\text{For} n = ak^c, m \geq t \geq k, a > \sqrt{2}, v = bk^{c-1}\ln k,$$

$$c \geq 1.5, u = 2, b > 5a, \Pr[\text{Failure}] \leq \text{k/p} + \text{o}(1)$$

其中 Pr[Failure]是指資料不能正確取回的事件發生的機率，而 p 值則是我們使用的容錯編碼運算所操作的群(group)大小。

在容錯能力上來說，我們的系統能夠容忍(n-k)個儲存伺服器錯誤與(m-t)個金鑰管理伺服器錯誤。只要有 k 個儲存伺服器與 t 個金鑰管理伺服器仍正常運作，則使用者可以有很高的機率將資料取回。

在資料隱私性方面，因為資料都是以加密的型態被儲存，所以即使是所有的儲存伺服器都被攻擊者控制，資料內容仍能保密。我們對於金鑰管理伺服器則有較高的信任要求，我們假設這些金鑰管理伺服器有較好的安全機制以保障使用者的各個部分解密金鑰。

e. 系統實做

我們的行動式智慧裝置系統使用的是 android，主要是它因為採用開放原始碼的策略，讓軟體開發者能不受限制的開發符合使用者需求的軟體。在加密系統方面，我們採用 public key system，並加上 Erasure Code 來確保我們的資料完整性，至於加密的部分則是採用 pairing 來確保資料的保密性。此系統能讓儲存的成本降低，且提高資料的可靠度及可信度。以下將分別就系統架構、使用者功能、伺服器管理、與軟體開發等部分加以說明：

系統架構

我們實作了 Hsiao-Ying Lin, Wen-Guey Tzeng, "A Secure Decentralized Erasure Code for Distributed Network Storage," *IEEE Transaction on Parallel and Distributed Systems,* 2010. 論文中的方法，我們的系統建置主要分為三大部份:

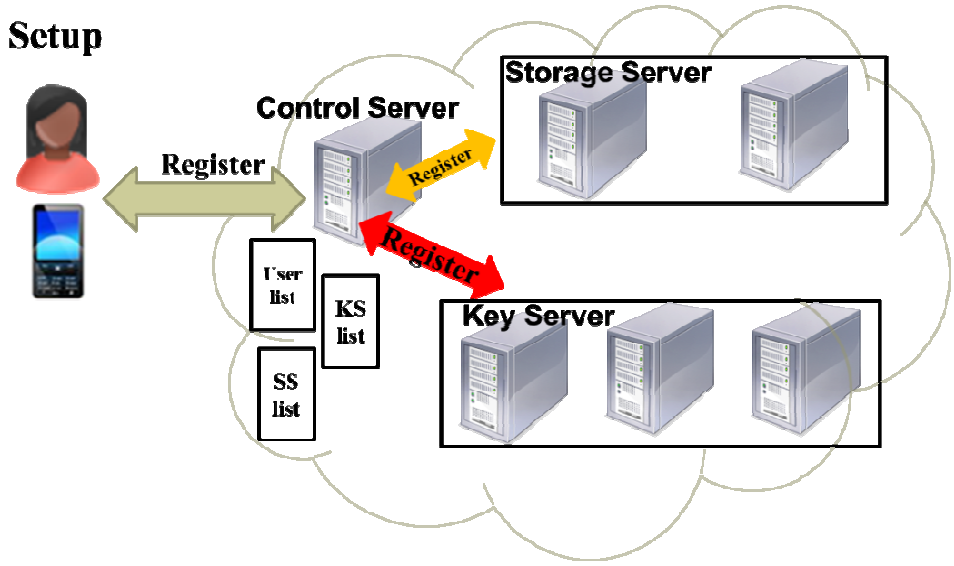
(1) 使用者(client): Android 智慧型手機

(2) 控制端伺服器(control server): 負責紀錄人員及伺服器數量並代替智慧手機傳送給儲存伺服器(storage server)

(3) 服務型伺服器: 包含儲存伺服器及金鑰伺服器(key server)

系統 Setup 如圖二十一所示。首先 Android 手機會產生一對 key pair 跟一個 generator 以及 random 取的質數 p，跟一個有效的 pairing 運算，緊接著將 secret key 利用 threshold 來產生多把不同的 secret key，然後將這些 secret key 依照我們編號陸續的不同存放在相對應的金鑰伺服器上。另外在控制端我們要先啟動我們的控制端伺服器，讓其他儲存伺服器啟動後可以來控制端伺服器這登記，最主要的目的是有效的掌握儲存伺服器的數量。而由於使用者人數眾多，所以必須管理每一位使用者的權限，讓使用者不能任意的下載及修改他人的檔案及資料，每位使用者都必預先跟 database 註冊一個帳號及密碼，註冊成功後，登入個人帳號及密碼，系統會依據帳號的權限而給予不同

檔案及資料，我們可以利用 web browser 來查看目前有哪些檔案已經在遠端的儲存伺服器上並可知道有幾台儲存伺服器已經提供服務。



圖二十一：安全分散式儲存系統架構圖

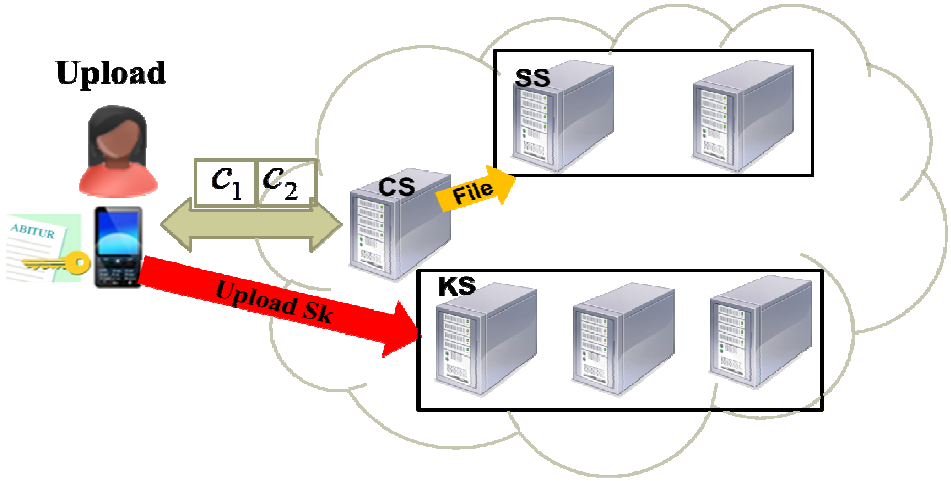
使用者功能:

使用者可用的功能分別為登入、登出、上傳和下載這四個功能。

(a) 登入：在登入之前，使用者已經在 database 裡註冊了一個帳號，所以我們只頇在網頁上輸入我們的帳號跟密碼，經過系統確認後，網頁上只會顯示之前上傳的資料，不會走任何情況而使得使用者可以存取到別人的檔案。

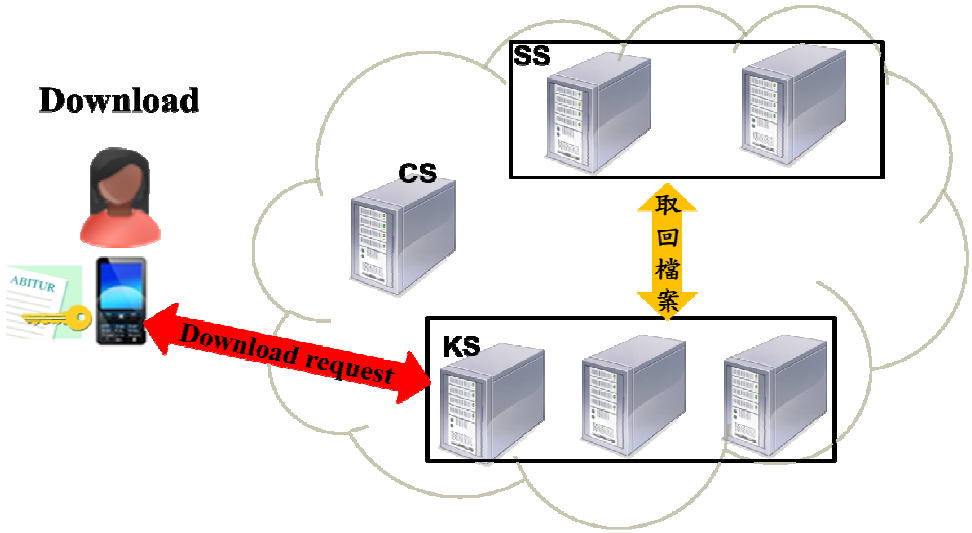
(b) 登出：使用者想要離開此系統，則會由手機發出一個離開的訊息，緊接控制端伺服器會將目前線上清單裡有關於此使用者資訊一一消除。

(c) 上傳：在上傳之前 Android 手機會將檔名透過一個 hash function 來產生一個 generator，緊接著利用我們所產生的 generator 將資料以 pairing 來加密，然後把之前產生的 public key 跟我們加密後的資料一併上傳給控制端伺服器，另外使用者在上傳時為了避免智慧型手機網路無法負荷上傳，所以我們只需要把加密過後的資料，上傳給控制端伺服器，由控制端伺服器幫忙將一份資料傳給數個儲存伺服器。(如圖二十二所示)



圖二十二：使用者上傳資料示意圖

(d) 下載: 在下載之前，Android 手機會先跟金鑰伺服器發出下載的 query 訊息，當金鑰伺服器收到訊息後，就會跟控制端伺服器要求 storage server list，等到拿到 storage server list 後金鑰伺服器會隨機選取數個儲存伺服器來下載，透過一個金鑰伺服器我們可以得到一個方程式，所以需要數個金鑰伺服器的資訊來做解連立方程式，將加密的資料解密回來。(如圖二十三所示)



圖二十三：使用者下載資料示意圖

<u>伺服器管理:</u>

控制端伺服器需要管理的部分，分別為使用者帳號、儲存伺服器列表和金鑰伺服器列表。

(a) 使用者帳號管理: 控制端伺服器會連接到一個 database，裡面包含了使用者的帳號密碼以及檔案的權限，所以每位普通的使用者在瀏覽檔案時，都只會看到自己的檔案。

(b) 儲存伺服器列表管理: 當一個儲存伺服器架起來的時候，會發出一個訊息給控制端伺服器以進行註冊，然後我們在瀏覽網頁時可以看到現在有提供服務的儲存伺服器，裡面內容包含 Name、IP 以及 Port，

能讓使用者一目了然的知道哪些伺服器是有提供服務的。當 storage 關閉時會發出一個關閉的訊號，使得控制端伺服器能從 storage server list 裡做修改，萬一儲存伺服器因為某些不確定的因素而導致不正常關閉，則當控制端伺服器在送訊時給儲存伺服器時，會發現對方無回應，則我們控制端伺服器也會將沒回應的儲存伺服器從 storage server list 中清除。

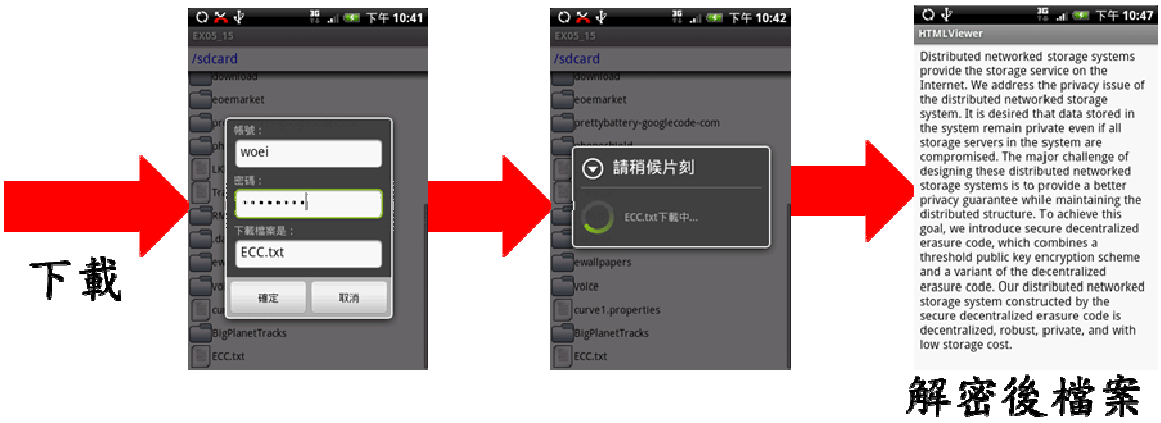
(c) 金鑰伺服器列表管理: 當一個金鑰伺服器啟動完成的時候，會發出一個訊息給控制端伺服器註冊登記，然後我們在瀏覽網頁時除了看得到 name、port、IP 外，還提供一項 information 讓使用者能知道哪些伺服器是可被信任的。

軟體開發:

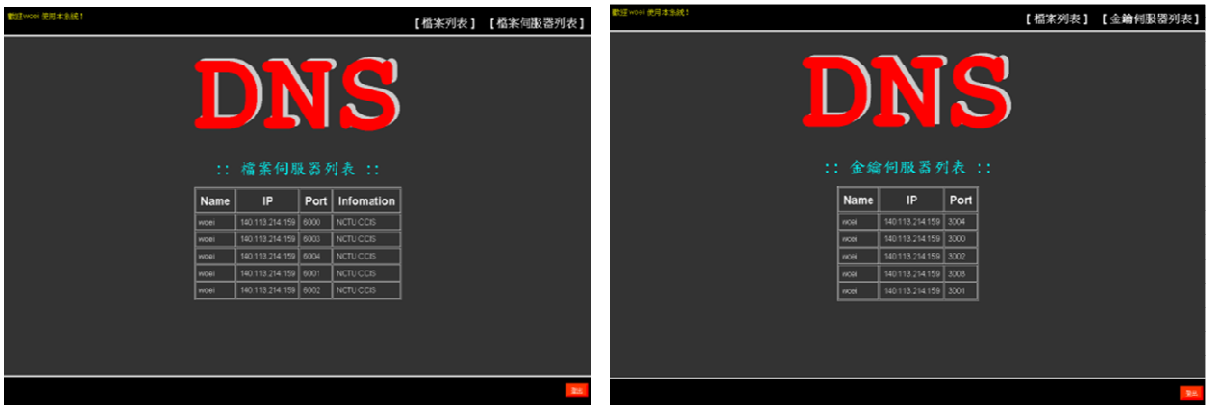
在系統實作上，首先需要大量的伺服器，所以我們先將大量的伺服器都實作在同一台機器上面，依據port的不同區分不同的伺服器，另外我們會依據不同資料夾來區分不同儲存伺服器的儲存空間，所以即使在同一個機器上，我們仍然有區分出不同的伺服器。在開發環境上，我們所使用的機器為core i7，硬碟容量為500GB，Operating system為Window vista 的桌上型電腦，日後會將機器逐步的分散出去，使用者端我們使用的手機為HTC Magic， android SDK的版本為1.5，另外我們所使用的開發環境為java的JDK 1.6，開發軟體為eclipse 3.4.0，並使用套件JPBC 1.0.0。圖二十四、圖二十五、圖二十六以及圖二十七為我們所實作出的系統執行時的相關畫面。

圖二十四：安全雲端儲存系統使用者介面 – 資料原始檔以及功能清單

圖二十五：安全雲端儲存系統使用者介面 – 使用者上傳資料



圖二十六：安全雲端儲存系統使用者介面 – 使用者下載資料



檔案伺服器列表　　金鑰伺服器列表

圖二十七：安全雲端儲存系統使用者介面 – 伺服器管理介面

2. 行動平台的入侵偵測系統

本系統之研究內容共分為「系統設計」與「系統實做」兩部分，以下將分別針對這兩部分進行說明：

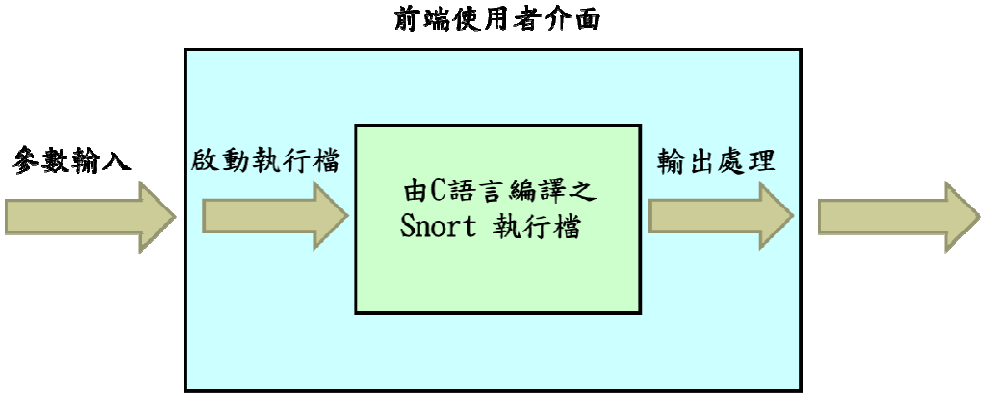
系統設計

在Android系統上，目前限定所有應用程式都必須以JAVA語言來撰寫，再以Dalvik Virtual Machine 來轉換成DX bytecode，因此程式開發和JAVA ME類似，應用程式的介面部分則是由XML程式來設計。簡單來說，即是系統由LINUX來執行，應用程式則透過JAVA語言來開發以及執行。要在Android上執行以C語言所構成的程式，有兩種方法：

(1) 利用Android NDK，在JAVA應用程式中去呼叫以C語言所構成的Library以執行我們要的功能。

(2) 將C語言寫成的程式直接編譯成執行檔，然後再從JAVA應用程式中，去直接執行該執行檔。

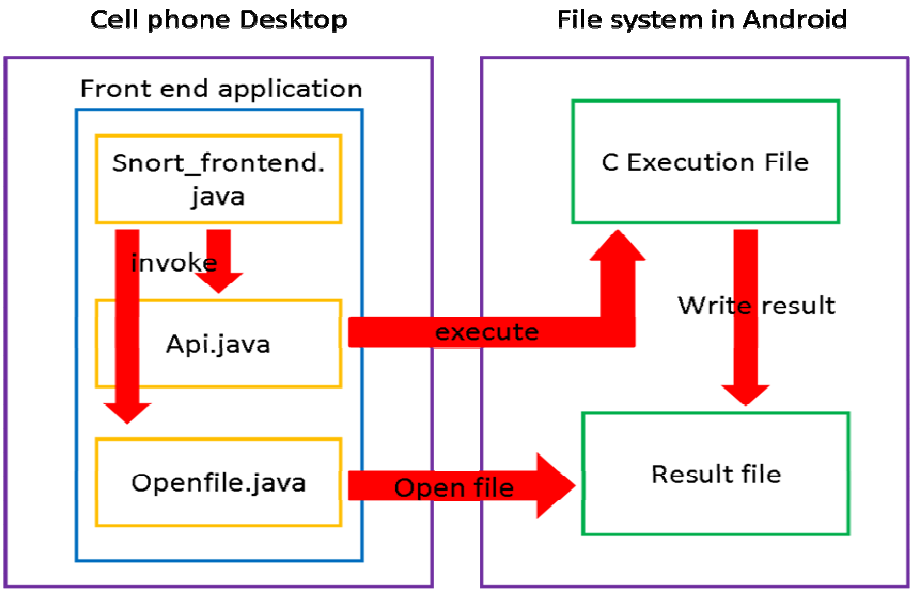
這裡我們移植 Snort 的方法，即是利用方法二，以 Cross-Compiler 將 Snort 編譯成可以在 Android 上執行的版本，並將其放進 Android file system 中的/data 資料夾中，再利用前端的 JAVA 應用程式介面，讓我們直接從手機中去執行我們的 Snort 執行檔。這個方法的好處是程式的功能部分可以像以前在 Linux 上開發一樣，在 JAVA 應用程式上只要改寫如何去設置程式執行參數即可(如圖二十八所示)。



圖二十八： 入侵偵測系統架構圖

在這裡Snort為一個statically compiled的執行檔，所以將它放入Android的file system中時，便是一個可以獨立執行之指令，但是我們並無法直接在手機螢幕上存取file system中的執行檔，所以我們需要一個前端控制系統，利用這個前端控制系統去執行放在/data中的Snort執行檔，並且將回傳的結果存起來，再利用前端控制程式去存取傳回之結果以顯示在手機螢幕上。前端控制程式的組成，如圖二十九所示，主要的source file共有以下三個。

- snort_frontend.java：前端程式的主要進入畫面，定義各元件以及執行動作
- Api.java：包含Snort執行所需的函數
- openfile.java：提供記錄檔(Log file)相關功能。



圖二十九：入侵偵測系統運作流程

系統實作

整體系統移植的步驟上，大略可以分成三個步驟: 先取得Android上的Root權限，將Snort編譯成可以在Android上執行的執行檔以及實作前端控制介面並連結到Snort執行檔。

(一) 實作環境建置

(1) Rooting Android

Rooting的方法不只一種，這裡我們採用比較簡單的方法，即是利用去覆寫別人已經Rooting好的image，以達到Android手機Root許可權解鎖之功能。這裡Rooting的動作也可以在Windows上去執行，並沒有特別規定要在哪個平台上。在這裡選擇了CyanogenMod所製作的image，因為他有Root的許可權，且image裡面也沒有加入太多的應用程式，以避免我們系統記憶體上可能會有不足的情形發生，此外也可以讓Android的file system的data資料夾解鎖，以便進行資料夾內資料的讀取。並且選用HTC Magic手機作為我們的硬體平台，因為目前現行HTC手機Rooting的資訊，以此種手機最多，且也有網路上的使用者寫好的備份軟體Nandroid可供使用。下面的表六將我們所使用的軟硬體版本做了一個簡單的整理。

作業系統	Ubuntu 8.10
手機硬體	HTC Magic 32A
手機作業系統	CyanogenMod
Cross-Compiler	Sourcery G++ Lite
手機系統備份軟體	Nandroid

表六：IDS軟硬體使用版本整理

首先進行備份的動作，為避免覆寫失敗，而讓機體變成完全無法使用之情況：

I. 先把手機用fastboot模式開機(同時按著電源以及返回鍵開機，然後就會看到三個機器人在玩滑板)。fastboot執行檔可以在網路上下載，但是如果找不到fastboot執行檔的話，可以自行下載Android source code，以進行編譯，編譯完成之後就會有fastboot執行檔可以使用。(如果電腦上已經編譯過andorid source code則可以在[source code directory]/out/host/[安裝平臺]/bin裡面找到fastboot檔案)。

II. 在電腦上打入下列指令以便讓手機讀取電腦中的image以進行開機。要先將當前資料夾移至放置fastboot指令之資料夾，且也要把還原備份用的image放到同一資料夾內。

```
$ ./fastboot boot
recovery-RA-magic-v1.2.3H.img
```

這裡recovery-RA-XXX.img的版本沒有太大的問題，只要可以備份就行了。

III. 開啟以後可以看到一個命令選單，選擇Nandorid v2.x backup即可備份系統，備份完可以到sdcard底下去看Nandroid資料夾裡面是否有儲存系統資料，以檢查是否正確備份完成。在之後進行Rooting動作若覺得Rooting失敗，即可利用一樣的方法進入recovery image，選取restore恢復至原本內建的Android OS即可。

再來就要開始進行Rooting，原本要進行Rooting需要相當繁瑣的指令，但是現在已經可以利用Recovery image去進行覆寫了，將前面有提到的CyanogenMod下載下來解壓縮，可以看到三個image，其中的HTC_ADP_1.6_DRC83_rooted_base.zip以及update-cm-4.2.4-signed.zip即是我們要進行複寫的image，利用fastboot進入Recovery image之後，選擇覆寫的選項，依序將HTC_ADP_1.6_DRC83_rooted_base.zip以及update-cm-4.2.4-signed.zip和bc-4.xxxx.zip覆寫，之後約等待10分鐘，即可完成Rooting之動作。如果在Rooting詳細步驟上有所疑問可以參考CyanogenMod[11]的網站中有詳細的指引。

(2) 安裝 Cross-compiler

這裡我們將Cross-Compiler安裝在Ubuntu 8.10上，並安裝CodeSourcery所開發的要Sourcery G++ Lite做為我們的Cross-Compiler，安裝檔可由[4]的網址取得。下載完畢之後點擊，以將其解壓縮並且安裝，建議將Cross-compiler所安裝的資料夾設為指令預設搜尋路徑，這樣在之後進行Cross-compilation時會方便許多。

(二) 交叉編譯 Snort 至 Android 上

這裡為避免出現未知問題，首先我們利用編譯BusyBox以測詴Cross-Compiler是否可以正常編譯出可在Android上執行之程式，再來利用編譯Tcpdump檢查是否有正確編譯Libpcap，最後則是進行Snort的Cross-Compilation工作。利用此種漸進式的編譯方法，在出現執行檔無法正常執行的問題時，即可判斷是哪個步驟上出問題，以快速找出問題之解決方法，並可判斷出Library與執行檔相容性之問題。所以此種方法不只可以用來編譯Snort至Android上，也可編譯架構類似的程式至Android上執行。

經過我們的測詴，要想編譯出可在Android上執行的Snort，可以使用最新的library版本去編譯，使用舊版的library也是可以，但是最新版本的Library，在對未來要將Snort的版本提升上，相容性會比舊版的好。

為了避免混淆，以下以$符號開頭之字串皆為在電腦上輸入之指令，以#符號開頭之字串皆為在手機上輸入之指令。

首先我們先進行BusyBox的編譯，已確定Cros-Compiler是否可以正常執行工作，首先將路徑移至BusyBox資料夾後輸入

```
$make
menuconfig
```

然後BusyBox會自動進入畫面編譯選單，先進入BusyBox setting，再進入Build Options，然後在Cross Compiler Prefix的地方，輸入之前安裝Cross Compiler的路徑，輸入的路徑大概如下所示

```
…/Sourcery_G++_Lite/bin/arm-none-linux-gnueabi-
```

輸入完之後儲存跳出，然後輸入make以及make install之指令即可測詴BusyBox是否可以正常使用，在編譯的機器上不能執行是正常的情況，然後將BusyBox以下列指令放入手機內進行測詴。以下由adb開頭之指令皆由Android SDK所提供，如果電腦裡面沒有這些指令可供使用，請先安裝Android SDK。

```
#adb push
BusyBox/data
```

```
#adb shell
```

```
#cd /data
```

再來就要進入我們的主要工作，編譯Snort這裡我們使用libpcap1.1以及libpcre 8.00之libraries，先compile成static library，即.a結尾之檔案。以下以/TARGET表示編譯完成之目標資料夾，可以改成任意一個放置編譯結果檔案的資料夾。

檔案下載完成之後解壓縮，然後進入資料夾裡面修改Makefile設定，要修改的重要參數如下所述：

CC：指定.c檔案的compiler
host：指定編譯出來的執行檔要執行的平台
prefix：編譯出來的執行檔要放在本機端的哪個資料夾
LDFLAGS：linking的選項，這邊主要用來指定是要statically linking還是dynamically linking
CXX：指定.cpp檔案的compiler

再來我們要開始編譯Libpcap，一開始沒有Makefile之存在時，需執行./configure指令以生成Makefile。

```
$CC=arm-none-linux-gnueabi-gcc ./configure
--host=arm-linux --prefix=/TARGET
 LDFLAGS="-static"
CXX=arm-none-linux-gnueabi-g++
```

```
$make
```

```
$make install
```

最後則是編譯Snort，這裡我們對Snort1.7版本進行compile，將其compile成static linking之執行檔。之所以要使用Snort1.7的版本，是因為1.8.6之後的版本即使給定LDFLAG=”-static”仍然會compile成dynamically linking之執行檔，此為1.8.6之後的Makefile都會有的問題，dynamically linking之執行檔放到手機中會產生./snort not found的執行結果。這裡若編譯出來之結果仍然無法執行，請檢查Makefile中LDFLAG是否有正確改成”-static”然後再進行compile，因為有時候會有在configure的時候給定LDFLAG但是Makefile中仍然沒有指定成功的情形，如果有發生這種情況，請自行打開Makefile進行參數修改。

由於是編譯Snort1.7的版本，所以可能會發生在compile過程中，出現缺少bpf.h不存在的情形，這時候要利用

```
$ln –sf
/usr/local/include/pcap-bpf.h
 /usr/local/include/net/bpf.h
```

之指令進行連結以解決此問題。此為標頭檔更名造成之問題。所以我們這裡建立一個連結讓電腦在編譯時可以找到他需要的標頭檔。然後輸入以下指令以進行Configure

```
$CC=arm-none-linux-gnueabi-gcc
AR=arm-none-linux-gnueabi-ar
RANLIB=arm-none-linux-gnueabi-ranlib ./config
ure --host=arm-linux --disable-shared
--prefix=/TARGET LDFLAGS="-static"
CXX=arm-none-linux-gnueabi-g++
--with-libpcap-libraries=/TARGET /lib/
  --with-libpcre-libraries=/TARGET /lib/
```

除了之前給定的參數之外，還要另外給定libpcap以及libpcre的位置好讓Snort在進行linking時知道要去哪裡找libraries。這裡的目的資料夾即為上面libpcap和libpcre產生的libraries的目錄。

最後生成的Snort執行檔可以利用Linux上的file指令去檢查其是否為statically compiled。

在確認了執行檔有正確編譯之後，將執行檔放入Android裡面的/data/資料夾底下，即可執行。

```
#adb push
Snort/data
#adb shell
#cd /data
#./snort –v
```

即可看到執行結果，這個指令將會執行Snort的Sniffing mode，如果出現圖三十之結果即代表成功。 另外編譯出來之Snort執行檔大小約為1.7MB，這個大小對於我們來說是可以接受的，畢竟它包含了兩個Library以及相當豐富的功能。

圖三十：成功編譯後之執行結果

(三) 前端控制程式設計

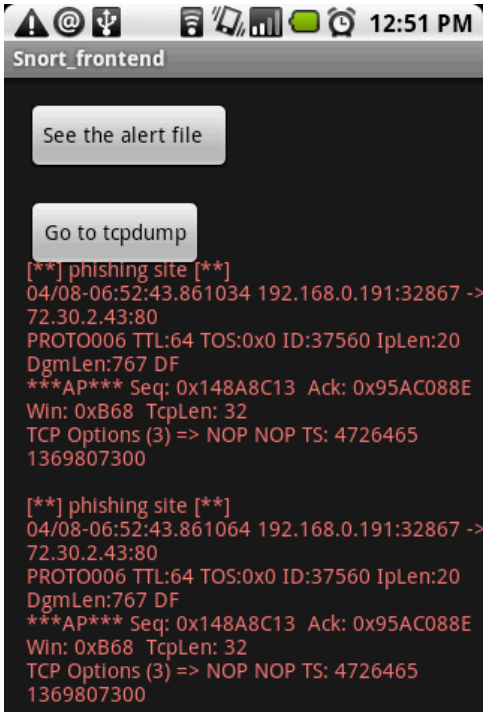
在這裡Snort為一個statically compiled的執行檔，所以將它放入Android的file system中時，便是一個可以獨立執行之指令，但是我們並無法直接在手機螢幕上存取file system中的執行檔，所以我們需要一個前端控制系統，利用這個前端控制系統去執行放在/data中的Snort執行檔，並且將回傳的結果存起來，再利用前端控制程式去存取傳回之結果以顯示在手機螢幕上，前端程式的大小目前約28KB，預計之後即使再進行功能上的修改，大小也不會增加太多。接下來介紹前端控制程式的組成，主要的source file共有以下三個。

- snort_frontend.java：前端程式的主要進入畫面，定義了主畫面的元件以及執行動作。
- Api.java：一些會用到的函數，包含我們用來呼叫Snort執行檔的函數。
- openfile.java：snort_frontend.java會呼叫此java檔以執行開啟記錄檔(Log file)之功能。

(四) 功能測試與應用

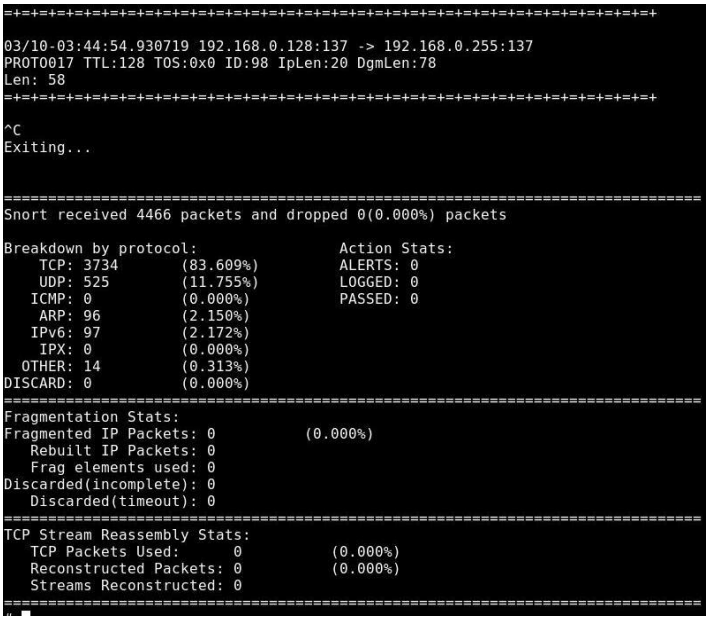
在將Snort移植到Android之後，我們需要測試一下Snort執行檔是否可以正確執行我們所要的功能。經過測試，由前端程式去執行Snort並且讀取Rule set已可正常執行。接著我們利用一個簡單的signature去進行測試，alert tcp any any -> any any (msg:"phishing site";content:"HTTP1.1";nocase;)，這個簡單的signature可以用來檢測該網站是否為釣魚網站，只要修改content內容為該網站的封包內常出現之字串即可。設定完signature之後我們就利用手機去瀏覽網頁，接著我們可以看到如圖三十

一的檢測結果。在記憶體的使用上，Snort前端控制程式的記憶體耗用量和一般JAVA應用程式沒有太大的差異，只有在/data/snort 也就是利用root去執行Snort的部分會多耗用約3352KB的記憶體，預計在之後會再進行更加詳細的記憶體耗用和signature數量關係之測試，這裡我們只有先觀察以上述之單一signature進行檢測時之記憶體耗用量。



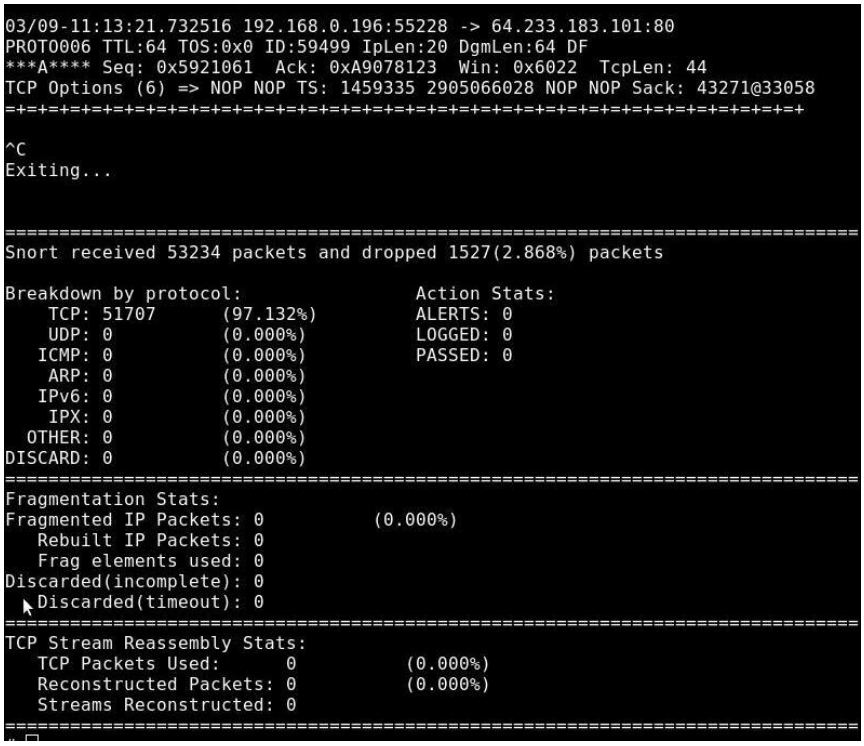
圖三十一：封包檢查結果圖

再來就是在我們進行多次的封包捕捉動作時，即使在不特別指定封包類別的情況下，且使用上網或者下載應用程式的功能時，封包也不會有漏掉的情況發生，實驗結果如圖三十二所示。



圖三十二：snort 擷取封包結果圖

除了在比較吃重的讀取影片進行即時資料串流時，會有少量的封包漏失的情形，以下實驗我們在觀看網路影片即時串流時擷取封包，我們在六分鐘內總共擷取了53234個TCP封包，實驗結果顯示我們僅漏掉了約3%左右的封包，結果如圖三十三所示。但是在一般用途上的封包擷取功能是可以被保證的。除此之外，Snort還可以檢查是否有接收到被切割開來的過小封包，並且在收集之後進行重組的動作。



圖三十三：snort擷取封包結果圖

最後是進行Snort電量測試的部分，由於手機是使用電池做為電力來源，所以電量對手機來講是非常珍貴的，我們將進行兩部分的測試，一部分是在接收封包的時候，Snort的執行會多消耗多少電量的測試；以及另外一部分，在待機時，只執行固定背景程式，以及多加入Snort背景執行，電源消耗上的差異。在消耗上的依據，我們使用電池監控程式，以百分比來記錄我們消耗的電量。

	時間	電量消耗
無 Snort	1 小時	2.8%
有 Snort	1 小時	4.6%
使用 Snort 作為背景監控程式 1 小時會多花 2%左右的電量。		

表七：待機時有無使用snort的電源消耗比較

	時間	電量消耗
無 Snort	20 分鐘	2.4%
有 Snort	20 分鐘	3.2%
在讀取影片進行資料串流的工作持續 20 分鐘，多消耗的電量不到 1%。		

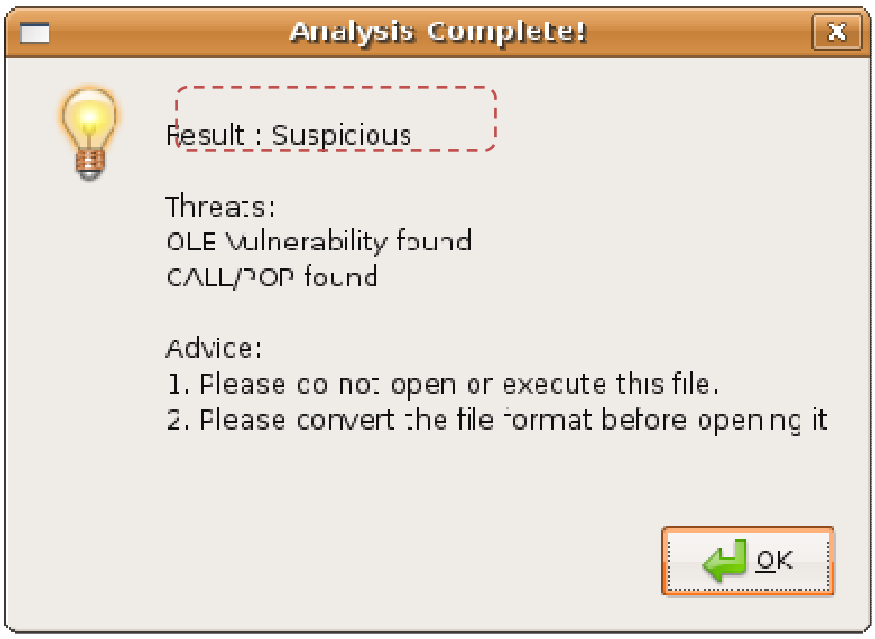
表八：上網時有無使用snort的電源消耗比較

在待機時的測詴由於擔心電量消耗太少，所以將實驗時間延長至1小時，另外上網時的動作為找數部影片來進行播放，以增加電量消耗的明顯程度，經過五次測詴後的結果平均即如上表中所示。從實驗結果可知，在持續播放影片20分鐘並且開啟Snort進行監控的情況下，多消耗的電量也不到1%，可以知道Snort並不會對電源消耗帶來太大的負擔。

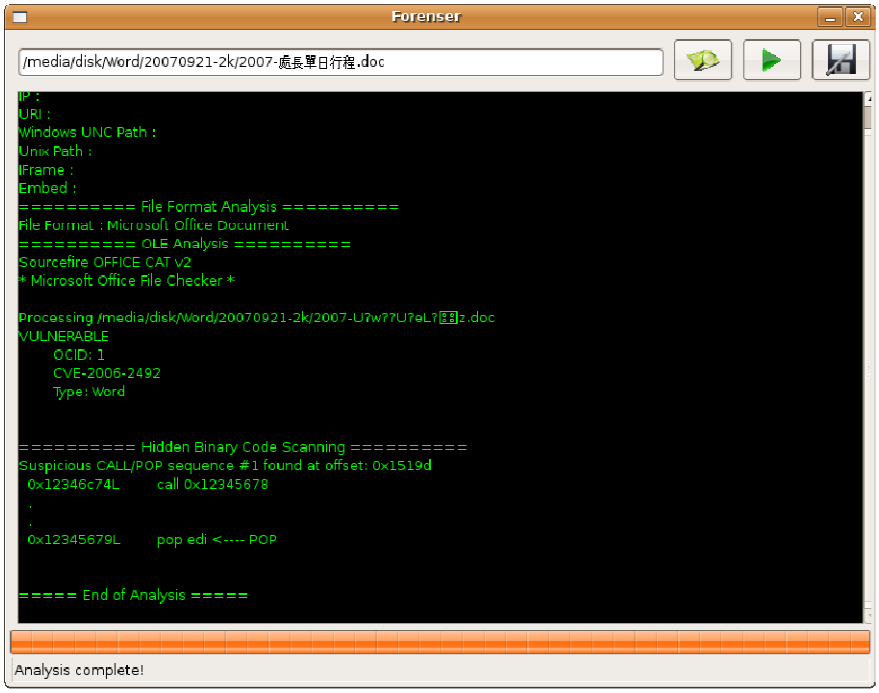
- **惡意檔案文件分析系統**

本項目開發了一個檔案檢測系統，並且有圖形化介面供使用者方便檢視報告。接下來將列出八種不同的系統檢測標的執行畫面，並簡單說明之：

Case 1： 偵測一個微軟 office 的文件檔案，其中有包含一個 OLE 漏洞，並且內嵌有動態取得記憶體位置之 shellcode，其檢測畫面與報告。

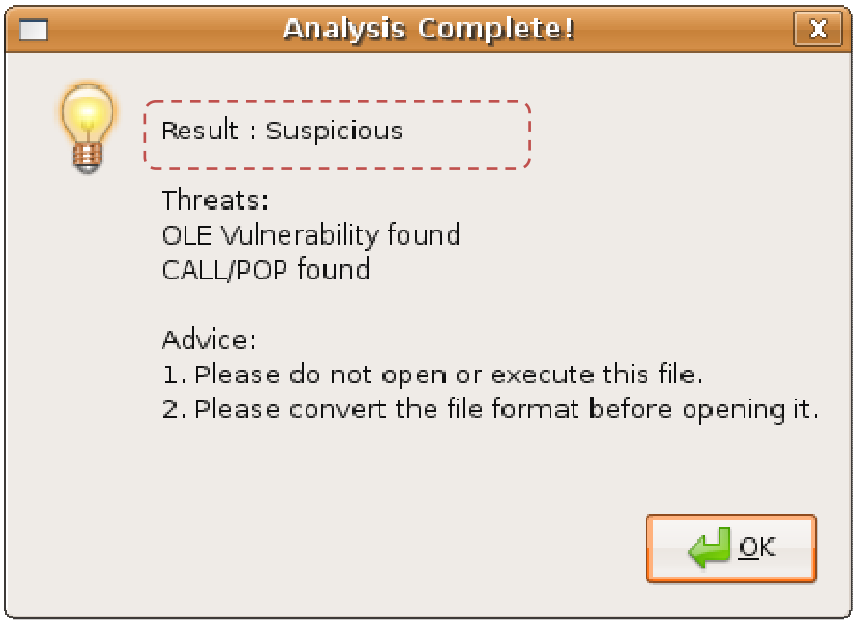


圖三十四：含有 OLE 漏洞以及 call/pop 的 doc 檔案之結果

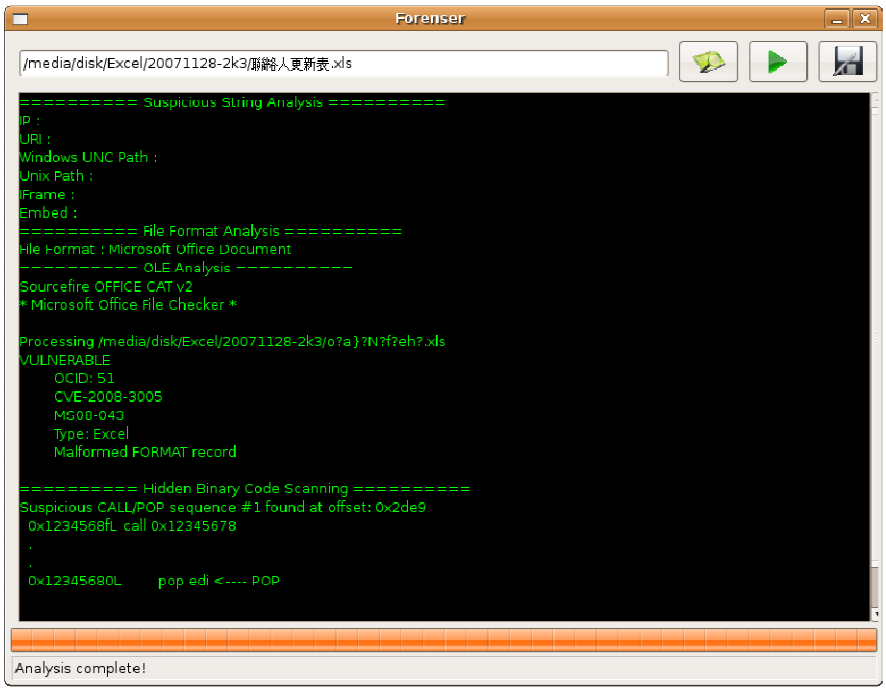


圖三十五：含有 OLE 漏洞以及 call/pop 的 doc 檔案之報告

Case 2： 輸入一個含有 OLE 漏洞以及內嵌動態取得記憶體位址之 shellcode 的微軟 xls 檔案，其結果為可疑檔案。並且建議不要開啟。

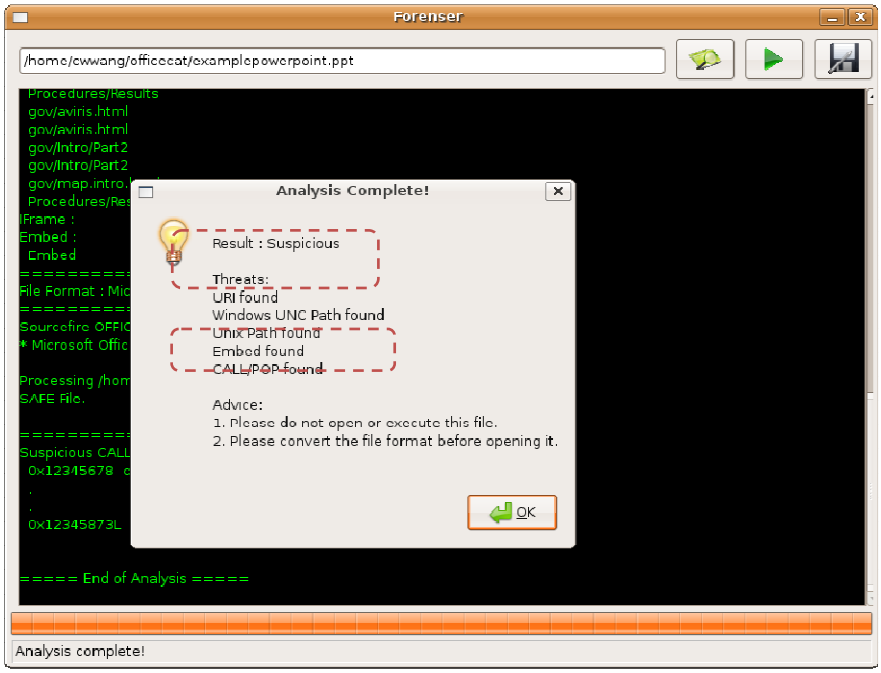


圖三十六：輸入一個含有 OLE 漏洞以及 call/pop 的 xls 檔案之結果

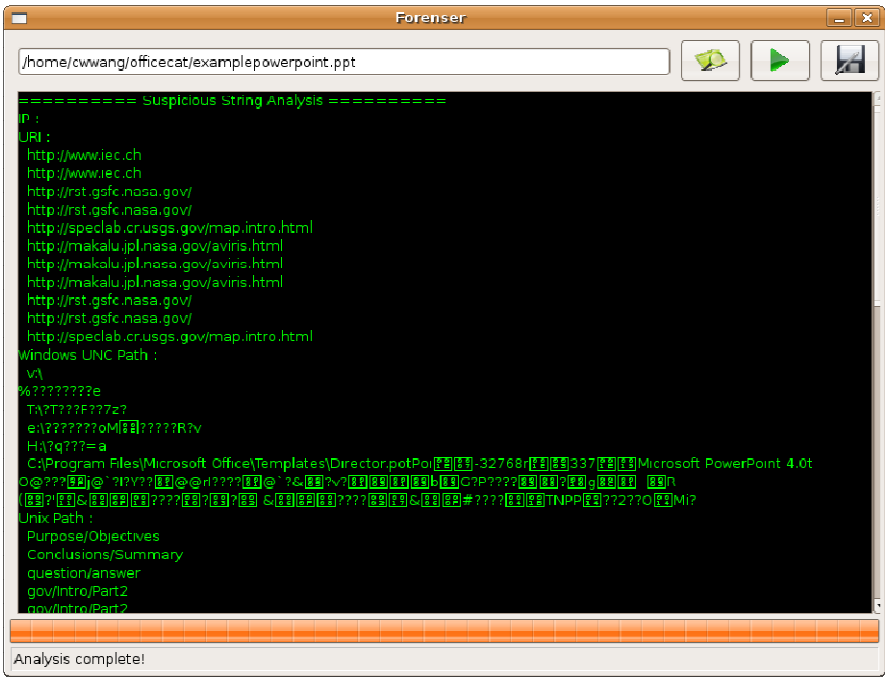


圖三十七：輸入一個含有 OLE 漏洞以及 call/pop 的 xls 檔案之報告

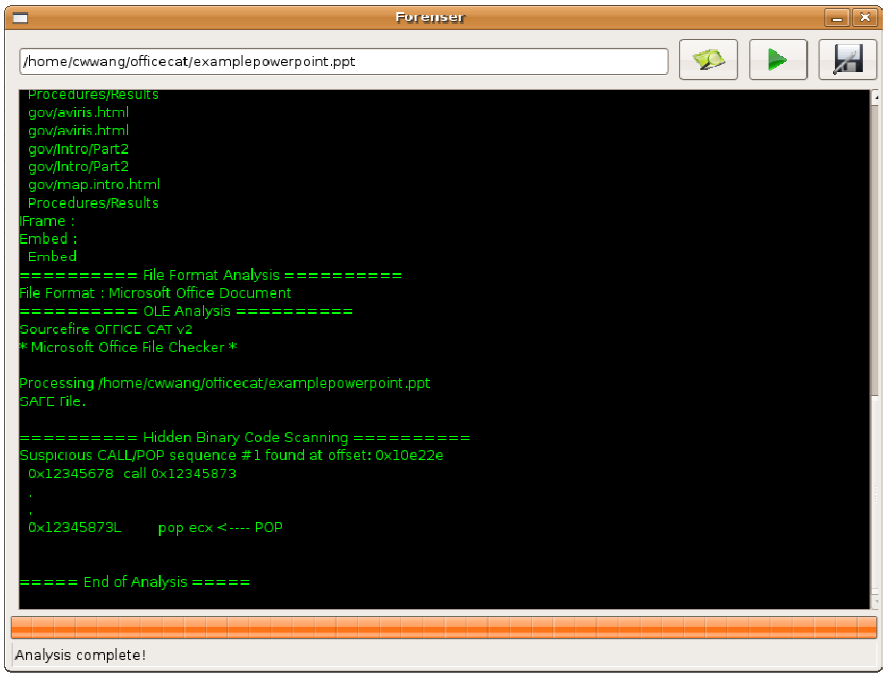
Case 3： 輸入一個含有動態取得記憶體位置之 shellcode 以及包含網路連結的微軟 ppt 檔案，該檔案將會有對外作連線的可疑性。建議使用者不要開啟該檔案。



圖三十八：含有 call/pop 以及可疑字串的 ppt 檔案之結果



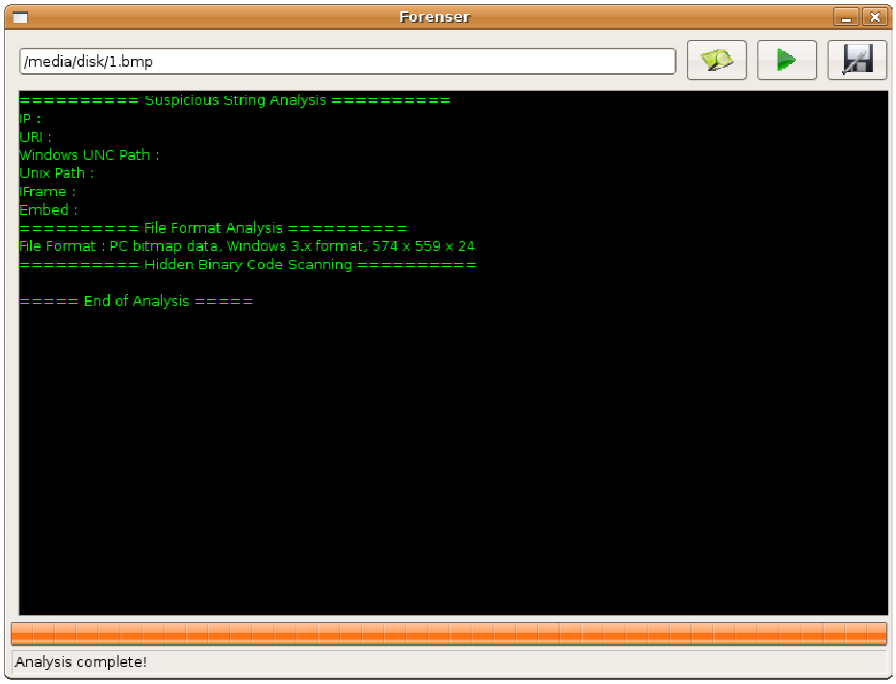
圖三十九：含有 call/pop 以及可疑字串的 ppt 檔案之報告



圖四十：檢測結果輸出畫面

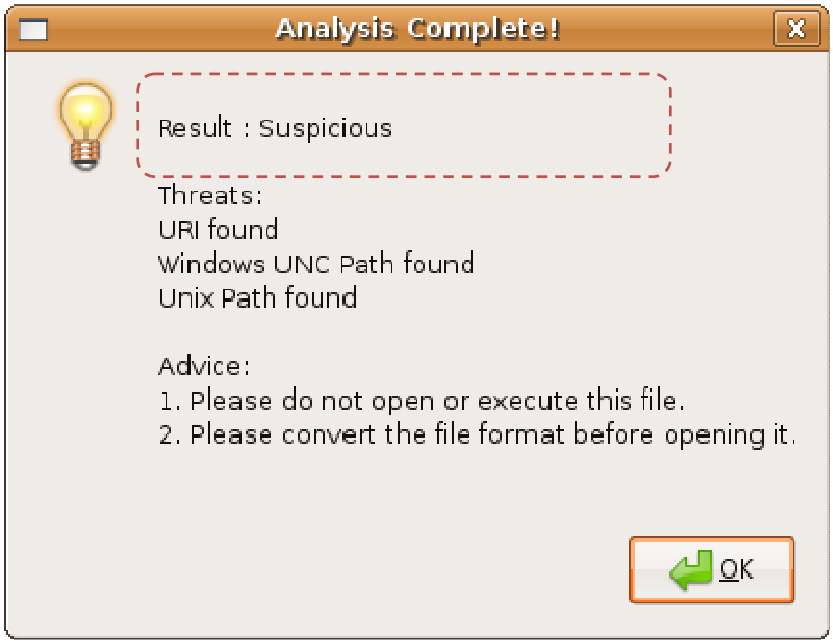
Case 4： 輸入一個正常的圖形檔案，檢測結果是安全。

圖四十一：輸入安全的 bmp 檔案之結果

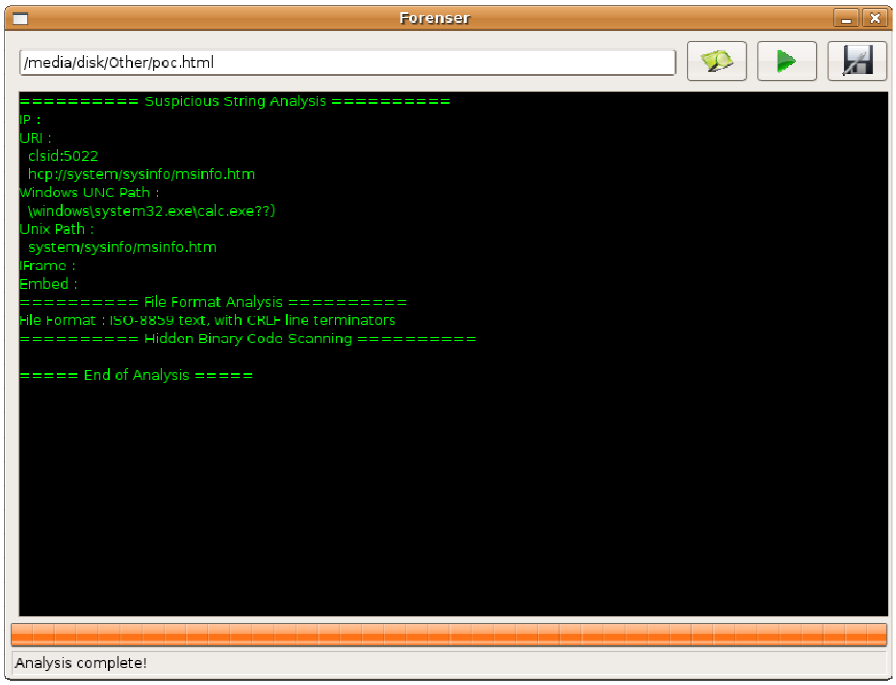


圖四十二：輸入安全的 bmp 檔案之報告

Case 5： 輸入一個包含系統檔案路徑以及網路連結之檔案，本系統將會提醒使用者該檔案具有這些可疑字串，建議不要開啟。

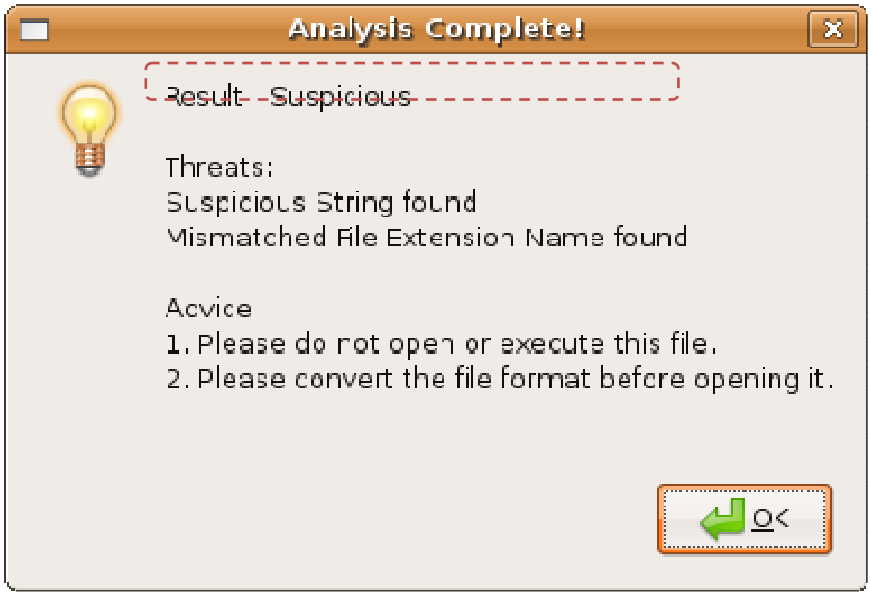


圖四十三：輸入含可疑字串的 html 檔案之結果

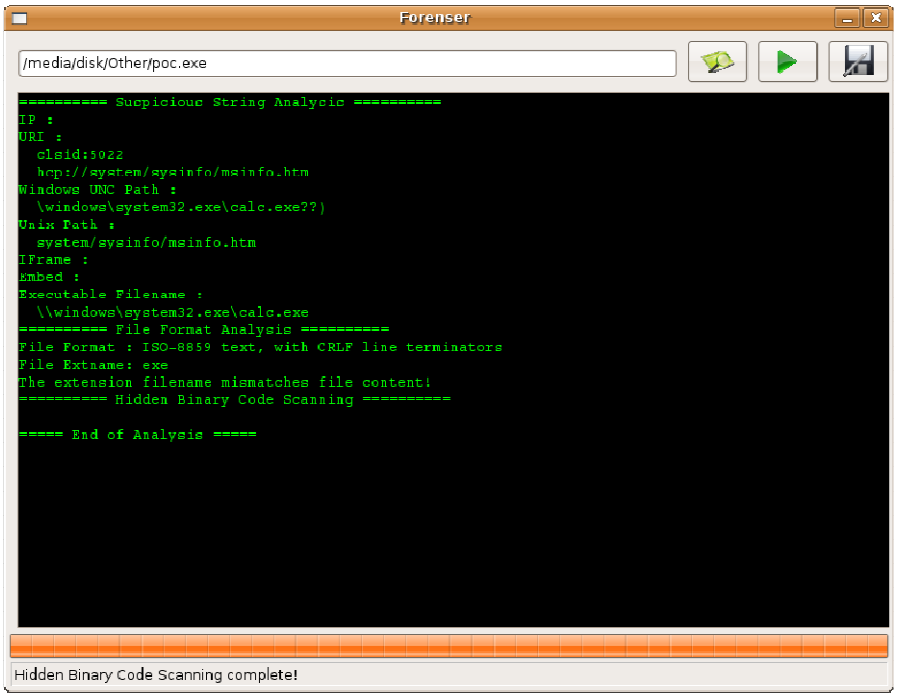


圖四十四：輸入含可疑字串的 html 檔案之報告

Case 6： 輸入一個檔案格式與副檔名不相同的檔案文件，本系統將會偵測出來，並且提醒使用者該檔案為可疑的檔案。

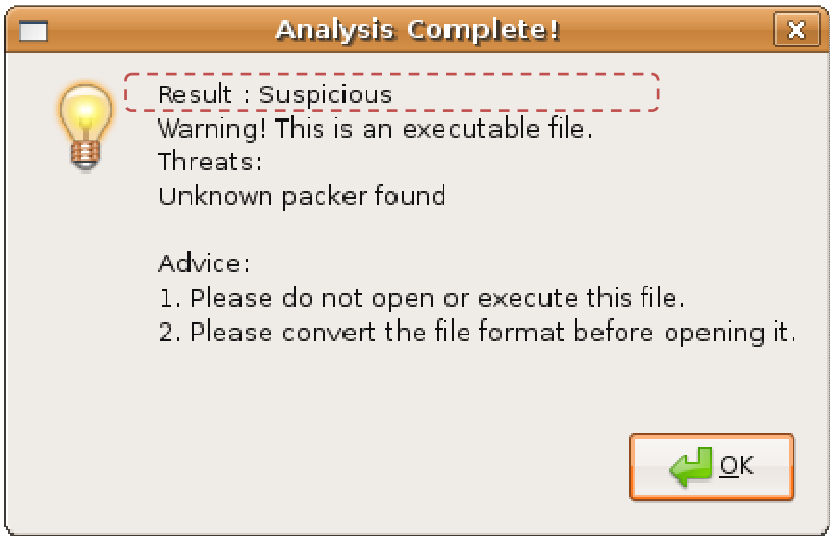


圖四十五：輸入副檔名與檔案格式不相符檔案之結果

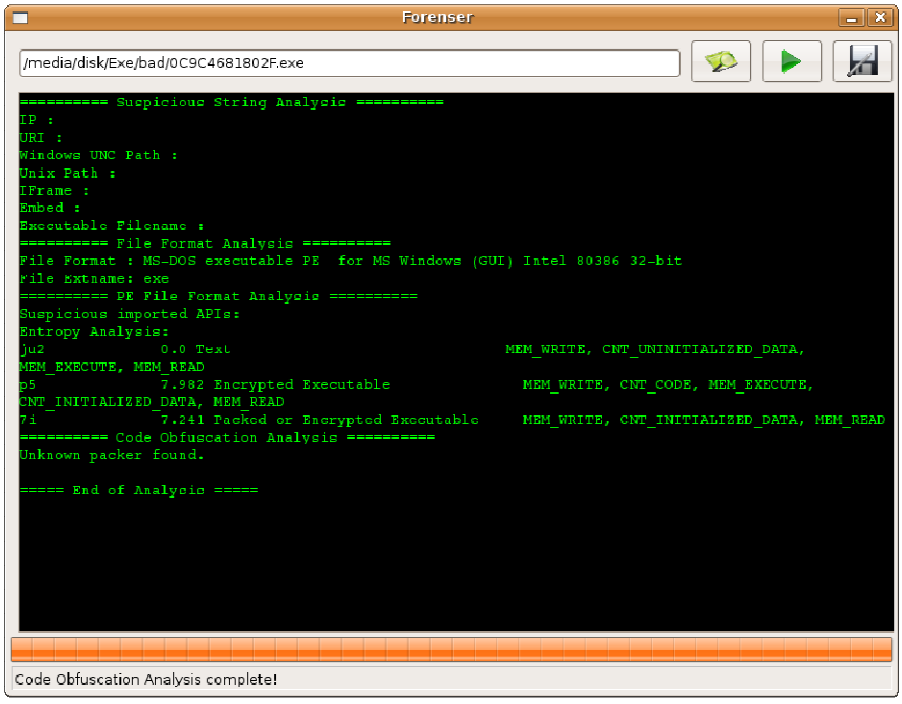


圖四十六：輸入副檔名與檔案格式不相符檔案之報告

Case 7： 輸入一個帶有未知殼的執行檔，本開發系統利用靜態分析來辨識該檔案為加殼檔案。

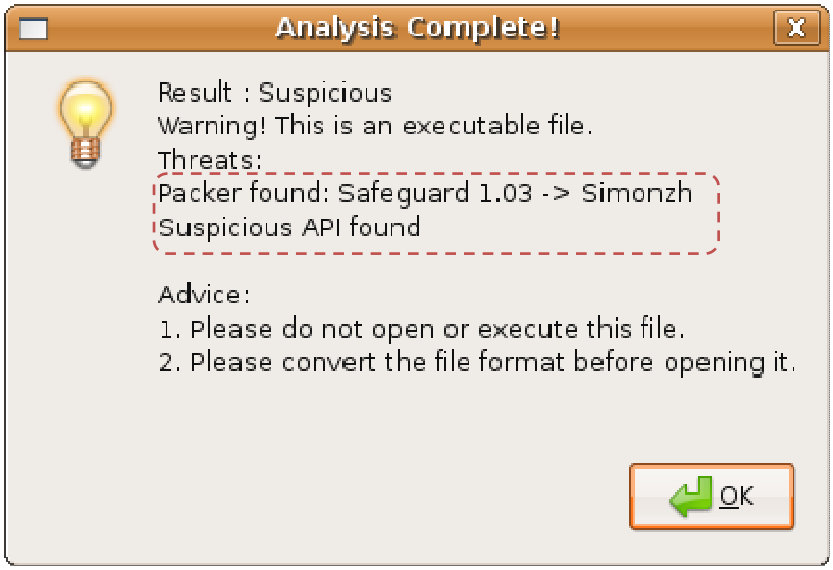


圖四十七：輸入一個帶有未知殼的執行檔之結果

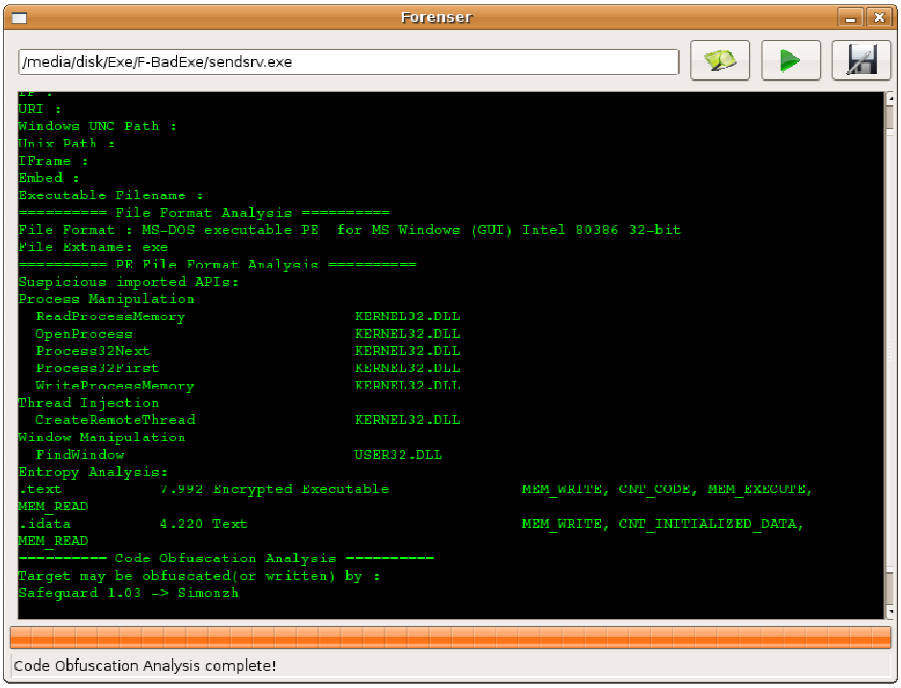


圖四十八：輸入一個帶有未知殼的執行檔之報告

Case 8： 輸入一個帶有可疑 API 與已知殼的執行檔，利用靜態分析來判斷出是加殼程式，且利用到可惜的 API。



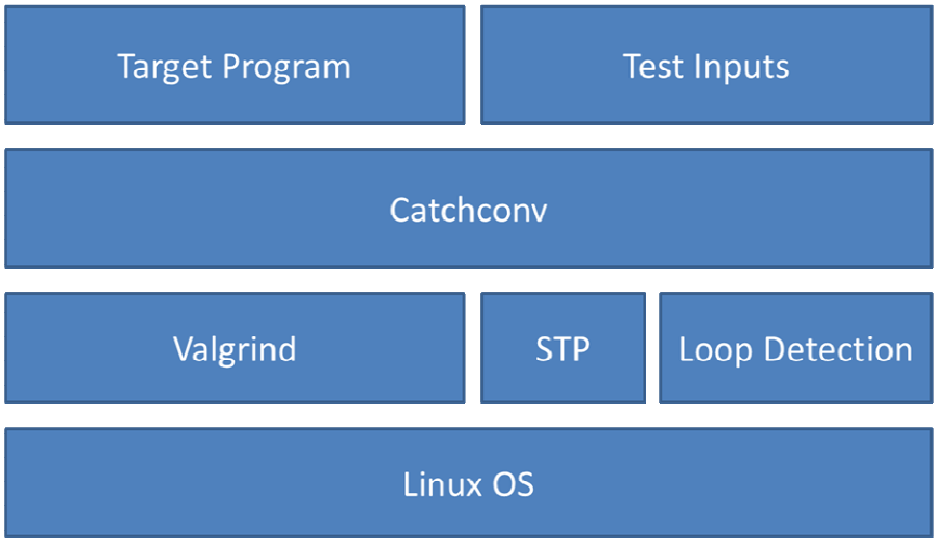
圖四十九：輸入一個帶有可疑 API 與已知殼的執行檔之結果



圖五十：輸入一個帶有可疑 API 與已知殼的執行檔之報告

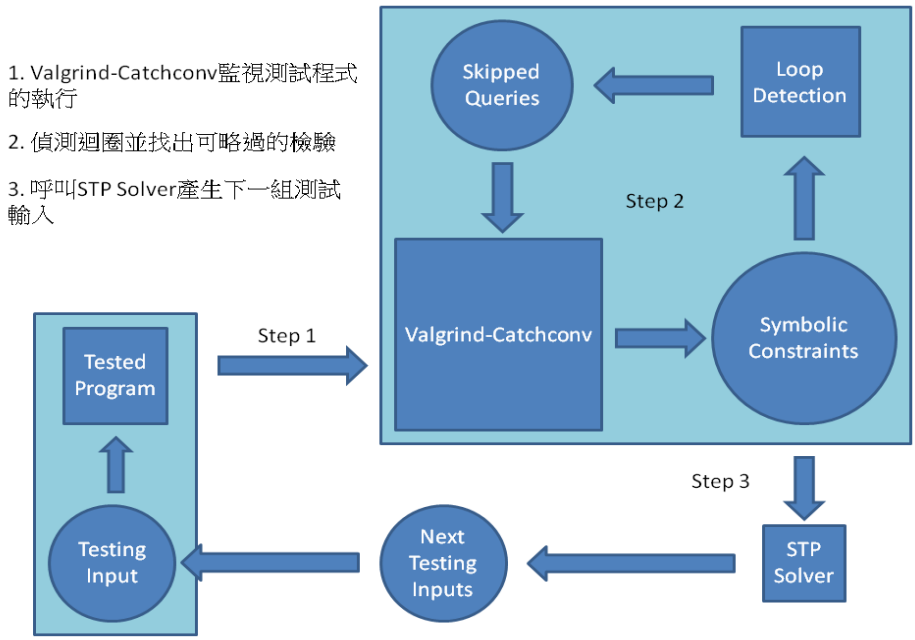
- **程式漏洞檢測系統**

Catchconv 以 Unix-Like 作業系統為系統平台，利用 Valgrind 為 instrumented framework，STP（Simple Theorem Prover）為 Query Solver，以及本計畫所開發之迴圈偵測程式為最佳化工具，所開發而成的 Gray-Box Testing 系統，不需要程式原始碼，只要有程式執行檔跟測試輸入，變能夠利用程式執行時期的資訊，產生觸發程式漏洞的測試資料。



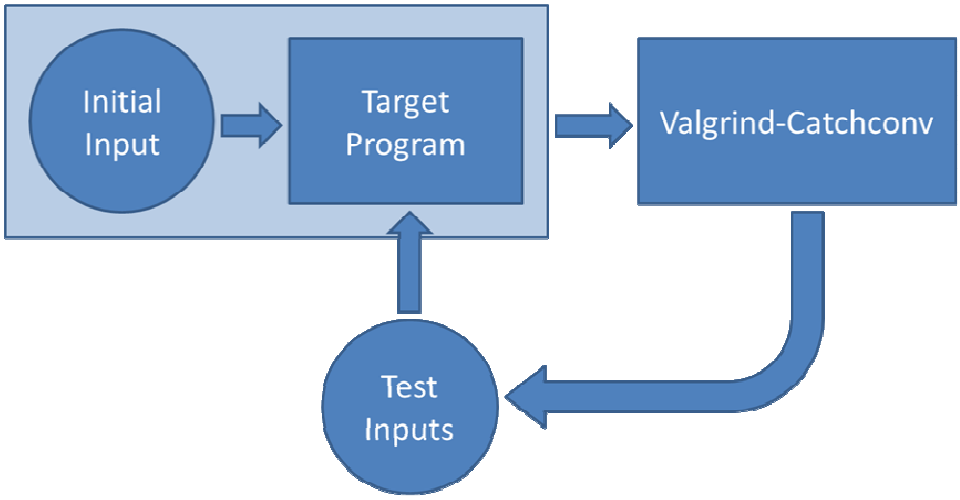
圖五十一：系統架構圖

Catchconv 在程式檢測上可以大致分為三個步驟：監控受測程式的執行過程，尋找可能發生的程式漏洞、偵測迴圈位置，省略重複的 Query 和程式區塊、呼叫 STP 產生 Query 的答案，進而發掘隱藏的程式漏洞，並自動產生測試資料。



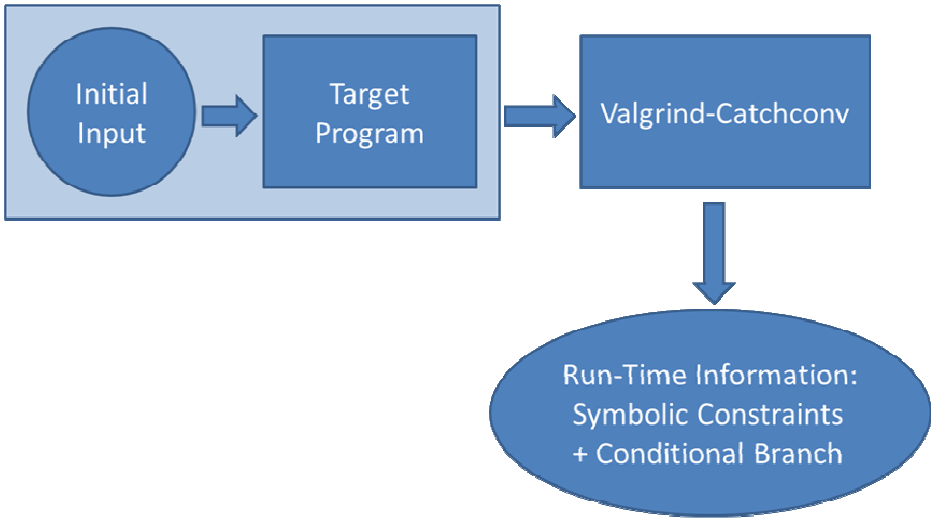
圖五十二：系統執行流程圖

利用 Catchconv 檢測程式漏洞，使用者必須先提供一測試輸入（Initial Input）給目標程式執行，Catchconv 監控程式的執行狀態，在檢測過程中利用目標程式的執行資訊，嘗試產生更多的測試輸入（Test Inputs）進行檢測。



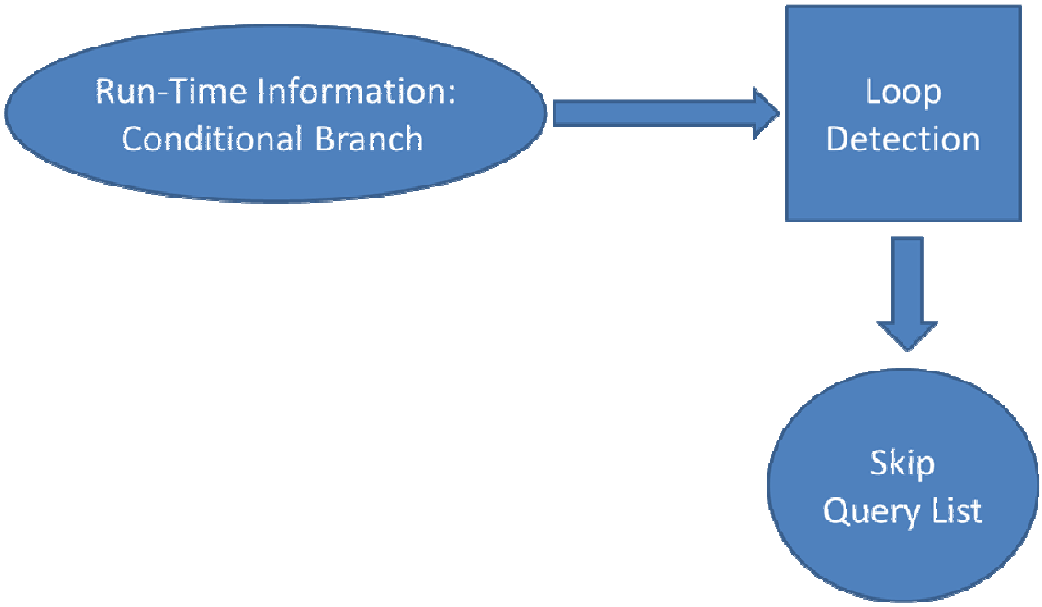
圖五十三：檢測流程圖

Catchconv 監控目標程式的執行狀態，檢測是否有程式漏洞發生，並且利用追蹤 Tainted Data Flow 的方式，找出執行過程中所有會受到測試輸入影響的變數、資料結構，以及所有根據這些變數、資料結構而有不同執行結果的迴圈、條件分支等程式區塊。Catchconv 接下來產生這些 Tainted Data 的資訊，和受到 Tainted Data 影響到的條件分支的 Query，以便利用 STP 來產生對應的測試資料。



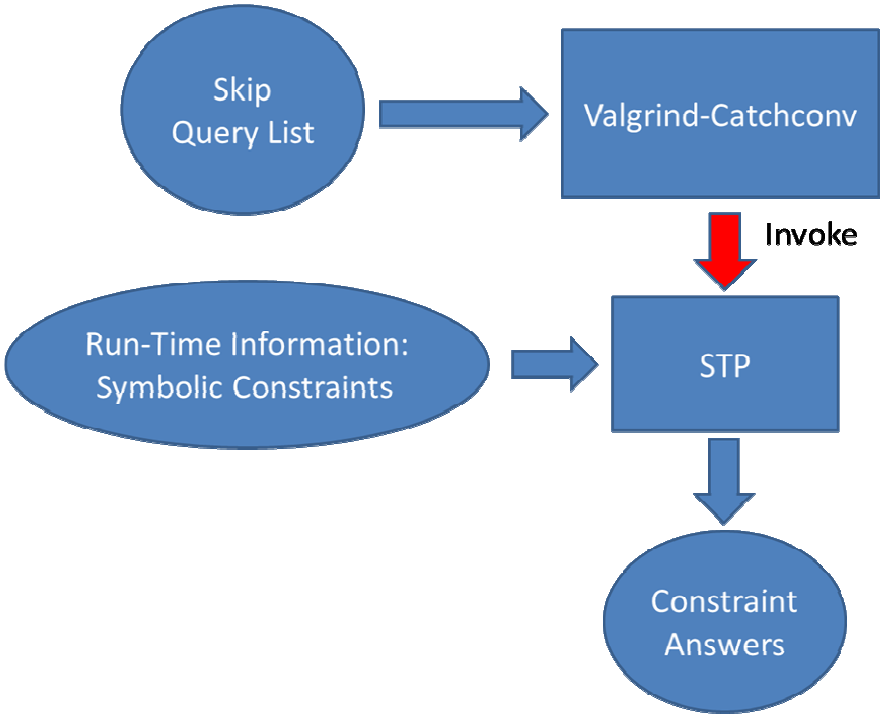
圖五十四：利用 STP 來產生對應的測試資料

程式中的迴圈會造成執行過程中，一直執行重複的程式區塊。為了讓 Catchconv 不受到重複的程式區塊影響，以至於檢測進度一直停留在單一區塊中，而忽視了程式中其他的部分，Catchconv 利用 Loop Detection 找出程式中的迴圈部分，並略過此迴圈所形成的條件分支，藉此消除迴圈所造成的影響，使 Catchconv 能夠檢測到其他的程式區塊。



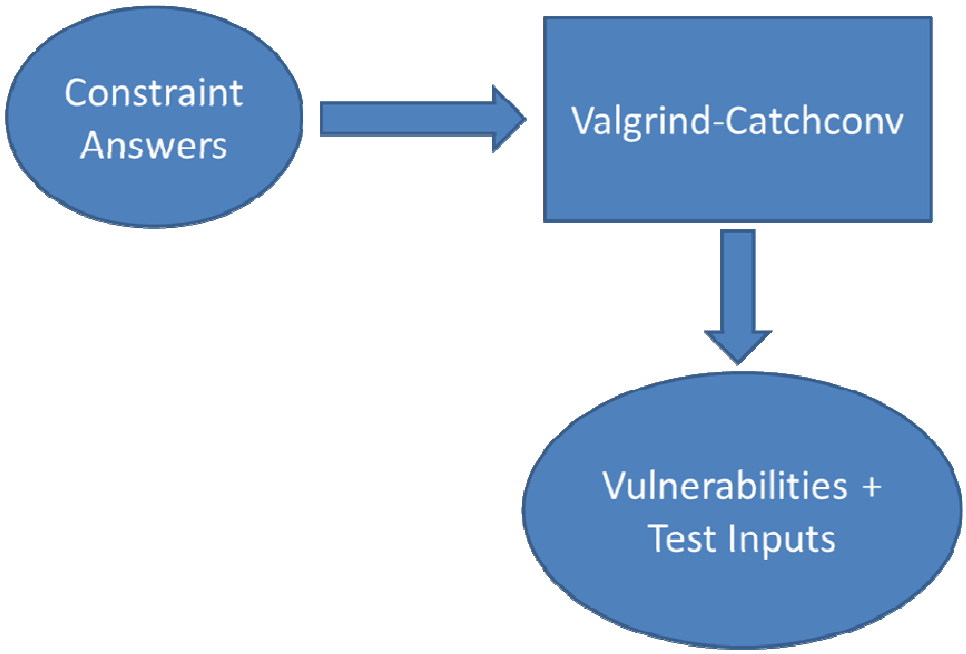
圖五十五：迴圈偵測機制

Catchconv 利用 Loop Detection 的結果，得知可被省略的 Query 清單，接下來將之前監控受測程式執行過程所產生的 Query，利用 STP 去尋找 Query 的解答，以便利用得到的解答去發掘潛在的程式漏洞，以及產生後續的測試資料。



圖五十六：取得可被省略的 QUERY 清單

利用 STP 求得的解答，Catchconv 尋找受測程式中潛在的程式漏洞，同時輸出可引發此程式漏洞的測試資料；Catchonv 也利用 STP 的解答，產生更進一步的測試輸入，將受測程式導向尚未探索過的程式區塊，以尋找受測程式其他部分所含有的程式漏洞。



圖五十七：尋找受測程式其他部分所含有的程式漏洞

在實際安裝使用 Catchconv 去檢測程式漏洞方面，使用者需要先確認系統是否已安裝 Catchconv 所需的相關套件和程式庫。要安裝相關套件及程式庫，必須具有系統管理者之權限，使用者可利用指令：

"*su*"或是"*sudo su*"

切換為系統管理者（root），接下來利用指令：

"*apt-get install autoconf automake g++ make libcurl3 libcurl3-dev libexpat1 libexpat1-dev mplayer cvs imagemagick*"

安裝 Catchconv 所需的套件及程式庫。

```
vink@debian:/$ sudo su
debian:/# apt-get install autoconf automake g++ make libcurl3 libcurl3-dev libex
pat1 libexpat1-dev mplayer cvs imagemagick
Reading package lists... Done
Building dependency tree
Reading state information... Done
autoconf is already the newest version.
automake is already the newest version.
g++ is already the newest version.
make is already the newest version.
libcurl3 is already the newest version.
Note, selecting libcurl4-openssl-dev instead of libcurl3-dev
libcurl4-openssl-dev is already the newest version.
libexpat1 is already the newest version.
libexpat1-dev is already the newest version.
mplayer is already the newest version.
cvs is already the newest version.
imagemagick is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 9 not upgraded.
debian:/# _
```

圖五十八：安裝 Catchconv 所需的套件及程式庫

完成 Catchconv 的相關套件、程式庫安裝後，使用者需要登入 Catchconv 的 CVS 系統，並且透過 CVS 下載最新版本的 Catchconv，使用者可利用指令：

"*cvs -d:pserver:anonymous@catchconv.cvs.sourceforge.net:/cvsroot/catchconv login*"

使用空白密碼登入 Catchconv 的 CVS 系統，接下來利用指令：

"*cvs -z3 -d:pserver:anonymous@catchconv.cvs.sourceforge.net:/cvsroot/catchconv co -P valgrind-catchconv*"

下載最新版本的 Catchconv；

```
debian:/# cvs -d:pserver:anonymous@catchconv.cvs.sourceforge.net:/cvsroot/catchc
onv login
Logging in to :pserver:anonymous@catchconv.cvs.sourceforge.net:2401/cvsroot/catc
hconv
CVS password:
debian:/# cvs -z3 -d:pserver:anonymous@catchconv.cvs.sourceforge.net:/cvsroot/ca
tchconv co -P valgrind-catchconv_
```

圖五十九：下載最新版本的 Catchconv

CVS 系統將會下載Catchconv所有的程式碼，根據網路速度及網路狀態需要花費不定的時間，通常約在十分鐘到三十分鐘左右；

```
U valgrind-catchconv/coregrind/m_sigframe/sigframe-amd64-linux.c
U valgrind-catchconv/coregrind/m_sigframe/sigframe-ppc32-linux.c
U valgrind-catchconv/coregrind/m_sigframe/sigframe-ppc64-linux.c
U valgrind-catchconv/coregrind/m_sigframe/sigframe-x86-linux.c
cvs checkout: Updating valgrind-catchconv/coregrind/m_syswrap
U valgrind-catchconv/coregrind/m_syswrap/priv_syswrap-generic.h
U valgrind-catchconv/coregrind/m_syswrap/priv_syswrap-linux-variants.h
U valgrind-catchconv/coregrind/m_syswrap/priv_syswrap-linux.h
U valgrind-catchconv/coregrind/m_syswrap/priv_syswrap-main.h
U valgrind-catchconv/coregrind/m_syswrap/priv_types_n_macros.h
U valgrind-catchconv/coregrind/m_syswrap/syscall-amd64-linux.S
U valgrind-catchconv/coregrind/m_syswrap/syscall-ppc32-linux.S
U valgrind-catchconv/coregrind/m_syswrap/syscall-ppc64-linux.S
U valgrind-catchconv/coregrind/m_syswrap/syscall-x86-linux.S
U valgrind-catchconv/coregrind/m_syswrap/syswrap-amd64-linux.c
U valgrind-catchconv/coregrind/m_syswrap/syswrap-generic.c
U valgrind-catchconv/coregrind/m_syswrap/syswrap-linux-variants.c
U valgrind-catchconv/coregrind/m_syswrap/syswrap-linux.c
U valgrind-catchconv/coregrind/m_syswrap/syswrap-main.c
U valgrind-catchconv/coregrind/m_syswrap/syswrap-ppc32-linux.c
U valgrind-catchconv/coregrind/m_syswrap/syswrap-ppc64-linux.c
U valgrind-catchconv/coregrind/m_syswrap/syswrap-x86-linux.c
cvs checkout: Updating valgrind-catchconv/dependencies
U valgrind-catchconv/dependencies/stp.gz
_
```

圖六十：CVS 系統將會下載 Catchconv 所有的程式碼

完成 Catchconv 的下載後，使用者可利用 Catchconv 的安裝程式進行安裝，使用者可利用指令：

"*cd valgrind-catchconv/*"

切換到 Catchconv 的程式碼目錄中，接下來利用指令：

"*chmod u+x bootstrap*"

設定 Catchconv 的安裝程式具有可以被使用者執行的權限，然後利用以下指令：

"*./bootstrap 安裝目錄*"

安裝 Catchconv 到想要的目錄下；如圖六十一安裝目錄為/vgcc。

```
debian:/# cd valgrind-catchconv/
debian:/valgrind-catchconv# chmod u+x bootstrap
debian:/valgrind-catchconv# ./bootstrap /vgcc_
```

圖六十一：安裝 Catchconv 到想要的目錄下，如圖安裝目錄為/vgcc。

Catchconv 進行安裝時，會先檢查相關的程式庫套件以及系統設定，已便進行後續的程式編譯及執行檔連結動作。

```
checking whether make sets $(MAKE)... yes
checking whether to enable maintainer-specific portions of Makefiles... no
checking whether ln -s works... yes
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking for style of include used by make... GNU
checking dependency style of gcc... gcc3
checking how to run the C preprocessor... gcc -E
checking for g++... g++
checking whether we are using the GNU C++ compiler... yes
checking whether g++ accepts -g... yes
checking dependency style of g++... gcc3
checking for ranlib... ranlib
checking for perl... /usr/bin/perl
checking for gdb... /usr/bin/gdb
checking dependency style of gcc... gcc3
checking for a supported version of gcc... ok (gcc (Debian 4.3.2-1.1) 4.3.2)
checking build system type... _
```

圖六十二：安裝時，會先檢查相關的程式庫套件以及系統設定

當 Catchconv 完成系統安裝後，需要使用者提供 e-mail 的資訊。使用者完成 e-mail 資訊設定後，Catchconv 將會提示使用者在進行程式檢測之前，需要設定 Catchconv 相關的環境變數，使用者可利用指令：

"*source 安裝目錄/bin/cc_envars*"

更新 Catchconv 環境變數；圖六十三安裝目錄為/vgcc。

```
make[2]: Entering directory `/vgcc/bin/zzuf-0.12/doc'
make[2]: Nothing to be done for `install-exec-am'.
test -z "/vgcc/share/man/man1" || /bin/mkdir -p "/vgcc/share/man/man1"
 /usr/bin/install -c -m 644 './zzuf.1' '/vgcc/share/man/man1/zzuf.1'
test -z "/vgcc/share/man/man3" || /bin/mkdir -p "/vgcc/share/man/man3"
 /usr/bin/install -c -m 644 './libzzuf.3' '/vgcc/share/man/man3/libzzuf.3'
make[2]: Leaving directory `/vgcc/bin/zzuf-0.12/doc'
make[1]: Leaving directory `/vgcc/bin/zzuf-0.12/doc'
make[1]: Entering directory `/vgcc/bin/zzuf-0.12'
make[2]: Entering directory `/vgcc/bin/zzuf-0.12'
make[2]: Nothing to be done for `install-exec-am'.
make[2]: Nothing to be done for `install-data-am'.
make[2]: Leaving directory `/vgcc/bin/zzuf-0.12'
make[1]: Leaving directory `/vgcc/bin/zzuf-0.12'
Current dir: /vgcc/bin

These scripts work together with www.metafuzz.com to give you
   an easy way to manage bugs you have found in different
   test runs. Metafuzz needs an e-mail address to correlate all the
   different runs you launch.
Please enter your e-mail address: vink.cs95g@nctu.edu.tw
Bootstrap complete. Type 'source /vgcc/bin/cc_envars' to set up environment vari
ables.
debian:/valgrind-catchconv# source /vgcc/bin/cc_envars
debian:/valgrind-catchconv# _
```

圖六十三：更新 Catchconv 環境變數，如圖安裝目錄為/vgcc。

利用 Catchconv 進行程式檢測，使用者需要先提供一個測試輸入，並且確定受測程式能夠執行，使用者可以利用指令：

“*docatchconv 測試輸入 受測程式執行命令*”

進行檢測動作；圖六十四 檢測/hello 這支程式，測試輸入為 test.txt，執行/hello 的命令為/hello test.txt。

```
debian:/# docatchconv test.txt /hello test.txt _
```

圖六十四：進行檢測動作

Catchconv 會在安裝目錄下的 tests 資料夾內，建立新的目錄以便存放所有檢測相關的資料，Catchconv 執行程式檢測的過程中，也會將檢測的狀態和相關資訊上傳到 Catchconv 的資料庫中，以便進行 Catchconv 的效能分析，作為 Catchonv 開發的參考；圖六十五 **錯誤! 找不到參照來源**。提供了使用者 e-mail、受測程式、測試資料、等相關的檢測資訊。

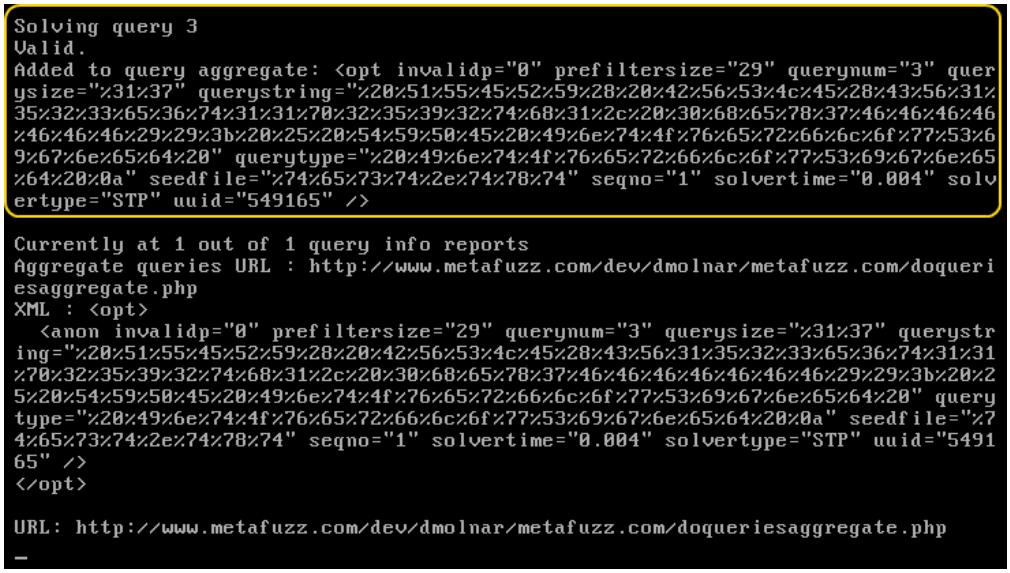
```
Content-Type: text/html

INSERT DELAYED INTO runresults (uuid,
email,
fuzz_type,
command_line,
input_file,
time_spent,
num_tests,
total_buckets,
unique_buckets,
blocks_added,
results_link)
VALUES (109558, 'vink.cs95g@nctu.edu.tw', 'catchconv', ' /hello test.txt', 'test
.txt',
'0', '0', '0', '0',
'0', '/exp/vgcc/tests/gensearch-1284989137-test.txt
')

JobID:
running memcheck:  /hello test.txt
vgcommand: ulimit -t 300; /exp/vgcc/inst/bin/valgrind --tool=memcheck --xml=yes
--log-file-exactly=test.txt-memcheck  /hello test.txt > test.txt-stdout 2> test.
txt-stderr
_
```

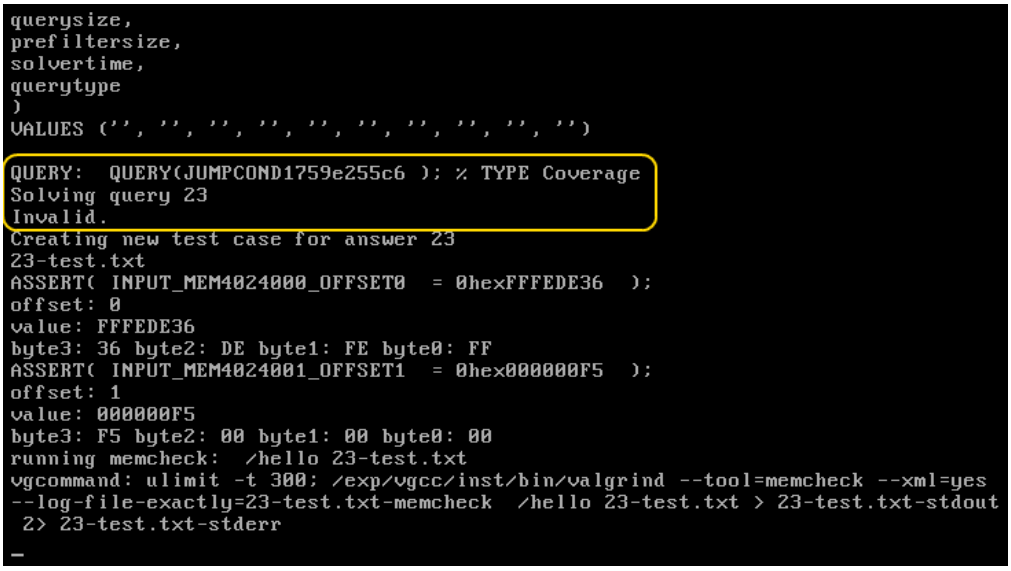
圖六十五：提供使用者 e-mail、受測程式、測試資料、等相關的檢測資訊。

Catchconv 檢測的過程中，會產生很多 Constraint Query，並且利用 STP 嘗試求得 Query 的解答；圖六十六為可解的 Constraint Query，STP 花費 0.004 秒求得此 Query 的解答。



圖六十六：為可解的 Constraint Query，STP 花費 0.004 秒求得此 Query 的解答

Catchconv 發現了一個條件分支，發出對應的 Query，並且產生相關的測試資料；圖六十七產生的 Query 為 QUERY(JUMPCOND1759e255c6)，此條件分支發生在程式區塊 1759，對應的測試資料為 23-test.txt。



圖六十七：產生對應的 Query

Catchconv 發現到程式漏洞後，會自動產生對應的測試資料，儲存在 results/errors 資料夾內；圖六十八測試資料為 12-2-test.txt，程式漏洞種類為可疑的系統呼叫參數。

```
VALUES ('', '', '', '', '', '', '', '', '', '')

QUERY:  QUERY(BULT( CV1535e18t4p3493th1 , 0hex000000FF));  % TYPE Conversion32to
8

Solving query 12
Invalid.
Creating new test case for answer 12
12-2-test.txt
ASSERT( INPUT_MEM4024003_OFFSET3  = 0hex00000100  );
offset: 3
value: 00000100
byte3: 00 byte2: 01 byte1: 00 byte0: 00
running memcheck:  /hello 12-2-test.txt
vgcommand: ulimit -t 300; /exp/vgcc/inst/bin/valgrind --tool=memcheck --xml=yes
--log-file-exactly=12-2-test.txt-memcheck  /hello 12-2-test.txt > 12-2-test.txt-
stdout 2> 12-2-test.txt-stderr
Added to bug aggregate: <opt filename="12-2-test.txt" fuzztype="catchconv" fuzzy
stackhash="e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855" seq
no="5" stackhash="822076393" type="SyscallParam" uuid="549165" />

Currently at 1 out of 1 bug info reports
Aggregate bugs URL : http://www.metafuzz.com/dev/dmolnar/metafuzz.com/dobugsaggr
egate.php
_
```

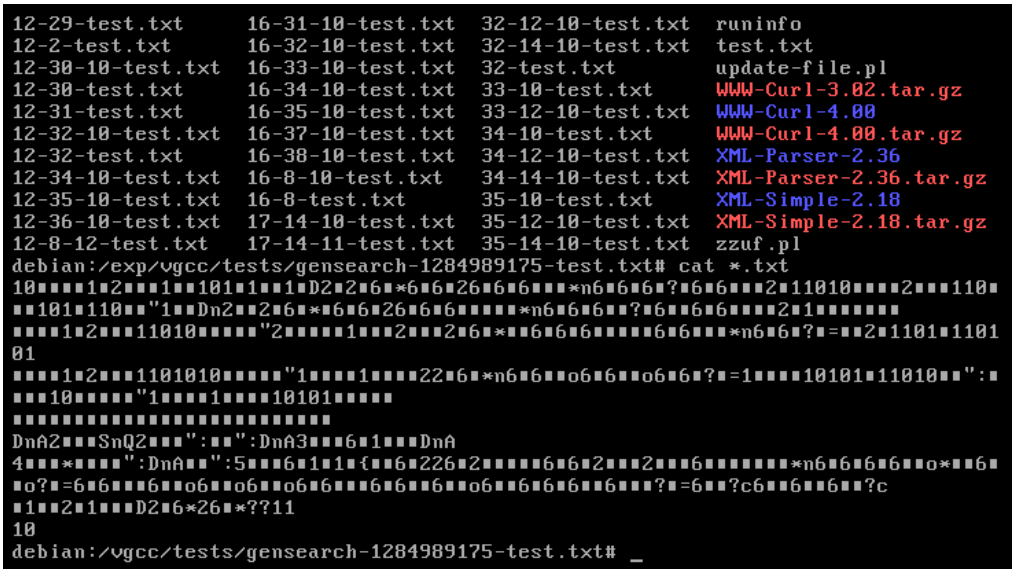
圖六十八：測試資料為 12-2-test.txt，程式漏洞種類為可疑的系統呼叫參數

在儲存檢測資料的資料夾中，可以看到目前產生的所有測試資料以及檢測相關資訊，其下的 results/errors 資料夾內則是儲存了可以引發程式漏洞的測試輸入，見下圖。

```
12-17-test.txt     16-10-test.txt     28-10-test.txt     libstats.pl
12-18-test.txt     16-11-test.txt     2-8-test.txt       libtriage.pl
12-20-8-test.txt   16-12-10-test.txt  28-test.txt        LOOP
12-21-8-test.txt   16-12-12-test.txt  29-10-test.txt     loop.err
12-21-test.txt     16-12-17-test.txt  29-12-10-test.txt  loop.path
12-22-test.txt     16-12-18-test.txt  29-16-10-test.txt  malformedxml
12-23-test.txt     16-12-2-test.txt   29-test.txt        POE
12-24-test.txt     16-12-8-test.txt   2-test.txt         POE-cc.tar.gz
12-25-test.txt     16-2-10-test.txt   30-10-test.txt     POE.pm
12-26-test.txt     16-27-10-test.txt  30-test.txt        prune.pl
12-27-10-test.txt  16-28-10-test.txt  31-10-test.txt     QUERY
12-27-test.txt     16-29-10-test.txt  31-test.txt        rco
12-28-test.txt     16-30-10-test.txt  32-10-test.txt     results
12-29-test.txt     16-31-10-test.txt  32-12-10-test.txt  runinfo
12-2-test.txt      16-32-10-test.txt  32-14-10-test.txt  test.txt
12-30-10-test.txt  16-33-10-test.txt  32-test.txt        update-file.pl
12-30-test.txt     16-34-10-test.txt  33-10-test.txt     WWW-Curl-3.02.tar.gz
12-31-test.txt     16-35-10-test.txt  33-12-10-test.txt  WWW-Curl-4.00
12-32-10-test.txt  16-37-10-test.txt  34-10-test.txt     WWW-Curl-4.00.tar.gz
12-32-test.txt     16-38-10-test.txt  34-12-10-test.txt  XML-Parser-2.36
12-34-10-test.txt  16-8-10-test.txt   34-14-10-test.txt  XML-Parser-2.36.tar.gz
12-35-10-test.txt  16-8-test.txt      35-10-test.txt     XML-Simple-2.18
12-36-10-test.txt  17-14-10-test.txt  35-12-10-test.txt  XML-Simple-2.18.tar.gz
12-8-12-test.txt   17-14-11-test.txt  35-14-10-test.txt  zzuf.pl
debian:/vgcc/tests/gensearch-1284989175-test.txt# _
```

圖六十九：可以看到目前產生的所有測試資料以及檢測相關資訊

使用者可以利用 cat 指令觀察所產生的測試輸入，因採用 Fuzz Testing 的緣故，測試輸入產生了很多不可印的字元及符號；見圖七十。



圖七十：利用 cat 指令觀察所產生的測試輸入

Catchconv 在檢測過程中產生的執行時期資訊，包括變數、資料結構的資訊，條件分支所產生的 Query，都儲存在 driverbase 這個檔案裡；見下圖。

```
XXX e: 143 t: 27
ASSERT(CV1650e143t11p27909th1[7:0] = CV1650e143t27p27909th1[7:0] );

CV1650e143t29p27909th1 : BITVECTOR(32);
ASSERT( CV1650e143t29p27909th1 = CV1650e143t11p27909th1);

CV1651e144t9p27909th1 : BITVECTOR(32);

CV1651e144t7p27909th1 : BITVECTOR(32);
ASSERT( CV1651e144t9p27909th1[7:0] = CV1528e11t17p27909th1[7:0] );

CV1651e144t8p27909th1 : BITVECTOR(32);
ASSERT( CV1651e144t8p27909th1 = CV1651e144t9p27909th1);

CV1651e144t2p27909th1 : BITVECTOR(32);
XXX e: 144 t: 8
ASSERT(CV1651e144t2p27909th1[7:0] = CV1651e144t8p27909th1[7:0] );

CV1651e144t10p27909th1 : BITVECTOR(32);
ASSERT( CV1651e144t10p27909th1 = CV1651e144t2p27909th1);
XXX Stat counters:
XXX Unhandled binop: 0
XXX Unhandled unop: 0
XXX CalcCondition: 0
debian:/vgcc/tests/gensearch-1284989175-test.txt# _
```

圖七十一：檢測過程產生的執行時期資訊

從 driverbase 檔案中節取出來和條件分支有關的 Query，被儲存於 QUERY 檔案中，包含 Query 資訊以及條件分支的目標位置。

```
XXX QUERY(JUMPCOND6951e87795c40 ); % TYPE Coverage XXX The destAddr = 0x5BBC7D7 ;
XXX QUERY(JUMPCOND6943e87813c41 ); % TYPE Coverage XXX The destAddr = 0x819A1B8 ;
XXX QUERY(JUMPCOND6951e87816c42 ); % TYPE Coverage XXX The destAddr = 0x5BBC7D7 ;
XXX QUERY(JUMPCOND6943e87834c43 ); % TYPE Coverage XXX The destAddr = 0x819A1B8 ;
```

```
XXX QUERY(JUMPCOND6951e87837c44 ); % TYPE Coverage XXX The destAddr = 0x5BBC7D7 ;
XXX QUERY(JUMPCOND6943e87855c45 ); % TYPE Coverage XXX The destAddr = 0x819A1B8 ;
XXX QUERY(JUMPCOND6951e87858c46 ); % TYPE Coverage XXX The destAddr = 0x5BBC7D7 ;
XXX QUERY(JUMPCOND6943e87876c47 ); % TYPE Coverage XXX The destAddr = 0x819A1B8 ;
XXX QUERY(JUMPCOND6951e87879c48 ); % TYPE Coverage XXX The destAddr = 0x5BBC7D7 ;
XXX QUERY(JUMPCOND6943e87897c49 ); % TYPE Coverage XXX The destAddr = 0x819A1B8 ;
XXX QUERY(JUMPCOND6951e87900c50 ); % TYPE Coverage XXX The destAddr = 0x5BBC7D7 ;
XXX QUERY(JUMPCOND6943e87918c51 ); % TYPE Coverage XXX The destAddr = 0x819A1B8 ;
XXX QUERY(JUMPCOND6951e87921c52 ); % TYPE Coverage XXX The destAddr = 0x5BBC7D7 ;
XXX QUERY(JUMPCOND6943e87939c53 ); % TYPE Coverage XXX The destAddr = 0x819A1B8 ;
XXX QUERY(JUMPCOND6951e87942c54 ); % TYPE Coverage XXX The destAddr = 0x5BBC7D7 ;
XXX QUERY(JUMPCOND6943e87960c55 ); % TYPE Coverage XXX The destAddr = 0x819A1B8 ;
XXX QUERY(JUMPCOND6951e87963c56 ); % TYPE Coverage XXX The destAddr = 0x5BBC7D7 ;
XXX QUERY(JUMPCOND6943e87981c57 ); % TYPE Coverage XXX The destAddr = 0x819A1B8 ;
XXX QUERY(JUMPCOND6951e87984c58 ); % TYPE Coverage XXX The destAddr = 0x5BBC7D7 ;
XXX QUERY(JUMPCOND6943e88002c59 ); % TYPE Coverage XXX The destAddr = 0x819A1B8 ;
XXX QUERY( NOT JUMPCOND6951e88005c60 ); % TYPE Coverage XXX The destAddr = 0x5BBC7D7 ;
```

利用 Loop Detection 掃描分析條件分支的 Query，提供重複的程式區塊資訊，儲存於 QUERY 檔案中；如下是以上面的 Loop Detection 的輸入資料，偵測到位於程式區塊 6951 的條件分支，當其目的位置為 0x5BBC7D7 時，是由迴圈所產生的條件分支。

```
0x5BBC7D7 1
1 2
JUMPCOND6951e87795c40
NOT JUMPCOND6951e88005c60
```

Catchconv 找到的程式漏洞將會存放於 results/errors 資料夾內，包含可引發此漏洞的測試輸入，使用者可利用 tree 指令列出 results/errors 資料夾內包含的所有漏洞清單，得知受測程式的檢測結果。

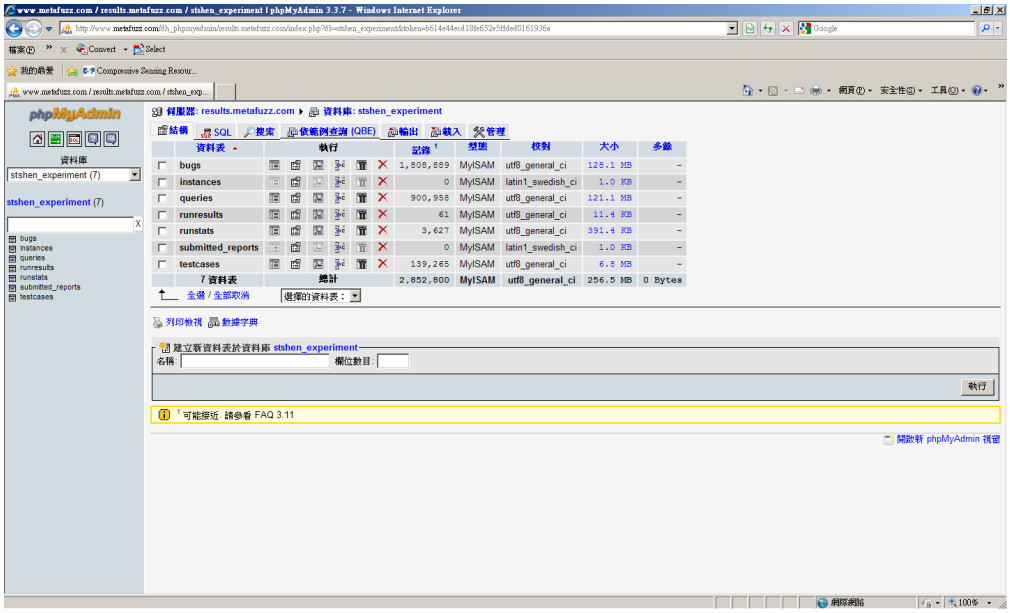
```
.
|-- SyscallParam
|   |-- 1679461208
|   |   |-- test.txt
|   |   |-- test.txt-memcheck
|   |   |-- test.txt-stderr
|   |   `-- test.txt-stdout
|   |-- 1793740102
|   |   |-- test.txt
```

```
|     |     |-- test.txt-memcheck
|     |     |-- test.txt-stderr
|     |     `-- test.txt-stdout
|     `-- 822076393
|           |-- test.txt
|           |-- test.txt-memcheck
|           |-- test.txt-stderr
|           `-- test.txt-stdout
|-- UninitCondition
|     |-- 1058365317
|     |     |-- test.txt
|     |     |-- test.txt-memcheck
|     |     |-- test.txt-stderr
|     |     `-- test.txt-stdout
|     |-- 1413540393
|     |     |-- test.txt
|     |     |-- test.txt-memcheck
|     |     |-- test.txt-stderr
|     |     `-- test.txt-stdout
|     |-- 1428533681
|     |     |-- test.txt
|     |     |-- test.txt-memcheck
|     |     |-- test.txt-stderr
|     |     `-- test.txt-stdout
|     |-- 2220999866
|     |     |-- test.txt
|     |     |-- test.txt-memcheck
|     |     |-- test.txt-stderr
|     |     `-- test.txt-stdout
|     |-- 2234422911
|     |     |-- test.txt
|     |     |-- test.txt-memcheck
|     |     |-- test.txt-stderr
|     |     `-- test.txt-stdout
|     `-- 2558958562
|           |-- test.txt
|           |-- test.txt-memcheck
|           |-- test.txt-stderr
|           `-- test.txt-stdout
|-- crashes
```

```
|-- test.txt
|-- test.txt-memcheck
|-- test.txt-stderr
`-- test.txt-stdout

12 directories, 41 files
```

為了方便評量 Catchconv 的效能及準確度，需要 Catchconv 執行檢測時的相關狀態和資訊，Catchconv 在檢測過程中，會自動上除這些狀態和資訊到資料庫系統中，Catchconv 的開發人員可以連上 http://www.metafuzz.com 去檢閱、分析這些資訊與狀態，提供 Catchconv 最佳化的可能性和效能分析與評量。



圖七十二：Catchconv 資料庫系統

五、 技術方案優越性

本計畫成果包含多項可實際應用之系統以及技術，以下各節將針對所提出之各項技術與現行之狀況比較，並分析其優越性：

- **無線異質網路測試平台**

本計畫開發的異質無線網路模擬平台與現有之網路測試平台之比較表如表九所示：(1) 本平台同時具備有線/無線之網路模擬功能，並支援較多之無線網路協定 (Wi-Fi、WiMAX)，易於使用者建立多樣化之異質無線網路實驗；(2) 本平台也針對無線網路模擬提供安全攻防模組，協助使用者驗證網路協定之安全性；(3) 本平台採用仿真無線訊號的方式模擬無線網路實驗，使用者能自訂較廣泛的無線網路拓樸，而不受限於實體無線傳輸裝置之實際分布位置；(4) 本計畫提出的虛擬驅動程式和虛擬天線之設計，易於無線網路之開發者將新型無線網路的協定與訊號傳輸方式移植本平台上進行測試，加速新型無線網路之開發與佈局。

本計畫之研發團隊亦與 UCB 的 DETER 平台開發團隊密切合作，進行技術整合與相容性改良，使本計畫能相容於 DETER Federation 之架構，以期能透過網路與 EmuLab、DETER 等平台共享實驗資源，提供使用者進行大規模無線網路模擬實驗之資源。

	Emulab	DETER	CMU [7]	本平台
有線網路	✓	✓	×	✓
無線網路	✓ (Wi-Fi, WSN)	×	✓ (Wi-Fi)	✓ (Wi-Fi, WiMAX)
安全攻防模組	×	✓	×	✓
自訂無線拓樸	Δ (無線拓樸受限於實體裝置的實際位置，使用者需從實體裝置中選出符合需求的節點)	×	Δ (無線拓樸受限於實體裝置的實際位置，使用者需從實體裝置中選出符合需求的節點)	✓
無線網路技術擴充性	✓ (採 SDR 技術，可以軟體設定無線通訊技術)	×	×	✓ (採虛擬天線與虛擬驅動設計，易於擴充可支援的無線技術)

表九：本平台與其他有線/無線網路測試平台之比較

- **行動平台的安全管理機制**

行動平台的安全機制 – 入侵偵測系統

- 為目前已知文獻中首度將入侵偵測系統(IDS)移植至 Android 手機，可加強行動裝置無線上網的安全性。
- 前端程式和後端執行檔為獨立模組化設計，前後端系統易於抽換或獨立升級，未來擴充彈性大。
- 支援入侵行為特徵資料庫的擴增，使用者可自行擴充入侵行為特徵以增強其安全性。
- 電源消耗量低，低耗電量有利於常駐使用。
- 記憶體使用量依據載入之特徵資料庫大小決定，可依行動裝置之硬體條件彈性決定特徵載入數量。
- 播放網路影音串流之電力消耗（見表十）

	每小時電力消耗量	預估電力持續時間
原始狀態	7.2%	13.9 小時
開啟IDS	9.6%	10.4 小時

表十：播放網路影音串流之電力消耗

- 待機狀態之電力消耗（見表十一）

	每小時電力消耗量	預估電力持續時間
原始狀態	2.8%	33.4 小時
開啟IDS	4.6%	21.7 小時

表十一：待機狀態之電力消耗

行動平台的安全機制 – 安全雲端儲存系統

(1) 系統理論設計部份

研究成果的主要貢獻可以從兩個角度來說明。從學術理論上來看，我們提供了一個結合了容錯技術與公開金鑰加密系統的密碼學工具，這個工具能夠在一個非集中式的儲存系統環境中被使用，使得系統同時具有資料可信賴與高度隱私性並且兼顧了分散式的優點，另外針對系統中資料儲存的取回正確率上，我們亦提供了一個完整的分析方式並建議了一組通用的系統參數。

從儲存系統發展與應用上來看，我們強調了資料隱私性在雲端儲存系統上的重要性與一個強度上的分野，早期網路儲存系統的隱私性是建立在完全信任儲存伺服器的假設下，僅對登入的使用者進行身分認證，我們則是強調資料隱私性的強度應該要能夠消除對儲存伺服器的信任的假設條件。

(2) 系統實現設計部份

- ◆ 系統能有效率地將檔案加解密以及儲存，同時也提供網頁介面讓使用者能輕易管理檔案。
- ◆ 系統位於應用層，可包裝成 Android 的安裝檔.apk，使用者可簡單地安裝、卸載、及維護。
- ◆ 本系統可支援多種作業系統與多種無線裝置，亦有利於移植至不同平台。

- **惡意檔案文件分析系統**

本系統主要分析普通檔案文件為主，相較於一般傳統的防毒軟體不同的地方在於它藉著許多啟發式的分析手法，來達到判斷該文件檔案是否有可疑的行為，並且建議使用者在不信任檔案來源的情況下，是否可直接開啟該檔案。以下有三大技術優越性：*(１）多面向的分析技術、(２）新穎且準確的分析、(３）可調式的設定檔。

(1) 多面向的分析技術

靜態分析能夠快速地取出檔案資訊，但是無法得知該檔案實際執行時的資訊。相對來說，動態分析需要比較重的計算資源，所以無法叫即時的回應給使用者知道。可疑字串分析與檔案格式分析皆為靜態分析的一種，這些利用正規表示式 (regular expression) 和檔案內容特徵 (pattern-based) 來觀測目標檔案是否含有可疑的內容，或是騙取使用者點選的意圖，來做第一階段的檢驗。接下來，將會結合動態的模擬處理器對記憶體存取的行為，來得知該檔案是否嵌有惡意的 shell code 來判斷該檔案的安全性。以上技術是目前防毒軟體所沒有的，原因在於防毒軟體為了要提供較低的誤判率 (false positive rate)，故無法作太多推測性的分析方式。但對於較進階的電腦使用者或資安人員，較希望能夠有多方面的資訊來防止新穎的病毒入侵，以免造成極大的損失。為了能夠更全面性的檢測檔案的安全性，藉著結合靜態與動態的分析方式，達到多面向的分析，以提供更準確的判斷依據。

(2) 偵測動態取得記憶體位置的分析技術

本系統利用模擬 CPU 執行來得知該文件檔案是否內嵌可疑的記憶體存取行為，其動態啟發式偵測法對零時攻擊(Zero-day Attack)亦具有偵測能力。此分析技術將包含一個模擬的 Intel x86 CPU，來供使用者得知堆疊與 CPU 暫存器的狀況。我們透過這個 CPU 模擬器，可以完全觀測此程式運行的結果，並且檢查會不會取出事先寫入在記憶體中的特定數值。若該數值若存在在暫存器中，我們可以說此程式具有動態取得記憶體位址的行為。該偵測技術為動態分析，故需要較長的時間來運行，相對於靜態分析來說，該技術是本研究的一大突破項目之一。

(3) 可調式的設定檔

設定檔的可調整性讓該系統能夠有彈性的擴充。由於惡意程式的攻擊手法日新月異，為了方便往後的擴充性，本系統將提供可調整參數的設定值。包含如下：觀測之檔案格式與副檔名，印出偵錯值，判斷臨界值設定。這些參數可能依照不同的輸入樣本，而可以有不同的設定值以加強偵察的能力。

- **程式漏洞檢測系統**

驗證程式的正確性及安全性相當重要，但卻是不容易達到的目標，目前的檢測方法可區分為三類：White-Box Testing、Black-Box Testing 以及 Gray-Box Testing，其中 Gray-Box Testing 則整合了 White-Box Testing 及 Black-Box Testing 的優點，是現今最有效的檢測方式。而本系統即為 Gray-Box Testing，具體的系統優勢如下：

- 本系統屬 Gray-Box Test，為近年最受重視的檢測技術之一。
- 不需程式原始碼即可進行測試，對於無法取得原始碼的程式亦能進行檢測。
- 將 Fuzz testing 之測試資料輸入被檢測程式，直接觀察其行為是否正常，不用花費時間分析複雜的原始碼。
- 合併執行時期取得之資訊，進一步了解程式內部運作，藉此產生有用的測試資料，改進單純 Fuzz testing 無法輕易涵蓋所有執行路徑的問題。
- 自動產生漏洞測試資料，可幫助程式人員找出漏洞所在。
- 迴圈處理機制可跳過重複執行的程式迴圈，將檢測能量集中在不重複的程式區段。

肆、 結論與展望

行動網路資訊服務為未來資訊產業的趨勢，本計畫開發之「異質無線網路模擬平台」目標在於提供使用者一個安全的、具隔離性的、可重覆性的異質無線網路攻防實驗環境，並且期望以高彈性和易擴充性的架構讓使用者能透過此平台進行更多新興網路技術的驗證與測試。本團隊未來將持續與 UCB 之傑出學者進行合作，吸取 DETER 測試平台之建置經驗，加速異質無線網路模擬平台之擴充，實現更全面之攻防模組以及更具前瞻性之測試工具，以期協助推動國內資安產業之發展，加速安全機制及安全產品之研發，減少產業單位開發新產品的時間與人力成本。

而在「安全雲端儲存系統」的研究中，我們已發展出一套具高度安全性且低成本的分散式網路儲存系統，其系統設計同時提供儲存服務和金鑰管理。本系統是架構在分散式的環境，且每個儲存伺服器在執行加密的過程中分散各別執行，因此即使是在不受信任的儲存環境，此分佈式網路儲存系統依舊可以保障個人的隱私。由於本分佈式網路儲存系統的設計目的是為了便於行動智慧型裝置使用，因此未來其服務執行的效率與系統的易用程度亦為可以進行研究與改良的項目。

在「入侵偵測系統」的研究中，鑒於手機的功能越來越強大，構造也越來越複雜。隨著手機的計算能力增加，可以預測未來手機將會具有多樣化的功能性，故手機上的安全性將會更難確保。我們將功能強大的 Snort 移植到 Android 上，並利用其封包搜尋的功能，對傳入手機的網路封包進行檢查，以達到監控惡意封包之目的。只要定期選定 Linux 系統上更新的穩定版 Snort 去進行 Cross- compilation，並且更新 Snort 對應的 Rule 設定檔即可，前端程式幾乎可以設計成和執行檔獨立的方式，以減少維護以及程式開發的成本。此外在電源消耗的部分，根據我們的實驗結果，待機時每小時只會多消耗 2%的電量，上網時每小時只需多負擔不到 1%的電量，在電源消耗上是非常經濟的一個結果。未來我們可以持續利用這種方式移植更多適合的安全軟體至 Android 上，且只要撰寫前端控制程式，即可使用這些安全軟體增加 Android 的安全性。

除此之外，本計畫所開發之「檔案文件檢測系統」，未來將研究區分為使用者端與專家端：在使用者端可安裝前端鑑識程式，對檔案進行初步的掃描，程式將產生報告給使用者參考，並讓使用者選擇是否要將此程式送至專家端進行進一步的鑑識工作。在專家端我們將建立虛擬機器以處理待檢測的檔案，系統中包含的檢測軟體將能進行更深入的分析以供專家參考。

最後，驗證程式的正確性及安全性相當重要，但卻是不容易達到的目標，Gray-Box Testing 整合了 White-Box Testing 及 Black-Box Testing 的優點，是現今最有效的檢測方式。因此本計畫與美國加州大學柏克萊分校(UCB)合作，與 EECS 系上教授 David Wagner 以及他的博士生 David Molnar 共同進行 Catchconv 系統的開發；在 David Wagner 和 David Molnar 的協助下，進行了 Catchconv 程式檢測的最佳化。針對 Catchconv 所進行的程式檢測最佳化部分，皆已完成開發及實作，並整合於 Catchconv 計畫中，放置於 Source Forge 網站上，使用者可利用 CVS 下載經過最佳化的最新版本之 Catchconv。希望透過 Open Source 的方式，能夠推廣並吸引更多使用者利用 Catchconv，檢查軟體是否存在某些漏

洞，幫助程式設計師快速地發現程式中的 bug，增進軟體的可靠度及開發效率。也希望能夠有更多的開發人員能夠加入 Catchconv 的開發計畫，使 Catchconv 在使用介面、自動化、核心最佳化以及支援的程式漏洞種類上，能夠更加完善。

綜合以上所述，本計畫除了提出一套具有多面向的「行動平台安全檢測方案」之外，亦已與 UC Berkeley 的相關研究團隊建立長期而穩定的交流研究關係；本計畫亦將部分研究成果於 OpenFoundry 上建立專案並開放原始碼，可望能將安全技術研發與自由軟體社群連結，並引入自由軟體社群的研發能量以吸引產業界投入的意願、厚植國內資訊安全技術的研發潛力。

伍、 參考文獻

[1] B. White et al., "An Integrated Experimental Environment for Distributed Systems and Networks," in *Proc. Of the 5th USENIX Symposium on Operating Systems Design and Implementation*, Dec. 2002

[2] T. Benzel et al., "Experience with DETER: A Testbed for Security Research," in *Proc. Of Tridentcom*, 2006

[3] J. Mirkovic et al., "Automating DDoS Experimentation," in *Proc. Of the DETER Community Workshop on Cyber Security Experimentation and Test*, 2007

[4] D. Raychaudhuri et al., "Overview of the ORBIT Radio Grid Testbed for Evaluation of Next-Generation Wireless Network Protocols," in *IEEE Wireless Communications and Networking Conference*, vol. 3, pp. 1664 – 1669, 2005

[5] P. Mahadevan et al., "MobiNet: a Scalable Emulation Infrastructure for Ad Hoc and Wireless Networks," in *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 10, no. 2, pp. 26—37, 2006

[6] "Wireshark," http://www.wireshark.org/

[7] K. Borries et al., "FPGA-Based Channel Simulator for a Wireless Network Emulator," *in IEEE 2009 IEEE 67th Vehicular Technology Conference*, 2009

[8] J. Kubiatowicz, D. Bindel, Y. Chen, S.E. Czerwinski, P.R. Eaton, D. Geels, R. Gummadi, S.C. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B.Y. Zhao, "Oceanstore: An Architecture for Global-Scale Persistent Storage," Proc. Ninth Int'l Conf. Architec- tural Support for Programming Languages and Operating Systems (ASPLOS), vol. 35, pp. 190-201, 2000.

[9] S.C. Rhea, C. Wells, P.R. Eaton, D. Geels, B.Y. Zhao, H. Weath- erspoon, and J. Kubiatowicz, "Maintenance-Free Global Data Storage," IEEE Internet Computing, vol. 5, no. 5, pp. 40-49, Sept. 2001.

[10] F. Dabek, M.F. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Wide Area Cooperative Storage with cfs," Proc. 18th Symp. Operating Systems Principles (SOSP), pp. 202-215, 2001.

[11] S. Acedan ski, S. Deb, M. Me´dard, and R. Keettor, "How Good Is Random Linear Coding Based Distributed Networked Storage," Proc. First Workshop Network Coding, Theory, and Applications— NetCod, 2005.

[12] C. Gkantsidis and P. Rodriguez, "Network Coding for Large Scale Content Distribution," Proc. IEEE INFOCOM, vol. 4, pp. 2235- 2245, 2005.

[13] R. Ahlswede, N. Cai, S.-Y.R. Li, and R.W. Yeung, "Network Information Flow," IEEE Trans. Information Theory, vol. 46, no. 4, pp. 1204-1216, 2000.

[14] S.-Y.R. Li, R.W. Yeung, and N. Cai, "Linear Network Coding," IEEE Trans. Information Theory, vol. 49, no. 2, pp. 371-381, Feb. 2003.

[15] A.G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Decen- tralized Erasure Codes for Distributed Networked Storage," IEEE Trans. Information Theory, vol. 52, no. 6, pp. 2809-2816, June 2006.

[16] Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, "Proactive Secret Sharing or: How to Cope with Perpetual Leakage," Proc. 15th Ann. Int'l Cryptology Conf.—CRYPTO, pp. 339-352, 1995.

[17] Cachin, K. Kursawe, A. Lysyanskaya, and R. Strobl, "Asyn- chronous Verifiable Secret Sharing and Proactive Cryptosystems," Proc. Ninth ACM Conf. Computer and Comm. Security (CCS), pp. 88- 97, 2002.

[18] R. Canetti and S. Goldwasser, “An Efficient threshold Public Key Cryptosystem Secure against Adaptive Chosen Ciphertext At- tack,” Proc. Int’l Conf. Theory and Application of Cryptographic Techniques, pp. 90-106, 1999.

[19] D. Boneh, X. Boyen, and S. Halevi, “Chosen Ciphertext Secure Public Key Threshold Encryption without Random Oracles,” Proc. Topics in Cryptology (CT-RSA), pp. 226-243, 2006.

[20] P.S.L.M. Barreto, B. Lynn, and M. Scott, “Efficient Implementa- tion of Pairing-Based Cryptosystems,” J. Cryptology, vol. 17, no. 4, pp. 321-334, 2004.

[21] V.S. Miller, “The Weil Pairing, and Its Efficient Calculation,” J. Cryptology, vol. 17, no. 4, pp. 235-261, 2004.

[22] Android Developers ,Available 2010年3月at http://developer.android.com/index.html.

[23] Java.sun.com , Available 2010年3月at http://java.sun.com/.

[24] Eclipse.org home , Available 2010年3月at http://www.eclipse.org/.

[25] Erasure code, Available 2010年3月at http://en.wikipedia.org/wiki/Erasure_code.

[26] Cloud computing , Available 2010年3月at http://en.wikipedia.org/wiki/Cloud_computing

[27] Java Pairing-Based Cryptography Library , Available 2010年3月at http://gas.dia.unisa.it/projects/jpbc/index.html

[28] Hsiao-Ying Lin, Wen-Guey Tzeng, "A Secure Decentralized Erasure Code for Distributed Networked Storage," IEEE Transactions on Parallel and Distributed Systems, 26 Jan. 2010. IEEE computer Society Digital Library.

[29] 佘志龍、陳昱勛、鄭名傑、陳小鳳、郭秩均，Google Android SDK 開發範例大全，第一版，2009年4月出版。

[30] Android - An Open Handset Alliance Project (2010/3/11) , http://developer.android.com/guide/basics/what-is-android.html

[31] Android Open Source Project (2010/3/11), http://source.android.com/

[32] Eclipse Integrated Development Environment (2010/3/11), http://www.eclipse.org/

[33] Codesourcery(2010/3/11), http://www.codesourcery.com/sgpp/lite/arm/portal/subscription?@template=lite

[34] Android Market(2010/3/11), http://www.android.com/market/

[35] Android中文資源站(2010/3/11) http://android.cool3c.com/

[36] Android internals(2010/3/11), http://groups.google.com/group/android-internals?pli=1

[37] Snort(2010/3/11), http://www.snort.org/

[38] XDA developers(2010/3/11) http://forum.xda-developers.com/index.php

[39] INSECURE.ORG, “Top 100 network security tools,”2006.[Online].Available:(2010/3/11) http://sectools.org/

[40] CyanogenMod(2010/3/11) http://www.cyanogenmod.com/

[41] CyanogenModWiki(2010/3/11) http://wiki.cyanogenmod.com/index.php/Full_Update_Guide_-_G1/Dream/Magic32A_Firmware_to_CyanogenMod

[42] System and Internet Infrastructure Security Lab: Understanding Android's Security Framework

[43] Leonid Batyuk, Aubrey-Derrick Schmidt, Hans-Gunther Schmidt, Ahmet Ahmet Çamtepe, Sahin Albayrak: Developing and Benchmarking Native Linux Applications on Android. MOBILWARE 2009:381-392

[44] A.-D. Schmidt, H.-G. Schmidt, J. Clausen, A. Camtepe, and S. Albayrak: Enhancing Security of Linux-based Android Devices. In: Proceedings of 15th International Linux Kongress. Lehman Verlag, Hamburg (2008)

[45] J. Cheng, S. H. Y. Wong, H. Yang, and S. Lu, “Smartsiren: virus detection and alert for smartphones, ”in International Conference on Mobile Systems, Applications, and Services (Mobisys2007), 2007, pp. 258–271.

[46] D. Samfat and R. Molva, “IDAMN: An Intrusion Detection Architecture for Mobile Networks, ”IEEE Journal on Selected Areas in Communications, vol. 15, no. 7, pp. 1373–1380, Sep. 1997.

[47] M. Miettinen, P. Halonen, and K. Hätönen, “Host-Based Intrusion Detection for Advanced Mobile Devices,” in AINA ’06: Proceedings of the 20th International Conference on Advanced Information Networking and Applications - Volume 2 (AINA’06). Washington, DC, USA: IEEE Computer Society, 2006, pp. 72–76.

[48] Dorothy E. Denning: An Intrusion-Detection Model. IEEE Trans. Software Eng. (TSE) 13(2):222-232 (1987)

[49] Alok Tongaonkar, Sreenaath Vasudevan, R. Sekar: Fast Packet Classification for Snort by Native Compilation of Rules. LISA 2008:159-165

[50] P. Kumar Manna, S. Ranka, 及 Shigang Chen, “Analysis of Maximum Executable Length for Detecting Text-Based Malware,” Distributed Computing Systems, 2008. ICDCS '08. The 28th International Conference on, 2008, 頁. 176-183.

[51] Vasudevan 及 R. Yerraballi, “Cobra: fine-grained malware analysis using stealth localized-executions,” Security and Privacy, 2006 IEEE Symposium on, 2006, 頁. 15 pp.-279.

[52] Heping Tang, Shuguang Huang, Yongliang Li, 及 Lei Bao, “Dynamic taint analysis for vulnerability exploits detection,” Computer Engineering and Technology (ICCET), 2010 2nd International Conference on, 2010, 頁. V2-215-V2-218.

[53] Hengli Zhao, Ming Xu, Ning Zheng, Jingjing Yao, 及 Qiang Ho, “Malicious Executables Classification Based on Behavioral Factor Analysis,” e-Education, e-Business, e-Management, and e-Learning, 2010. IC4E '10. International Conference on, 2010, 頁. 502-506.

[54] P. Royal, M. Halpin, D. Dagon, R. Edmonds, 及 Wenke Lee, “PolyUnpack: Automating the Hidden-Code Extraction of Unpack-Executing Malware,” Computer Security Applications Conference, 2006. ACSAC '06. 22nd Annual, 2006, 頁. 289-300.

[55] Willems, T. Holz, 及 F. Freiling, “Toward Automated Dynamic Malware Analysis Using CWSandbox,” Security & Privacy, IEEE, vol. 5, 2007, 頁. 32-39.

[56] Koushik Sen, Darko Marinov, Gul Agha. CUTE: A Concolic Unit Testing Engine for C. ESECFSE 2005.

[57] Cristian Cadar, Vijay Ganesh, Peter M. Pawlowski, David L. Dill, Dawson R. Engler. EXE: Automatically Generating Inputs of Death. CCS 2006.

[58] David Molnar, David Wagner. Catchconv: Symbolic execution and run-time type inference for integer conversion errors. Electrical Engineering and Computer Sciences, University of California at Berkeley, Technique Report No. UCB/EECS-2007-23. http://www.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-23.html

[59] Kuen-Han Huang, Shin-Kun Huang. Detecting Buffer Overflow Vulnerabilities by Search-based Testing. CISC 2010.

[60] Michael Sutton, Adam Greene, Pedram Amini. Fuzzing: Brute Force Vulnerability Discovery. ISBN 0321446119.

行動無線網路安全與惡意程式行為分析
跨國產學合作計畫(國際合作)

出國報告書

指導單位：行政院國家科學委員會
撰寫人員：陳毅睿

日期：中華民國 99 年 5 月 14 日

1、報告摘要

此行的目的有主要是與 U. C. Berkeley 的教授和學生在學術研究上交流請益，並共同撰寫論文。U. C. Berkeley 是美國著名大學，尤其是在電腦科學學門是很傑出的學校。此行得到 U. C. Berkeley 的教授直接指導並且與 U. C. Berkeley 的學生一同學習，彼此在學術上的研究亦得到完整的交流。合著論文已經接近完成，並規劃在短期內進行投稿。

2、行程之目的

與 U. C. Berkeley 的教授和學生在學術研究上交流請益，並共同撰寫論文

3、行程之日期與議程

- March/01：抵達 U. C. Berkeley 並入住 YMCA Hotel
- March/02：申請 U. C. Berkeley Cal ID Card, 門禁系統(Cory Hall)
- March/03：與 Prof. Tygar 第一次會面，討論後來研究進行的方向、時及一些相關行程，Prof. Tygar 並希望我們能整理出有興趣的研究方向給他，以便他能指導我們研究。
- March/04~26：每星期至少與 Prof. Tygar 進行一次為時一小時的討論，主要在確認研究動機與主題以及一些文獻的收集與整理。Prof. Tygar 也針對我們所提出的研究方向進行建議以及一些相關論文的閱讀指導。
- March/27~April/04：U. C. Berkeley 春假週。
- April/05~28：每星期至少與 Prof. Tygar 進行一次為時一小時的討論，主要在討論一些論文的想法。Prof. Tygar 也對我所提出的想法進行修正並給予撰寫以及投稿建議。

— April/29： 搭機返台

4、行程之內容重點摘述

此次行程的時間共計兩個月，目的為學術交流與技術合作，U. C. Berkeley 主要由 Prof. Tygar 負責帶領，所有在 U. C. Berkeley 的 iCast 台灣學生每星期固定與 Prof.Tygar 共同會面一次，回報各別的研究進度與系統發展進度，了解進行中所遇到的種種困難並協助我們克服，提供必要的軟硬體幫助。另外我們一週也與台灣方便的老師共同使用網路電話或電子信箱聯絡並回報狀況一次。學術研究方面重點摘述如下：

本人與 U.C. Berkeley 教授 Prof. Tygar 固定一週會面一次討論研究內容，時間約為一小時，並在本行程中完成一篇論文的改寫以及一篇論文的草稿預計於近期完成並進行投稿。Prof. Tygar 並提供一個辦公室的位置以利我研究以及與其他學生交流。

5、結論與建議

1. **績效評估**

— 學術研究方面，與 U.C. Berkeley 教授合作即有收穫，並合寫論文，將於短期內進行投稿。

— 此行加深對計畫總體內涵與參與研究人員(教授，學生)、顧問、辦公室成員之瞭解，對後續計畫之進行頗有助益。

— 瞭解美國研究生在美研究的方式與生活文化，並學習其研究方法，對後續研究與系統發展有深遠影響。

2. **結論與建議**

此行圓滿達成與 U.C. Berkeley 教授在學術研究上合作撰寫論文。我認為在臺灣校內之研究硬體環境普遍並不比 U.C. Berkeley 差

但空間規劃與學習心態上有眾多不同點。在空間規劃上面，UCB 有眾多開放空間讓學生可以彼此討論，並提供大量白板可供學生使用。學習心態上，美國的研究生較為主動，會自行組成讀書討論會，老師亦鼓勵這樣的學術交流活動，並出資提供餐點。在這次行程讓我瞭解美國研究的方式與生活文化可以拓展我的視野並增長研究的能力與生活中溝通與協調的智慧。主要是學習他們的思考邏輯與應用觸角的延伸，並善用廣博的學術演講機會，可讓學生更加獲益。

行動無線網路安全與惡意程式行為分析

跨國產學合作計畫(國際合作)

出國報告書

指導單位：行政院國家科學委員會

撰寫人員：彭博群

日期：中華民國 99 年 05 月 14 日

出國報告書

一、 參加會議經過

2010年三月一號到四月二十九號為期兩個月,前往美國加州大學柏克萊分校(University of California,Berkeley),與其電機與電腦科學系(EECS)教授Prof.Doug Tygar共同合作做學術研究。期間居住距離柏克萊校園西門不遠處的YMCA青年宿舍,同個宿舍中也有不少在柏克萊就讀的外國博士生以及訪問學者。其中,除了Prof.Tygar有其他要事之外,固定每周一,三,五會與Prof.Tygar在其EECS系館(Soda Hall)針對於個人的研究題目作討論。除此之外,Prof.Tygar在另一個EECS系館(Cory Hall)中的TRUST(Team for Research in Ubiquitous Secure Technology)研究中心裡面,提供了研究室的空間給我們使用。在一般工作日我們固定約八點步行前往Cory Hall進行個人的研究。因考慮到周遭環境在入夜後治安的問題,我們也固定在傍晚六點時返回YMCA青年宿舍。在TRUST中心也認識了西班牙裔博士後研究員Dr.Eladio Martin 以及另外一位華裔研究員Posu Yan。他們除了在我們人生地不熟時幫忙安頓我們,也在研究上面提供了我們很好的意見。除此之外,Prof.Tygar的大學部學生,印度裔Shaan Mulchandani,在Prof.Tygar的安排之下,從四月開始也加入我們的定期meeting。因為Shaan是在美國當地長大以及在柏克萊就讀的學生,因此用不同的角度給我們不同的意見,以及以英文為母語角度的使用者,提供在英文撰寫論文的寶貴意見。

與Prof.Tygar meeting以及研究進度:

3/1 – 3/7 安頓住宿以及其他生活必需,Prof.Tygar提供幫忙以及初次探討研究方向。

3/8 – 3/20 經過與Prof.Tygar討論之後,決定研究主題以及開始寫研究主題需要

的程式。

3/21 – 3/27 柏克萊春假,校園關閉,此週停止與Prof.Tygar meeting。

3/28 – 4/3 繼續撰寫研究主題需要之程式,Prof.Tygar提供修改的意見。

4/4 – 4/17 程式接近完成,與Pro.Tygar討論發表論文的可能性,並且討論細節。

4/18 –4/25 與Pro.Tygar決定論文的方向,並且對於研究的程式作除錯。

4/26 –4/29 因時間不夠,決定論文回台之後繼續研究,並且開始整頓回台灣。

二、 與會心得

這次學術交流除了能夠與世界頂尖的學者面對面作討論,親自體驗大師的風範以及感受到他們對於學術研究的熱情。更難得可貴的是,與柏克萊的學生以及不同國籍的研究員進行交流以及討論。因為與學生以及研究員的交流,能夠提供我們不同的角度去看待研究的內容。除此之外,我們還可以像朋友般的閒聊,一起去吃飯,逛校園。也因次,我們在不同國籍文化的交流下,更碰撞出不同的火花。例如柏克萊的學生 Shaan,分享美國大學生選擇科系的考量,以及評論近日加州政府要增加大學學費的感想。西班牙裔的研究員 Dr.Martin 分享歐洲就讀博士班的心得。華裔研究員 Yan 分享他目前所做的研究,以及向我們請教一些台灣中文俚語(slang)的問題,我們也向他請教美國當地一些俚語的用法,與我們學校學習的英文完全不同,因此在語言方面也得到的許多心得。

三、考察參觀活動

除了柏克萊的校園巡禮,華裔研究員 Yan 熱心的帶我們前往附近的舊金山市區參訪,以及參觀著名的金門大橋(Goldengate bridge)。

TRUST 研究室

與柏克萊學生 Shaan 討論之合影

EECS 系館 Soda Hall

柏克萊校園中最著名的鐘塔

四、建議

雖然台灣的大學規模與美國的大學規模以及預算還是有一段不小的差距,不過他們再針對於某些計畫的經費以及設備都能夠沒有後顧之憂。在台灣必須要被限制在計畫給予經費的期間,如果計畫結束則原本的研究就會被中斷,或是要擔心繼續去申請計畫。也可能因此還沒完成的計畫就此打斷。

另外,在柏克萊的系館可以發現,每個角落都有小小的討論區,提供白板桌椅。很常看到他們的學生就在那裏討論課業。相較之下,台灣的大學除了教室之外,似乎很少這樣的空間。建議系館內可以多設置這樣的空間,讓學生能夠便利舒適的討論課業。

五、攜回資料名稱及內容

無。

六、其他

無。

行動無線網路安全與惡意程式行為分析
跨國產學合作計畫(國際合作)

出國報告書

指導單位：行政院國家科學委員會
撰寫人員：陳柏愷

日期：中華民國 99 年 11 月 12 日

國科會補助專題研究計畫項下赴國外(或大陸地區)出差或研習心得報告

一、國外(大陸)研究過程

此研究開始於 2010/9/1，結束於 2010/10/31，為期兩個月，當我們到達 San Francisco，安頓好住宿與基本生活需求後，便開始與 UC Berkeley 聯絡上。我們在 Berkeley 主要的接觸對象為 TRUST center 執行主任與 computer science department 的 Doug Tygar 教授。

在 TRUST center 這邊安排了我們的座位，我們大部分時間都在那邊工作，我們也跟 TRUST center 的 Executive Director- Larry Rohrbough 針對我們的研究議題做了報告，他對我們的研究很感興趣，互相討論交流意見。從整個回應來看，我們所做的東西是受到認同的，也讓我們更有信心繼續做下去。

另外我們每週四會參加 TRUST 所舉辦的 seminar，會中邀請業界的專家或學界的教授演講，演講的主題以資訊安全為主要方向。在 seminar 當中，我們吸取到目前國外主流資安技術發展情況，得到一些新的研究想法，同時也加強了英文能力。

關於研究主題的部份，我的題目為” New Threat comes with DNSSEC”，就這個題目在去美國之前我們已經有初步的構想，而在與 Prof. Tygar 討論之後，他也覺得這個題目有實作的價值，並且他建議說這個內容並不複雜，如果要發論文的話，動作必須要快，避免其他人先做出了類似的東西，這樣我們就沒辦法公佈。在美國的大部分時間，我就是針對這個題目研究、寫程式、做實驗等等。而在台灣的實驗室有另兩位碩士生協助此研究，我也經常透過網路與台灣討論，並請求部份的實作協助。大約是每一個多禮拜，我會跟 Prof. Tygar 開一次會議，討論階段性的研究成果。而在兩個月之後，我們已完成所有程式與實驗，並完成論文初稿。

二、研究成果

此行主要的研究成果為論文-"New Threat comes with DNSSEC"。此論文探討的是 DNSSEC 技術當中，雖然加強了 DNS 的安全性，並有許多先進的安全性技術，然而其中的一個特性-NSEC3，卻隱含一個可能的安全漏洞。這個 NSEC3 的漏洞，可能造成資訊洩漏的問題，導致使用者的姓名、E-mail 等資訊被揭露，以致有 spam、social engineering attack 等後續的問題發生，本論文證明了這個漏洞確實存在，並提出解決方法。

此論文在出國前僅有基本構想，而經過這兩個月的研究，已完成了架構設計、程式實作、實驗數據等等，並將所有成果歸納整理，成為一篇英文論文。此論文目前所有章節的草稿已全部完成，現階段與 Prof. Tygar 討論修改當中，希望在一個月後能夠定稿，並投稿於國際期刊。

三、建議

此次出國研習的過程，參與多次 Berkeley 所舉辦的 seminar，跟交大校內的 seminar 有些不同，或許可供國內參考。Berkeley 所舉辦的 seminar，公開各界自由參加，不限 Berkeley 校內人士，舉辦的時間為午餐時間，並供簡便午餐予參加人士。因為這樣的設計，我們可以看到參加的人士比較多元，比較多教授參加，並且氣氛相當輕鬆，比較多討論交流。相較於國內的 seminar 氣氛比較嚴肅，大部分是學生在聽講，國外的輕鬆開放是可供參考的作法。

另外此次的經驗，感受到許多文化的洗禮，以 Berkeley 來說，雖然是在美國，包含學生跟教授，幾乎可以看到 1/4 都是亞洲人，整個文化相當多元豐富，同時也更感受到國際化的衝擊；這樣培養出來的學生，到世界各地都會有競爭力。英文能力是國際化的基礎，目前可以看到國內大學已經漸漸增加英語授課，但

距離國際化還有段距離；建議國內大學研究所以上課程盡量以英語授課，並多聘請外國籍教授，多廣納外國籍學生，打造一個國際化的環境。

四、其他

本次的研習相當有收穫，不但加強了英文，拓展了人脈與視野，並做了許多研究，完成一篇論文。如果能持續舉辦的話，將能栽培更多學生，培植國內資安能量。

行動無線網路安全與惡意程式行為分析
跨國產學合作計畫(國際合作)

出國報告書

指導單位：行政院國家科學委員會
撰寫人員：王嘉偉

日期：中華民國 99 年 11 月 12 日

出國報告書

一、 國外研究過程

抵達美國加州並於 Richmond 安頓好住、行問題後數日，便前往 Berkeley 大學 TRUST & CHESS Center 研究室，進行為期兩個月的短期研究。在 Berkeley 指導我們的為 Doug Tygar 教授，首次的會議主要介紹自己所研究的方向，並表示在未來兩個月內希望可以完成系統實做以及論文撰寫。除了繼續進行在國外已完成部份的研究外，我們也與教授討論是否有新的想法可以延伸以開展另外一個議題。

介紹完自己的方向後，教授表示希望看過實際的系統操作以更深入瞭解自己的研究，因此在之後的會議裡也實際 Demo 於國內完成，仍於測試階段的隱匿式惡意程式偵測系統。Demo 完畢後正式進入新的開發階段，提出數個論點以及其他現存相關論文共有的問題解釋為何選定這個研究主題，以及突顯自己論文的貢獻點。經過評估討論後，確定該方向確實有其研究價值，便開始著手思考並設法解決要完成此偵測系統所可能遭遇的問題。

此研究主要是基於虛擬機器內外比對的方法來偵測隱匿式惡意程式，任何程式都需要硬體資源來運行，惡意程式也不例外，利用虛擬機器技術可以輕易的掌控所有的虛擬硬體，從中觀察系統的狀態可觀察到所有需要硬體去運行的軟體，其中當然也包括了惡意程式。因此，不管隱匿式惡意程式用任何手法矇騙系統上層的使用者，在硬體層仍需暴露自己的資料方可利用硬體資源，將實際運行於硬體上的資料跟系統上層使用者對系統的觀察兩相比較，若存在差異，表示惡意程式介入其中並隱藏部份系統訊息，為目前偵測隱匿式惡意程式的主流方法之一。

為了實做此系統並彌補過去相關論文的不足，與教授討論的兩個重點為語意重建問題以及為自己的偵測系統加設保護機制。在克服上述兩個重點後，進一步討論如何去評估此偵測系統，不應只是與其他現存的偵測軟體比較，因可能存在不公平的因素，自己的偵測系統為後生的系統，比較上來說，較佔優勢。因此，更有特色的評估比較以突顯自身系統的價值、貢獻是需要的，這部份的討論則持續到回國至今。

二、 研究成果

克服上述兩個主要問題，完成第一版的隱匿式偵測系統實做，並完成論文草稿，完成章節為摘要、介紹、相關作品、設計理念與架構以及實做細節，文章內容持續修飾中，以期明確帶出此系統的學術價值，系統評估也還在設計階段，目標為達到突顯下列特點：

1. 此偵測系統可有效偵測未知的惡意程式
2. 相較於以往相關研究，可避免語意重建問題
3. 保護機制的有效性

三、 建議

往後若能與國外的團隊一起合作，相信學習成效會更加明顯，由於此行主要合作對象僅有 Tygar 教授一人，平時鮮少與其研究團隊接觸，為較可惜之處。國外的 Seminar 則應盡量參加，除了可以增進自己對國內外研究的瞭解，國外 present 時的高互動性相信也是可學習的重點之一。

四、 其他

此行實為一非常特殊的經驗，在國外解決食衣住行等問題，第一次真正以英文跟他人對應交談；體驗過國外的校園生活、見識並比較過國內外研究，對於自己未來的研究之路也會產生新的看法，希望未來也能有如此不可多得的機會前往不一樣的環境學習新事物。

出席 2010 國際密碼會議(CRYPTO 2010)報告

陳毅睿

一、時間與地點：99/8/16~8/19，U. C. Santa Barbara

二、論文：203 篇投稿，接受 39 篇

三、參加會議經過：

第 30 屆國際密碼會議（The 30th Annual Cryptology Conference，簡稱 CRYPTO 2010）為國際密碼研究會（International Association for Cryptologic Research，簡稱 IACR）主辦，今年的會議在美國加州大學的聖塔芭芭拉分校舉行，與會人數共約 350 人。會議為期五天（8/16~8/19），除了第二天以及第五天下午外，其餘皆為論文發表時間。第一天的早上有大師級的學者受邀演講，講者分別有 Shafi Goldwasser（MIT and Weizmamnn）以及 Silvio Micali（MIT），講題是 Zero knowledge – 25 Years 。在會議會場內的討論氣氛非常熱烈，休息時間都可看到許多學者在相互交流與討論。

四、發表論文介紹：

無論文發表。

五、與會心得：

本次會議共有 39 篇論文發表，每篇發表的時間為 25 分鐘，分為 12 個 sessions:

Day 1 (8/16)

Session 1: Leakage

Session 2: Lattice

Session 3: Invite Talk

Session 4: Homomorphic Encryption

Session 5: Theory and Applications

Day 2 (8/17)

Session 6: Key Exchange, +OAEP/RSA, CCA

Session 7: Attacks

Day 3 (8/18)

Session 8: Composition

Session 9: Computation Delegation & Obfuscation

Session 10: Multiparty Computation

Day 4 (8/19)

Session 11: Pseudorandomness

Session 12: Quantum

我們也從中場休息時間的互動中獲得許多研究上寶貴的想法。會議中由其他講者的報告中我們也獲得許多的啟發並得以掌握到最近的學術動態。以下就針對幾篇我們有興趣的論文來做一些簡單的介紹:

(1) Zvika Brakerski and Shafi Goldwasser, "**Circular and Leakage Resilient Public-Key Encryption under Subgroup Indistinguishability (or: Quadratic Residuosity Strikes Back)**"

Public key encryption (PKE) scheme 的設計在密碼學中是一項很重要的研究主題。近幾年來在討論並證明 PKE scheme 的安全性時，除了原先所常見的 semantically secure 之外，更進一步的會考量以下幾個在現實應用上可能所需保證的安全面向:

(a) Key-dependent message (circular) security. 證明 PKE scheme 的 semantic security 時，一般只能保證在加密攻擊者有辦法得知的明文訊息時的安全性。但是在實際的應用中，明文的訊息有可能包含了系統中的 secret key，而這些 secret key 在證明 semantic security 時卻是假設為攻擊者無法得知的訊息。

(b) Leakage resiliency.在現實的系統運行中，secret key 以某些方式洩露是有可能的，像是攻擊者可能會將使用者解密時的記憶體狀態記錄下來，然後從中得到使用者的解密金鑰(cold boot attacks)等方式。所

以在證明一個 PKE scheme 的安全時，也應考慮到系統 secret key 會被攻擊者得知的情形。

(c) Auxiliary-input security.在證明一個 PKE scheme 的安全性中，攻擊者除了可以從系統公開訊息中試著取得系統私密資訊之外，可以更進一步地推廣成攻擊者還有辦法得知 f(secret key,public key)，其中 f 是攻擊者任意選定的一個函數。在這樣增強攻擊者能力的 PKE scheme 所證明的安全性的即是 auxiliary-input security。

在這篇論文中，作者提出了一個 PKE scheme，除了滿足一般常見的 semantic security 外，更同時滿足了上述的三種安全性。而這篇論文中所提出的 PKE scheme 是基於 quadratic residuosity (QR) assumption (或是 Paillier's decisional composite residuosity (DCR) assumption)。

(2) Ali Juma and Yevgeniy Vahlis, **"Protecting Cryptographic Keys against Continual Lekage"**.

在一般現實中應用的加密系統中，side-channel attack 是攻擊者常會使用的一類攻擊法，其主要的精神是藉由記錄系統運行時的各種狀態來進行系統私密資訊的分析及偷取，像是 cold boot attacks。所以在設計加密演算法時，需要將這類攻擊考慮到安全的 model 之中，而以往的加密系統在證明其安全性時大多都是假設系統在運行時，私密的資訊是不會洩露的，但在現實中攻擊者在系統運行中利用 side-channel attack 來獲得系統私密的資訊卻是有可能的。而一個能被證明抵擋 side-channel attack 的加密系統可稱為 leakage-resilient scheme。

在這篇論文當中，作者們提出一種利用"fully homomorphic encryption with re-randomizable ciphertexts"來保護系統中的私密資訊免於受到 side-channel attack 的方法。其中作者們在證明安全性時是假設系統只有在進行運算時，才有可能讓攻擊者利用 side-channel attack 來偷取私密資訊，像是 cold boot attack 就是藉由使用者在解密時，會將解密金鑰 load 到記憶體中才會讓攻擊者有機會將當時的記憶體狀態記錄下來並進行分析，而其他時候使用者的私密金鑰都是假設在系統中被安全地保護著。

(3) Shafi Goldwasser and Guy N. Rothblum, **"Securing Computation against Continuous Leakage"**.

在現實的密碼系統中，攻擊者可利用 side-channel attack 來偷取系統私密

資訊。近幾年來，在證明一個密碼系統的安全性時，都會將這類的攻擊放入安全性 model 中考量分析，而滿足這種安全性的這類的密碼演算法可稱之為 leakage-resilient cryptographc algorithms。

在這篇論文當中，作者們提出了一種通用的方法，可將現有的任何密碼演算法轉換成可抵擋 side-channel attack 的密碼演算法。而他們的安全性是假設系統只有在進行運算時，才有可能讓攻擊者利用 side-channel attack 來偷取私密資訊，像是 cold boot attack。作者們使用了一個 semantically secure 的加密演算法配合上 key refreshing, oblivious generation of cipher texts, leakage resilience re-generation, and blinded homomorphic evaluation of on single complete gate (e.g. NADN)。作者們在這篇論文中也同時展示了一個利用他們所提出的方法造出的加密演算法(在 DDH assumption 下)。

(4) Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan, “**i-Hop Homomorphic Encryption and Rerandomizable Yao Circuits**”.

Homomorphic encryption 在現實中有著相當多的應用，像是解決銀行問題: 假設在銀行存錢，而所有的關於財產的資料都是經過加密才儲存的，目的是不希望銀行知道任何有關財產的資料，但在存錢或領錢時，銀行要如何對加密的財產資料做增減的運算呢? 一個簡單的方法就是銀行先解開密文，做完運算後再將訊息重新加密回去。不過在運算期間銀行其實已經知道有關財產的資料了，所以一般希望直接以密文的方式對財產做增減的運算，這樣便能保護客戶的財產資料了。而解決這類問題的方式就是使用”homomorphic encryption”。

在 2009 年，Craig Gentry 提出了第一個 fully homomorphic encryption scheme (using ideal lattices)，雖然該 scheme 離實際應用還有著一段距離，但其解決了 Rivest, Adleman and Dertouzos 在 1978 年提出的 open problem，也使得現實中很多的安全性應用出現了希望。

在這篇論文中，作者更進一步地提出了一個 fully homomorphic encryption scheme，其可改進先前 scheme 中加密資料只能進行一次 homomorphic 進算的缺點，使其從 single-hop 變成了 multi-hop 的 fully homomorphic encryption scheme。不過這篇論文中所提出的 multi-hop fully homomorphic encryption scheme 的加密資料長度卻會隨著轉換資料成長 (complexity: $n^{0(i)}$)。

六、建議：

這次會議看到許多很優秀的論文發表，許多學者在論文的撰寫上面都很有經驗，各種安全性的定義及證明都寫得非常正式與嚴謹，這是國內密碼學者所該學習的。在密碼的論文中，除了要能夠有好的構想之外，如何證明該構想的可行性

與安全性非常重要，同時也是一個基礎的功力。以現在的環境來說，要能夠在較好的國際會議中發表的論文，一定要有著嚴謹的定義和詳盡的證明。而在密碼學的領域當中，好的國際會議往往是各國學者注目的焦點，發表在這些會議上，才會受到大家的重視。因此，我們也應該盡量發表在重要的國際會議上，才可以真正提升我國的密碼學研究層次。另外，各國學者在會議期間積極的討論與交流的態度，是我們應該學習的。若有足夠的經費補助，無論是學生或是教授，都應該能多出國吸取國外學者的研究經驗。

七、攜回資料名稱與內容：

(1) CRYPTO 2010 會議資訊: 記載本次研討會的時間、地點、會議流程、報告人員、以及報告題目等與研討會相關的會議資訊。

(2) CRYPTO 2010 會議論文集 (proceeding): 收錄本次研討會所接受論文之全文記錄。

出席 INSCRYPT 2010 國際會議報告

曾文貴

一、時間與地點：2010/10/20~24，中國上海

二、論文：135 篇投稿，接受 49 篇(36 regular + 13 short papers)

三、參加會議經過

第六屆中國資訊安全與密碼會議（The 6th China International Conference on Information Security and Cryptology，Inscrypt 2010）為中國科學院密碼重點實驗室（State Key Laboratory of Information Security）主辦，今年在上海交通大學的學術會議中心舉行，大約有 100 多位參加，其中來自國際的約有 30 於人，其餘為中國的學者，研究人員與學生。我是本次會議的議程委員之一，因此參加了這次會議，除了主持 session 之外，也和上海交大從事資訊安全研究的教授交流,其中包含來學嘉教授。主要會議為期三天 (10/21~23)，除了第二天晚上安排歡迎晚宴外，其餘皆為論文發表時間。這次會議邀請了兩個專題演講：

- “Perspectives on lightweight cryptography”, Bart Preneel
- “Public key cryptosystems with key-dependent message security”, Moti Yung

五、與會心得：

本次會議共有 49 篇論文發表，每篇發表的時間為 20-25 分鐘，分為 12 個 sessions:

Day 1 (10/21)

Session 1: Invited Talk I:
Perspectives on lightweight cryptography

Session 2: Encryption schemes

Session 3: Stream ciphers, sequences and elliptic curves

Session 4: Public key and elliptic curve cryptography

Day 2 (10/22)

Session 5: Invited Talk II:

"Public key cryptosystems with key-dependent message security"

Session 6: Hash functions

Session 7: Key management

Session 8: Cryptographic constructions

Day 3 (10/23)

Session 9: Digital signature (I)

Session 10: Privacy and algebraic cryptanalysis

Session 11: Digital signature (II) and authentication

Session 12: System security

六、心得與建議：

這次會議看到許多不錯的論文發表，尤其是一些大陸學者及學生的論文，比以往有很大的進步，他們報告時，有些人的英文很流利，也有些雖然不是很好，但可以看出他們很努力在學習。

會議期間，我特別早起來看看上海交大學生上課的情形，大約七點半到八點之間，一大堆學生騎自行車來，還有走路的，8 點之後就很少有學生在外面，可見他們都蠻遵守上課的時間，我也到教室外面看他們上課的情形，上課很專心。相對我們交大學生的生活和學習態度，真是感到憂心。

出國短期訪問報告書

撰寫時間： 99 年 2 月 6 日

報告人： 交通大學謝續平

一、出國目的：

本次出國目的為執行國科會國際合作研究計畫「行動無線網路安全與惡意程式行為分析跨國產學合作計畫（國際合作)」，參與柏克萊加州大學 TRUST (The Team for Research in Ubiquitous Secure Technology) Center 大型研究計畫，撰寫研究論文 “Proactive Distributed Digital Evidence Preservation with Tamper Resistance, Perfect Secrecy, and High Survivability”，該論文已經投稿 IEEE JSAC，另外一篇 “Tampering-Resistant Evidence Preserving for Digital Forensics,” 正在 Professor Doug Tygar 修改中。此次也有幸加入美國最新成立在柏克萊加州大學成立的 TRUST Center，促進國際合作。拜會柏克萊加州大學電機資訊系系主任、工學院院長 Shankar Sastry 教授、電機資訊系(EECS)教授同時也是中央研究院院士 Earnest Kuh（葛守仁)、與該校電腦安全權威同時也是美國國防部安全與隱私研究群主席 Doug Tygar、David Wagner 教授等。此次經由國科會國際合作計劃派遣四位研究所學生赴柏克萊加州大學成立的 TRUST Center 短期研究，本人與該校教授 Doug Tygar 熟識多年，

此次短期訪問除為了學生出訪前往該校做事先安排外（例如辦公空間、指導教授、研究主題、住宿、使用設備），並且 藉此機會與 Professor Tygar 與其學生共同研討，意見交流，撰寫論文。

二、參訪期間：

短期研究訪問期間為九十九年一月二十一日至九十九年二月六日止，於期間內赴美國舊金山柏克萊加州大學電機與資訊學系國際合作短期訪問，

三、出國人員：

謝續平現任交通大學資訊工程系教授暨 TWISC@NCTU(Taiwan Information Security Center at NCTU)主任，曾任交通大學資訊工程系系主任、交通大學計算機與網路中心主任、中華民國資訊安全學會理事長，現在擔任 IEEE Tran. On Dependable and Secure Computing、IEEE Trans. On Reliability 副編輯（Associate Editor）、IEEE RS Newsletter 總編輯(Editor-in-Chief)、IEEE Reliability Society Ad Com Member（議會委員）。同時擔任 ACM Symposium on Information, Computer and Communications Security（ASIACCS） 推動委員會主席（steering committee chair），2010 年並獲選為 ACM Distinguished Scientist（美國 ACM 卓越科學家）。

四、參訪經過及重要結果：

UC-Berkeley 電機、資訊領域排名通常都排在全世界前三名，為世界頂尖一流大學，短期訪問不僅和 UC Berkeley 電機與資訊系所與相關研究中心的堅強研究群互動，還因此與全美其他來 UC Berkeley 短長期訪問的學者互動，受益良多。

UC Berkeley 已經與本校交通大學簽訂合作協定，兩校未來將加強教授與學生交流與互訪。為進一步促進兩校實質合作，職和該校 Professor Tygar 共同提出本合作研究計畫，加大 Professor Tygar 向美國 National Science Foundation（NSF）提出申請，職向國科會（NSC）提出。

感謝國科會此次大力協助，支持此次國際合作計畫，使得職經由此次研究合作擴大國際視野，能與世界一流頂尖學者互動，相互學習。經由此次短期訪問得以落實本國際研究合作計畫合作， 交流兩校研究成果，實地瞭解系統開發的 know how，撰寫研究論文“Proactive Distributed Digital Evidence Preservation with Tamper Resistance, Perfect Secrecy, and High Survivability”，該論文已經投稿 IEEE JSAC，另外一篇“Tampering-Resistant Evidence Preserving for Digital Forensics,” 正在 Professor Doug Tygar 修改中。

執行本計畫也有帶來一些其他的國際合作效益，職在本計畫執

行期間擔任 IEEE Tran. On Dependable and Secure Computing、IEEE Trans. On Reliability 副編輯（Associate Editor）、IEEE RS Newsletter 總編輯(Editor-in-Chief)、IEEE Reliability Society Ad Com Member（議會委員）。同時擔任 ACM Symposium on Information, Computer and Communications Security （ASIACCS） 推動委員會主席（steering committee chair），2010 年並獲選為 ACM Distinguished Scientist （美國 ACM 卓越科學家）。

為瞭解美國網路科技學術研發與技術發展現況，特別安排此次實地參訪活動，經由密集的實地參訪，深切的瞭解到美國對軟體技術的重視與投入，尤其在重點大學投資更是驚人，目前美國重點大學與產業之合作明顯超越台灣重點大學，且經由校內設立研究中心，使得學術研究已經與產業所需密切的結合，大學的研究真正的達到提升產業技術的目的，例如柏克萊加州大學設立 CITRIS (Center for Information Technology Research in the Interest of Society)，其下設立許多研究中心，例如 TRUST Center 為美國 NSF、DHLS 所獎助設立，由 UC Berkeley 領導，參與學校包括 Stanford University, Carnegie Mellon University, Cornell University, Vanderbilt University, 充分展現群體卓越的成果。

美國目前電腦網路頻寬已經遠超過台灣，網路電話也較台灣更

為普遍，而全世界一流的高科技公司也紛紛在各重點大學設立研發中心，對產業技術發展具有催化作用，可以預見美國將在軟體技術持續保持領先。台灣過去教育頗為成功，創造了經濟奇蹟，現在因應大陸強力的競爭，台灣亦應大力投資教育，厚植我國高科技基礎。

國科會補助計畫衍生研發成果推廣資料表

日期:2011/01/25

國科會補助計畫	計畫名稱: 行動無線網路安全與惡意程式行為分析跨國產學合作計畫（國際合作）		
	計畫主持人: 曾文貴		
	計畫編號: 98-2218-E-009-020- 學門領域: 資訊安全		
研發成果名稱	（中文）行動平台安全防護與檢測方案		
	（英文）		
成果歸屬機構	國立交通大學	發明人（創作人）	曾文貴,謝續平,黃育綸
技術說明	（中文）本計畫已完成一套行動平台安全防護與檢測方案，包含以下四個子項目： ●無線異質網路模擬平台 本平台可支援涵蓋16節點的網路實驗。使用者可在此無線異質網路模擬平台上，進行異質性無線網路系統的安全測試，以模擬可能的異質網路拓樸與可能存在的攻擊手法。本系統亦與UC Berkeley所開發之有線網路測試平台DETER相容。 ●惡意檔案文件分析系統 本系統利用虛擬CPU及記憶體動態分析技術，可偵測出目標文件是否有夾帶惡意程式或不當獲取系統資訊的意圖。同時本系統亦可過濾各類不正常字串，並找出該惡意程式利用何種手法來達到攻擊目的。 ●程式漏洞檢測系統 本計畫針對知名自由軟體Catchconv進行改良，已開發完成的程式碼迴圈處理機制能偵測重複且可省略的動作，透過將計算能量集中在不重複的分析上增加其執行效率。 ●行動平台安全管理機制 本機制包含一套入侵偵測系統以及一套安全雲端儲存系統。其入侵偵測系統為Android上的第一套入侵偵測系統，並已針對執行速度、耗電量、與封包抓取率等方面進行最佳化，以確保於Android上保有高度的可用性。其安全雲端儲存系統將使用者資料加密後利用Decentralized Erasure Code的方式分散在雲端儲存上，可達到高度安全性及可靠性。		
	（英文）This project proposes a Security Insurance and Inspection Solution for Mobile Devices. The solution includes the following four sub-systems: 1. A heterogeneous network emulation testbed with wired and wireless networks: Users can emulate heterogeneous network topologies with the possible network attacks. 2. A detection system against malicious document files: Using virtual CPU and dynamic memory analyzing technology, this system can detect if the target file contains malicious codes or tries to obtain sensitive system information. 3. A program defects detection system: This project improved Catchconv which is a famous open source tool. The improvement is focused on a loop avoiding algorithm. 4. A security management mechanism for mobile devices: This mechanism includes an IDS (Intrusion Detection System) on Android and a secure cloud storage system to improve the security management for mobile platforms such as smart phones.		
產業別	其他專業、科學及技術服務業		
技術/產品應用範圍	無線網路安全、行動資安		
技術移轉可行性及預期效益	由於行動平台與無線網路的廣泛使用是未來的主要趨勢，相關的安全技術與服務一直是國內外各大網通廠商努力的目標。本計畫所提出之技術內容涵蓋行動平台與無線網路領域之先進安全議題，且依據本計畫之技術成果，計畫團隊已分別與友訊科技及中華電信簽訂三項委託研究計畫，足可證明其技術內容具高度的轉移可行性與產業效益。		

註：本項研發成果若尚未申請專利，請勿揭露可申請專利之主要內容。

98 年度專題研究計畫研究成果彙整表

計畫主持人：曾文貴	計畫編號：98-2218-E-009-020-
計畫名稱：行動無線網路安全與惡意程式行為分析跨國產學合作計畫（國際合作）	

成果項目			量化			單位	備註（質化說明：如數個計畫共同成果、成果列為該期刊之封面故事...等）
			實際已達成數（被接受或已發表）	預期總達成數(含實際已達成數)	本計畫實際貢獻百分比		
國內	論文著作	期刊論文	0	0	100%	篇	
		研究報告/技術報告	0	0	100%		
		研討會論文	6	4	70%		
		專書	0	0	100%		
	專利	申請中件數	0	0	100%	件	
		已獲得件數	0	0	100%		
	技術移轉	件數	0	0	100%	件	
		權利金	0	0	100%	千元	
	參與計畫人力（本國籍）	碩士生	17	17	70%	人次	
		博士生	4	4	70%		
		博士後研究員	0	0	100%		
		專任助理	2	2	80%		
國外	論文著作	期刊論文	0	0	100%	篇	
		研究報告/技術報告	0	0	100%		
		研討會論文	1	1	70%		
		專書	0	0	100%	章/本	
	專利	申請中件數	0	0	100%	件	
		已獲得件數	0	0	100%		
	技術移轉	件數	0	0	100%	件	
		權利金	0	0	100%	千元	
	參與計畫人力（外國籍）	碩士生	0	0	100%	人次	
		博士生	0	0	100%		
		博士後研究員	0	0	100%		
		專任助理	0	0	100%		

其他成果 (無法以量化表達之成果如辦理學術活動、獲得獎項、重要國際合作、研究成果國際影響力及其他協助產業技術發展之具體效益事項等，請以文字敘述填列。)	以下論文於CISC 2010 榮獲 Best Student Paper Award Chia-Wei Hsu, Shiuhpyng Shieh, ’’’’ FREE: A Fine-grain Replaying Executions by Using Emulation,’’’’ The 20th Cryptology and Information Security Conference (CISC 2010), Taiwan, 2010.

	成果項目	量化	名稱或內容性質簡述
科教處計畫加填項目	測驗工具(含質性與量性)	0	
	課程/模組	0	
	電腦及網路系統或工具	0	
	教材	0	
	舉辦之活動/競賽	0	
	研討會/工作坊	0	
	電子報、網站	0	
	計畫成果推廣之參與（閱聽）人數	0	

國科會補助專題研究計畫成果報告自評表

請就研究內容與原計畫相符程度、達成預期目標情況、研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性）、是否適合在學術期刊發表或申請專利、主要發現或其他有關價值等，作一綜合評估。

1. 請就研究內容與原計畫相符程度、達成預期目標情況作一綜合評估

■達成目標

□未達成目標（請說明，以 100 字為限）

□實驗失敗

□因故實驗中斷

□其他原因

說明：

2. 研究成果在學術期刊發表或申請專利等情形：

論文：■已發表 □未發表之文稿 □撰寫中 □無

專利：□已獲得 □申請中 ■無

技轉：■已技轉 □洽談中 □無

其他：(以 100 字為限)

3. 請依學術成就、技術創新、社會影響等方面，評估研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性）(以 500 字為限)

本計畫針對計畫內之各子項皆提出系統建置方案，且其理論內容經整理已發表為國際期刊論文。同時本計畫亦與美國加州大學柏克萊分校進行交流研究，研發出無線異質網路模擬平台、惡意文件檔案分析系統、程式漏洞檢測系統、與行動平台安全管理機制等四套雛型系統，並在技術上具有創新與優勢性，且部分研究成果已與中華電信及友訊科技等國內大廠簽訂產學合作計畫。此外，本計畫更於 0penFoundry 開啟專案並公布部分系統之原始碼，相信能將安全技術研發與自由軟體社群連結，並引入自由軟體社群的研發能量以吸引產業界投入的意願、厚植國內資訊安全技術的研發潛力。