行政院國家科學委員會專題研究計畫 成果報告

新型網路攻擊、風險評估與入侵追蹤的誘捕預警技術
研究成果報告(完整版)

計 畫 類 別 ：個別型
計 畫 編 號 ：NSC 98-2623-E-009-001-D
執 行 期 間 ：98 年 01 月 01 日至 98 年 12 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：謝續平
共 同 主 持 人：楊武、許富皓、趙禧綠、黃俊龍、彭文志
計畫參與人員：碩士級-專任助理人員：林真如
學士級-專任助理人員：卓季榆
碩士班研究生-兼任助理人員：黃曜志
碩士班研究生-兼任助理人員：陳韋任
碩士班研究生-兼任助理人員：鄭有倫
碩士班研究生-兼任助理人員：顏志豪
碩士班研究生-兼任助理人員：蔡天浩
碩士班研究生-兼任助理人員：朱信儒
碩士班研究生-兼任助理人員：張書綸
碩士班研究生-兼任助理人員：游釗俊
碩士班研究生-兼任助理人員：李政輝
碩士班研究生-兼任助理人員：張翔任
碩士班研究生-兼任助理人員：趙梨華
碩士班研究生-兼任助理人員：張景旭
碩士班研究生-兼任助理人員：廖政博
碩士班研究生-兼任助理人員：詹智涵
碩士班研究生-兼任助理人員：劉人僖
碩士班研究生-兼任助理人員：楊濬仲
碩士班研究生-兼任助理人員：林育任
博士班研究生-兼任助理人員：沈柏曄
博士班研究生-兼任助理人員：林佳潤
博士班研究生-兼任助理人員：林佳純
博士班研究生-兼任助理人員：王繼偉

博士班研究生-兼任助理人員：李秉翰
博士班研究生-兼任助理人員：許家維
博士班研究生-兼任助理人員：廖忠訓
博士班研究生-兼任助理人員：尤昱婷
博士班研究生-兼任助理人員：葉羅堯
博士班研究生-兼任助理人員：邱士銓

處 理 方 式 ：本計畫可公開查詢

中　華　民　國　99 年 03 月 29 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

新型網路攻擊、風險評估與入侵追蹤的誘捕預警技術

計畫類別：■ 個別型計畫　□ 整合型計畫
計畫編號：NSC 98－2623－E－009－001－D
執行期間：　98 年　1 月　1 日至　98 年　12 月　31 日

計畫主持人：謝續平
共同主持人：楊武、許富皓、趙禧綠、黃俊龍、彭文志
計畫參與人員：

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

本成果報告包括以下應繳交之附件：
□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
□出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、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

執行單位：國立交通大學資訊工程學系

中　華　民　國　99　年　3　月　25　日

摘要

本計劃為期兩年，此報告為第二年度的成果報告，今年度計畫主要針對計劃第一年所開發系統進行發展，並於今年度完成各系統的開發，實現兩大資安議題－「Windows 誘捕網情蒐之核心技術與回追技術軟體發展」與「風險分析與威脅預警之核心技術與軟體單元研發」－之目標：(一)利用誘捕網系統記錄攻擊者入侵行為，有效偵測新型網路攻擊。(二)收集區域網路內防火牆日誌，進行網路風險分析，偵測可疑主機並進行預警。此外，為有效運用協調各系統之功能性，於今年度我們完成中央監控系統的開發，使得誘捕網系統、網路風險分析及預警系統和封包標記追蹤系統可以完善的溝通，將各系統收集到資訊呈現在此監控中心，提供適當的指揮機制，即時因應網路上的攻擊。

兩大議題目標皆針對目前網路安全急需解決之問題。首先，面臨每年不斷成長的系統漏洞，使得各式各樣的新式病毒或攻擊手法產生，雖然傳統防禦機制在防禦已知的攻擊行為有良好表現，但是在面對新型攻擊行為時，偵測與防堵能力均十分有限。為了補強傳統防禦機制的不足，我們針對網路應用程式漏洞提出議題一———Windows 誘捕網情蒐之核心技術與回追技術軟體發展，在第一年完成之誘捕網系統雛型，進一步開發加入防護等級提升機制。利用不同的防護等級給予不同程度的入侵阻力，使攻擊者以為網站管理者已對漏洞做修補，誘使攻擊者使用進階方式攻擊網站，使攻擊者不易察覺落入了誘捕網中。為了立即追蹤查詢攻擊者位置，防範 IP 假冒行為。本系統整合了封包標記追蹤功能，進行攻擊路徑追蹤，讓攻擊者無所遁形。

然而，除了與網路應用程式相關的漏洞，許多區域網路內電腦主機或伺服器本身也隱藏著未知弱點，同樣會遭攻擊或利用。於是，我們提出了議題二———風險分析與威脅預警之核心技術與軟體單元研發。在此研究議題中，本系統整合了網路風險及分析主機，讀取使用者的防火牆日誌，運用網路風險預警分析技術計算出可疑主機，進一步修補漏洞以預防攻擊。今年度我們加強第一年發展出的

雛形，擴大成 2-tier 架構以方便管理並減少網路頻寬使用，並且運用資料倉儲技術加速資料庫查詢，減少 20%~40%的查詢時間。此外，我們加入法則及失誤樹編輯器，提供一客製化系統，同時我們也加入失誤樹風險量化分析技術，利用失誤樹計算主機之風險值，協助管理者偵測出可疑主機。本系統亦和封包標記追蹤系統整合，可預防 IP 的偽冒攻擊。

本計畫投入了大量研究人員進行理論研究與實務開發，每月至少與中科院研究人員團隊面對面研究討論一至兩次，研究與開發成果達成計劃預期目標以及工作項目，計畫成果除了論文的產出以及技術文件的撰寫，亦包含先進的技術研究與實現，共計二十件論文、二十二件技術文件、可供轉移的十四項技術以及十九件的系統軟體。本計劃系統的設計均開發成為雛形系統，能夠符合中科院實際的需求，所開發雛形系統以實際網路攻防進行功能測試，確保能偵測與防禦網路攻擊行為。此外，本計劃開發之技術及系統已全數轉移至中科院，並於小型區域網路內試運成功，可望未來於中科院正式上線運行。

關鍵字：誘捕網、風險評估、入侵追蹤

Abstract

This is a two-year project, and we will sum up results of the second year in this report. In the second year we mainly developed phototypes of systems that we proposed in the first year to ahieve two goals of information security issues- “The Research and Development of Honeynet with IP Traceback” and “A profile-based network security remote monitoring system”. The one goal is to use honypot attract attacks and record their actions. The other one goal is to analize the threats of computers in local network. For effectively managing these systems, we developed a suit of system, called Monitor and Control Center. It can make these systems communicate well and provide administrators with better control.

Two issues all aim at the urgent problems of imformation security. First, more and more vulnerabilities appeared, so hackers can use a variety of new methods to attack others’ computers. It will overload the traditional IDS in the future. To make up the weakness of traditional IDS, we propose the issue one- The Research and Development of Honeynet with IP Traceback. We set up the honeynet system to attract hackers to invide, and record their actions. Once we know their action, we can defense their attacks. In the second year of the plan, we improve the phototype of system, designing different security levels to gain the ability of attraction. Furthermore, this system was integrated with the system of IP Traceback. It can trace the hackers’ IP, and prevent IP spoofing. The system we developed assit the traditional IDS to achieve effective defense.

However, except vulneribilities of Web applications, many PC or web severs may also have lots of vulneribilities. Again we proposed another solution to solve the problem. We construct a system to read firewall data of computers, and analize network security threat to avoid attacks. In the second year, we promote the first first

years's system. Firstly, we expand the system to 2-tier construction system, and it can analysis the security threat more effectively. In addition, the system now can allow administrators to adjust imformation security principles, and control computers' behavior in the local network. Finally we applied the technique of error tree to compute the prability of threat PC. Also this system was integrated with the IP Traceback system to avoid IP spoofing.

This project occupied a lardge number of researches, and we finished all proposed action items. We also have fruitfull research results in the field of imfornation security. Not only did we complete practical implement, but we also tested these systems by real attacks. These systems can really defense attacks. We expect that we can apply these technologies to the field of information security.

Keyword: Honeynet, Security Information Management, Traceback

目錄

圖目錄

表目錄

第一章、計畫簡介

本章節中，分別敘述了計畫緣起、計畫時程、計畫目標及計劃工作項目。並於之後的章節詳細敘述本計畫實作之系統。

第一節、計畫緣起

隨著網路應用日趨複雜，許多 Internet 經濟犯罪也隨之興起，而各政府與金融機構更是網路駭客垂涎的攻擊標的。網路的便利性與透通性則成了資訊安全防護上的漏洞，因此，資訊安全防禦除了需要基本的防火牆與防毒軟體之外，針對偽裝成各式合法通訊協定的攻擊與層出不窮的後門弱點來說，入侵偵測與防禦將是建立更完整的資安解決方案的第一步。

傳統的防火牆僅能被動地就各式預定好的規則進行攔阻或放行網路流量，而這些規則僅能就 TCP 或 UDP 協定的不同服務埠來辨別網路流量的好壞，就如同銀行門口的警衛僅能用客戶是否頭戴安全帽或是手持尖刀來判斷客戶是否有意行搶。

而在現今的網路環境中，駭客攻擊的手法往往隱藏於合法的服務中，例如曾在全球造成無數網路癱瘓的 SQL Slammer 攻擊，它攻擊的手法就隱藏在合法的 Microsoft SQL Server 所使用的 UDP 1434 埠中。如果使用傳統式的防火牆，固然可以定下規則阻絕 UDP 1434 埠來避免 SQL Slammer 的攻擊，但同時也會因為該服務埠被阻擋而造成 SQL Server 無法使用。

(1)Windows 誘捕網情蒐之核心技術與回追技術軟體發展

至於目前 IDS 與 IPS 上入侵偵測技術的限制，目前網路型與主機型入侵偵測系統普遍有下列潛在不足之處：

一、只能偵測出已知的攻擊模式：

以比對特徵為基礎（Signature-based）的安全機制只能辨識出資料庫中有相對應的攻擊特徵之非法行為，所以攻擊特徵（Signature）的 開發速度會影響安全機制的有效性。以比對特徵為基礎（Signature-based）的入侵偵測系統在 1990

年中期成形，當時已知的安全漏洞數目並不 多。根據 CERT 的資料，在 1995 年，只有 171 個安全漏洞被發佈，Signature-based 的資訊安全產品只要每月更新增加 10-12 個攻擊特 徵（Signature）， 便足以應付當時的安全需求。

隨著公佈的安全漏洞的數目不斷快速增加，加上變形攻擊（Mutations）的出現，上述情形已逐漸改觀。針對每一種不同的攻擊，Signature- based 的入侵偵測產品都需要一個相對應的攻擊特徵。Signature-based 的安全機制無法跟已知上漏洞增加的速度。以安全漏洞的數目之多，再加上以驚人的速度增加，極有可能目前已知的安全漏洞知之中並沒有相對應的攻擊特徵（Signature）。

二、誤判率：

現行的攻擊模式與特徵比對的技術，常出現誤判（將正常的網路存取行為誤認為攻擊行為；或無法精確的辨識出攻擊行為）的狀況，（現已有資訊安全廠商針對此缺 點提出新的技術解決方案來降低誤判率），當誤判率過高，管理人員疲於調查追蹤錯誤的警訊，會造成安全設備與安全管理人員的效率降低。誤判的狀況，在軟體式 的網路型入侵偵測系統最為嚴重。因軟體式網路型入侵偵測系統多半有效能上的瓶頸，當效能跟不上網路流量的速度，就會開始掉封包（Drop packets），導致封包資訊不完全而容易造成誤判。

三、缺乏立即有效的回應：

入侵偵測系統，顧名思義強調偵測監控的功能。目前的入侵偵測系統如果偵測到攻擊行為，能夠馬上通知管理者並提供即時分析，但卻普遍缺乏立即回應與阻止攻擊的能力，所以比較不容易顯示出它的效益，也限制了它對網路系統所提供的防護功能。

在第一年的計畫中，除了針對 Windows 系統主機攻擊實作外，也針對各項新型網路攻擊研究議題現況進行深入的研讀及分析。

今年度我們將以 OWASP Top 10 十大資安漏洞為目標進行攻擊模式分析，並從研究分析中選出一般防護系統無法偵測或誤判率高之攻擊手法做為今年研究與實作的對象。

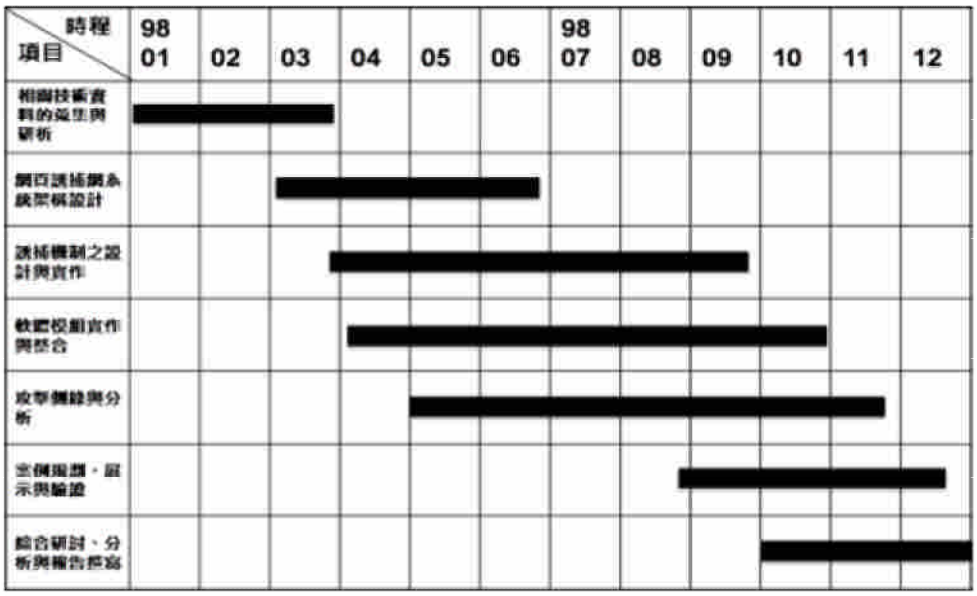
(2)網路威脅風險分析與威脅預警之核心技術與軟體單元研發

現今網路安全防護的議題已經受到全世界的注目，因為在資訊蓬勃發展的時代中，網路已成為現代人不可缺少的一部份，然而其安全的管理和防護的疏忽所造成的傷害更是不計其數，例如: eBuy、Yahoo 等等，都曾經因為資訊安全的重大事件而損失慘重。因此許多資訊安全防禦機制也如雨後春筍般冒了出來，如防火牆、入侵偵測系統。

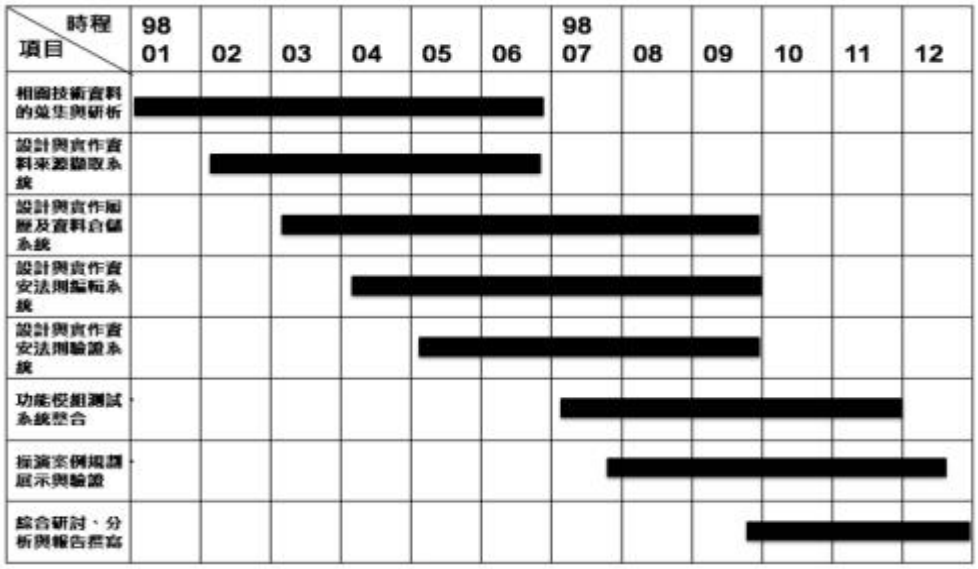
現有的防火牆、入侵偵測系統等措施，仍然有其限制，它們只能夠發掘已知的攻擊方式，對於新型的攻擊則無能為力。現有的系統無法定量地分析網路威脅的風險程度，提供預警服務，由於網路攻擊可能造成巨大的損害，我們實在不能承受網路攻擊，而只是在事後加以追查，一套有效的網路防禦系統，必須能夠隨時提高警覺，主動監控網路，即時發出預警訊號，讓網安人員可以預作防備，進而主動出擊，追查攻擊的來源。本計畫的目標之一即是對於網路威脅的風險分析與預警系統，作深入的研究，以及製作雛型系統，來驗證我們研究成果的實用性。

第二節、計畫時程

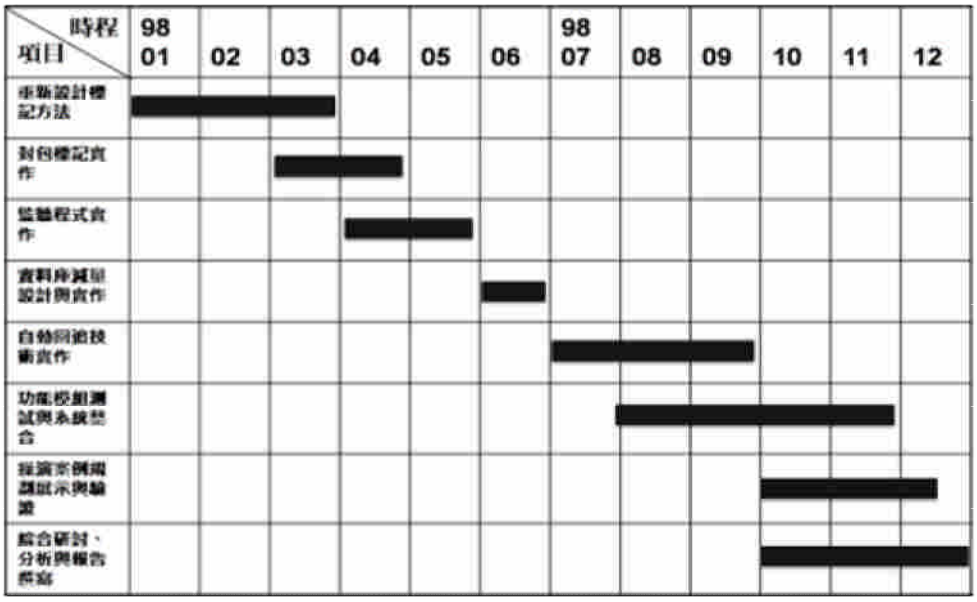
本計畫由於規模龐大，為顧及各系統需求，我們制定各系統完整的開發時程表，至2009年12月前各工作項目皆已如期完成，並藉由實際網路攻擊手法完成測試。



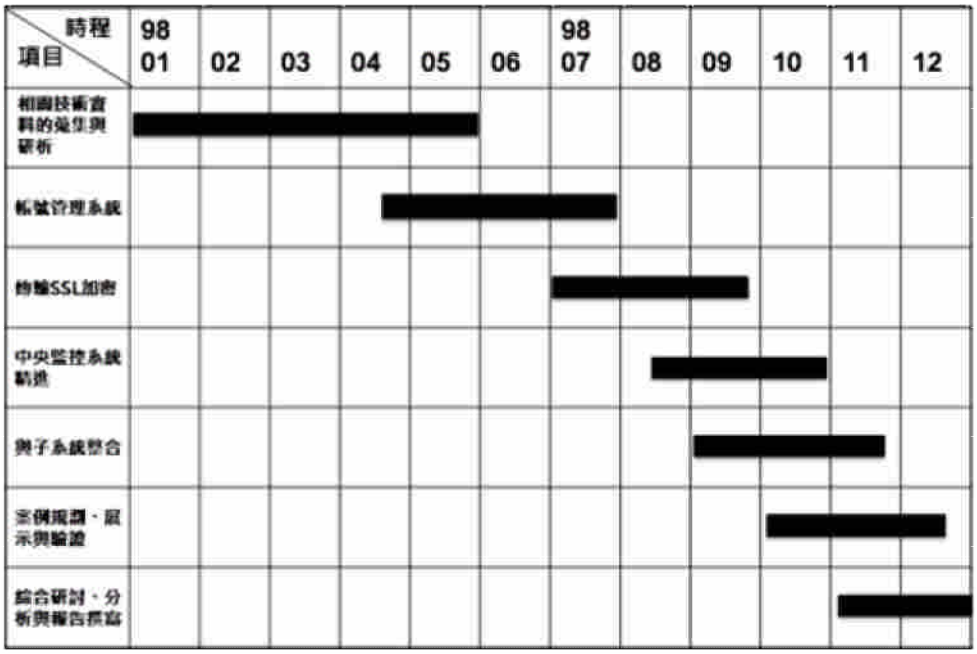
表格 1、誘捕網系統開發時程表



表格 2、網路風險分析及預警系統開發時程表



表格 3、封包標記與追蹤系統開發時程表



表格 4、系統整合時程表

第三節、計劃目標

本計畫將對第一年完成之成果進行發展，並且將其整合成一完整系統，我們在此說明此計劃目標，並在二、三、四、五章詳細描述系統原理以及使用。

議題一、Windows 誘捕網情蒐之核心技術與回追技術軟體發展：

本研究計畫將設計出針對網頁伺服器、網頁伺服器、網頁應用程式、資料庫伺服器攻擊之誘捕技術，目標是將攻擊者的攻擊型態以及行為進行誘捕與紀錄。紀錄的成果將有助於了解新型態的網頁攻擊以及攻擊造成的影響。此外，更能有效檢視我方網路的安全性及所面臨的未知威脅，從而制訂更有效的防護機制。

議題二、網路威脅風險分析與威脅預警之核心技術與軟體單元研發：

本計畫延續第一年度的成果，將對於第一年所開發的系統進行更深入的研發，包含充分利用資料探勘技術於履歷的建置、履歷的呈現、風險評估量化與威脅預警，使系統更具實用性且更完整。因此，我們將進一步擴充現有的軟體模組，包含：資料來源擷取系統、Two-tie 資料分析架構、履歷連結系統、資安規則編輯器、資安法則驗証系統，並加入風險量化分析、參數微調模組及威脅預警機制。另外，我們也將深入研究網路攻擊的形態，期望未來新的系統能夠更廣泛的應用在不同形態與未知的網路攻擊。

第四節、計劃工作項目

本計畫分為兩年的計劃，以下為計劃兩年內的工作項目，已經全部完成。

第一年度工作項目：

(一) WINDOWS 誘捕網情搜之核心技術與軟體單元發展

1. WINDOWS 系統誘捕技術與隱形記錄：WINDOWS 系統入侵路徑與模式、WINDOWS 系統入侵過程的側錄與解析技術、WINDOWS 系統側錄記錄之保護與隱藏技術 --- (林金城教授)
2. WINDOWS 網頁伺服器之誘捕風險與安全控管技術：虛擬系統之誘捕防護罩模式與架構、網頁伺服器之誘捕防護罩技術、網頁伺服器之誘捕風險控管技術 --- (何翊教授、陳建華教授)
3. 網路封包標記與追蹤技術：研究網路封包標記架構、入侵追蹤模式、封包資料摘要與壓縮技術、網路攻擊路徑重建技術 --- (趙禧綠教授)

(二) 網路威脅風險分析與威脅預警之核心技術與軟體單元研發

1. 容易使用的法則編輯與驗證技術：針對網路威脅分析及風險量化與威脅預警所需之經驗法則的表達技術、容易使用的經驗法則的編輯與驗證技術 --- (黃俊龍教授，彭文志教授)
2. 網路威脅分析及風險量化：網路威脅模式、風險量化模式、模糊理論之網路威脅分析技術、模糊理論之風險量化技術 --- (彭文志教授)
3. 威脅預警技術：模糊理論之威脅預警之經驗法則、威脅預警演算法 --- (黃俊龍教授)
4. Mining Fuzzy association rules, 以供本系統使用--- (黃俊龍教授)
5. 評估商用的法則推論引擎,以供本系統使用--- (彭文志教授)

(三) 系統整合 --- (楊武教授)

1. 管理上述各研究議題、雛形系統整合
2. 規劃與整合上述各研究議題之技術及其軟體介面
3. WINDOWS 誘捕網情搜軟體單元人機介面設計
4. 網路威脅風險分析與威脅預警軟體單元人機介面設計
5. 整合系統之人機介面設計

(四) 計劃統籌規劃與協調 --- (謝續平教授)

1. 計劃統籌規劃

2. 規劃研究技術整合、研究重點方向
3. 參與網路封包標記與追蹤技術研究、網路攻擊路徑重建技術研究
4. 參與模糊理論之網路威脅分析技術研究
5. 計劃管理

第二年度工作項目：

(一) WINDOWS 誘捕網情蒐之軟體雛型研發

1. WINDOWS 系統入侵過程的側錄與解析技術。 --- (許富皓教授)
2. WINDOWS 系統側錄紀錄之隱藏與保護技術：持續第一年的技術，開發其軟體。 --- (許富皓教授)
3. WINDOWS 網頁伺服器之誘捕網軟體雛形。--- (許富皓教授)
4. 網頁伺服器與誘捕系統之網路架構設計與實作,設計與實作展示用的網路架構與 topology，安排 ROUTER 與 MCC、HONEYNET 及 SIM 的位置及網路 IP 之安排等網路基礎建構。 --- (趙禧綠教授)
5. 網路攻擊技術與誘捕系統之對抗效力評估,以實際的攻擊技術對所建置的誘捕系統進行測試, 藉此評估此誘捕系統對攻擊技術的捕獲能力, 並檢討其成效。--- (許富皓、趙禧綠教授)

(二) 網路威脅風險分析與威脅預警之軟體雛型研發

1. 資料來源擷取系統：由防火牆日誌擷取出連線資料，存入資料庫中。--- (黃俊龍教授，彭文志教授)
2. two-tier 之分析架構：先由區域網路進行 local analysis，再將結果交由中央伺服器進行 global analysis。--- (黃俊龍教授，彭文志教授)
3. 資料探勘技術於履歷建構：根據過去的連線記錄，利用資料探勘建構每個電腦的履歷，並進一步使用資料探勘中的資料倉儲(data warehouse)技術，加速資料的統計、整合、計算及呈現。--- (黃俊龍教授，彭文志教授)
4. 自定規則編輯系統：針對網路威脅分析之經驗法則，設計一可自定規則之法則編輯器。--- (黃俊龍教授，彭文志教授)
5. 法則驗證系統：風險量化與參數微調模組。 --- (黃俊龍教授，彭文志教授)
6. 網路威脅分析及研究：各種新型態網路威脅模式之研究。--- (黃俊龍教授，彭文志教授)

7. 針對重要網路攻擊，將透過所提出之資料分析架構與履歷機制，進行實作與評估其效能。--- (黃俊龍教授，彭文志教授)
8. 威脅預警之分析與計算。--- (黃俊龍教授，彭文志教授)
9. 網路攻擊技術與其風險分析及威脅預警之對抗效力評估。--- (黃俊龍教授，彭文志教授)

(三) 系統整合 --- (楊武教授)

1. 管理上述各研究議題、子系統整合
2. 規劃與整合上述各研究議題之技術及其軟體介面
3. WINDOWS 誘捕網情搜軟體單元人機介面設計
4. 網路威脅風險分析與威脅預警軟體單元人機介面設計
5. 整合系統之人機介面設計

(四) 計劃統籌規劃與協調 --- (謝續平教授)

1. 計劃統籌規劃
2. 規劃研究技術整合、研究重點方向
3. 規劃整合系統之整體架構，掌握計劃之進度
4. 計劃管理

第二章、全系統設計描述

我們在此章節將詳細介紹全系統架構以及相關資訊，並在後面的章節講解各子系統。

第一節、全系統硬體架構

以下我們列出全系統架構圖以及各系統主機的硬體規格。

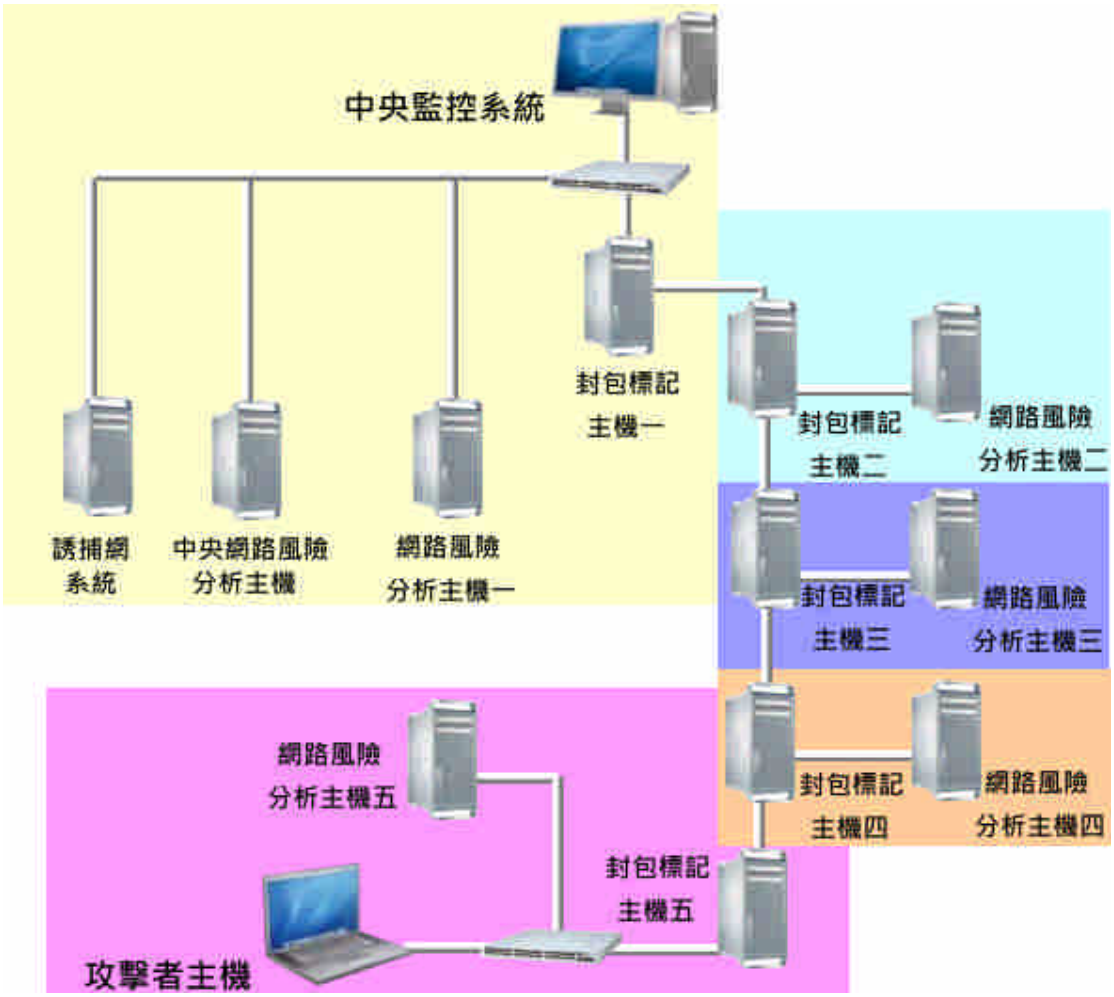


圖 1、全系統硬體架構圖

PC	**14 台**
Switch	**2 台**

表格 5、全系統硬體統計表

1.1 中央監控系統

硬體規格	
CPU	INTEL Pentium Dual Core E5200
主機板	ASUS P5KPL-AM
記憶體	Transcend 2G DDR2-800
光碟機	Pioneer DVR-217
硬碟	Western Digital 640G WD6400AAKS(三年保固)
Power	FSP 350W/ATX (ATX-350PNT)
網路卡	Network Interface Card(NIC)

表格 6、中央監控系統硬體規格表

1.2 誘捕網系統

硬體規格	
CPU	Intel Q6600
主機板	ASUS P5K-PRO
記憶體	4GB DDR2-800 RAM
顯示卡	VGA GeForce 7200 series (128M)
光碟機	DVD-ROM
硬碟	160G HDD*2
Power	PSU-500W
網路卡	Network Interface Card(NIC)

表格 7、誘捕網系統硬體規格表

1.3 網路風險分析及預警主機

a. 中央網路風險分析及預警主機

硬體規格	
CPU	Intel E7200
主機板	MD Gigabyte EP35-DS3L P35/ICH9
記憶體	4GB DDRII 800 RAM
光碟機	DVD-ROM
硬碟	640GB HD

Power	PSU 350W
網路卡	Network Interface Card(NIC) x 2

表格 8、中央網路風險分析及預警主機硬體規格表

b. 網路風險分析及預警主機

硬體規格	
CPU	INTEL Pentium Dual Core E5200
主機板	ASUS P5KPL-AM
記憶體	Transcend 2G DDR2-800
光碟機	Pioneer DVR-217
硬碟	Western Digital 640G WD6400AAKS(三年保固)
Power	FSP 350W/ATX (ATX-350PNT)
網路卡	Network Interface Card(NIC)

表格 9、網路風險分析及預警主機硬體規格表

1.4 封包標記追蹤系統

硬體規格	
CPU	Intel E7200
主機板	MD Gigabyte EP35-DS3L P35/ICH9
記憶體	4GB DDRII 800 RAM
光碟機	DVD-ROM
硬碟	640GB HD
Power	PSU 350W
網路卡	Network Interface Card(NIC):Realtek Semiconductor Co., Ltd. RTL-8169 Gigabit Ethernet (rev10) x 2

表格 10、封包標記與追蹤系統硬體規格表

1.5 網路交換機

廠牌	D-Link

表格 11、系統交換機廠牌

第二節、全系統軟體架構

以下我們將針對各系統軟體部份作介紹。

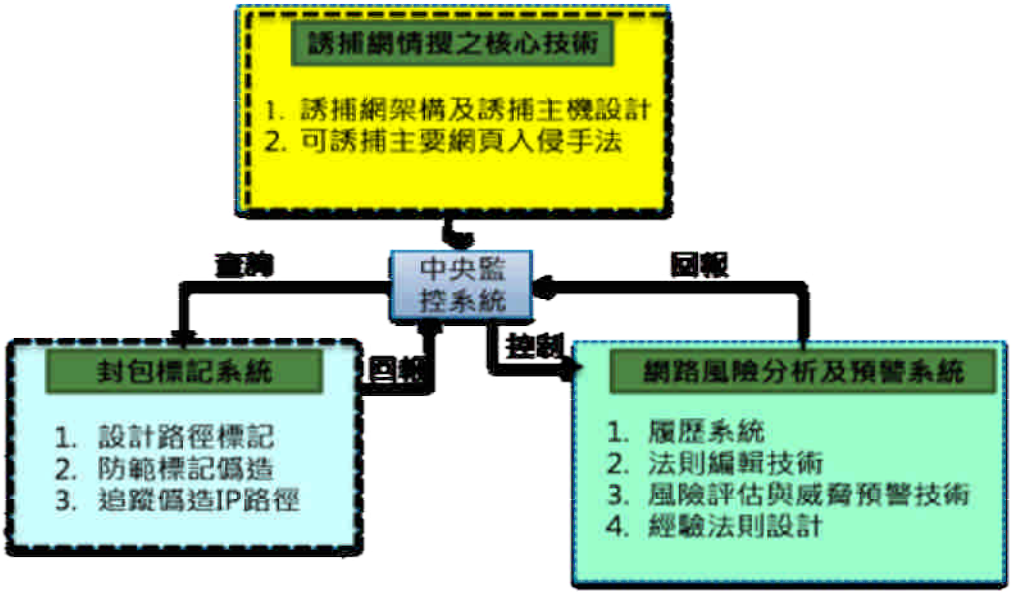


圖 2、軟體架構圖

2.1 中央監控系統

作業系統	
Fedora 11	
組成及功能	**說明**
Java 程式執行	路徑追蹤 Java GUI 介面的執行。
查詢路徑追蹤	中控中心透過 Java GUI 介面來查詢可疑 IP。
誘捕網系統回報	將有異常的行為進行誘捕並傳送至中控中心。
封包標記系統回報	將路徑封包傳回中控中心。
網路風險分析及預警回報	將六項資安規則裡面的資料傳送至中控中心顯示。
資料庫存取	資料庫存取只能由本地端進行存取，外部查詢需透過 Socket。

表格 12、中央監控系統功能表

2.2 誘捕網系統

作業系統	
Windows XP	
組成及功能	**說明**
Honeywall	資料的搜集與儲存
Honeypot	資料的誘捕與捕獲
資料分析	分析 Honeywall 和 Honeypot 之資料
每單元資料交換介面	皆以 TCP 之方式傳輸

表格 13、誘捕網系統功能表

2.3 網路風險分析及預警系統

作業系統	
Debian	
組成及功能	**說明**
資料庫存取	提供資料庫記錄(1)原始防火牆日誌(firewall log)、(2)資安規則、(3)各主機之履歷結構(Profile)、(4)MCC使用記錄及(5)HoneyNet 使用記錄。
履歷(profile)系統	將資料庫中之原始firewall log依不同的主機及連線建立各主機之profile，並存入資料庫中，以供查詢。
異常行為分析	以MCC設定之資安規則對資料庫中之各主機profile進行篩選，分別針對六項規定找出異常之主機。
中央監控介面	利用 Java 通訊平台，接受 MCC 下達之指令。

表格 14、網路風險分析及預警系統功能表

2.4 封包標記追蹤系統

作業系統	
Debian	
組成及功能	**說明**
封包標記	將傳送進來的封包進行封包標記。
封包轉送	將標記過後的封包進行轉送。
記錄封包標記	將傳送進來的封包標記內容讀取出來，記錄到資料庫。
資料庫存取	資料庫存取只能由本地端進行存取，外部查詢需透過 Socket。
Java 程式執行	路徑追蹤 Java GUI 介面的執行。

表格 15、封包標記與追蹤系統功能表

第三章、誘捕網系統

此章節我們將詳細介紹在本計劃所開發的誘捕網系統。

第一節、前言

我們根據第一年的網路攻擊模式研究分析與誘捕網實作及 OWASP 的安全性年度報告,可以瞭解網頁伺服器與網頁瀏覽器已成為現今攻擊者最為喜愛的攻擊對象，眾多破壞力強大的惡意攻擊皆以此二者為攻擊目標。

網路誘捕技術是一種欺騙駭客攻擊的技術，它被用來吸引入侵者，使他們進入受控的環境之中，並使用各種監控技術來捕獲入侵者的行為。本研究計畫著重在建立能夠記錄下 SQL injection 攻擊的網頁型態之 honeypot，針對 SQL injection 的攻擊手法，來記錄駭客攻擊和入侵模式。

常見之入侵偵測系統、Web FW 與 Honeypot 對於針前系統發起的攻擊有良好的表現，但是在面對新型態的 web attack 時，偵測與防堵能力均十分有限。此外，基於已知之漏洞雖然能夠防範與偵測，但對於未知之網頁漏洞，則無法以有效的方式作分析。藉由網頁誘捕網系統的研究分析與實作，不只對新型網路攻擊模式捕獲有幫助且可即時補足傳統防護系統的盲點。

第二節、SQL Injection 技術介紹

電腦系統的安全一直是個重要的議題，目前一般的電腦管理者都會替系統安裝修正檔，防毒軟體，架設防火牆等等，但可能由於程式設計者的疏忽，而導致系統中存在漏洞，使得攻擊者得以利用這樣的漏洞。

在現今的應用程式架構中，大部分都含有資料庫，以容納各式各樣的資料。而在各類型的資料庫中，又以結構化查詢語言(SQL Structure Query Language)為基礎的關聯式資料庫管理系統(RDBMS Relational Database Management System)最為流行。

一般的 web 應用程式的程式設計師在存取資料庫時，往往是利用 PHP 等第三代語言來組織 SQL 語言，然後再傳遞給關聯式資料庫系統執行，以建立或刪除資料結構，賦予或移除使用權限，乃至於新增、修改、刪除或查詢資料。因為關聯式資料庫所有的執行動作皆是遵循 SQL 命令，所以透過此種方式可以很方便地完成各種資料維護工作。但也正因為 SQL 語言無所不能，所以稍有漏洞就

會讓駭客有機可乘。

網站的資料存取一般來說是比較危險的，因為網際網路是一個開放的環境，而不像一般公司內部網路，除了有電腦本身的安全設計，還可以過濾篩檢員工的身分背景。網際網路上龍蛇雜處，大部分的使用者都循規導矩，但少數圖謀不軌的人卻處心積慮地要侵入我們的系統，竊取有價值的資料。但一般的網管人員及網頁設計師，可能在安全設定上有著重重防範，如架設防火牆，設計非軍事區(DMZ)，限制網站登入者的身分等等。但由於缺乏對 SQL 語言及資料庫管理系統的認知，而大開系統的後門。

SQL injection 是常見的一個重大威脅，而且針對含資料庫的應用程式。SQL injection 是利用輸入特殊命令，讓系統將之與標準的資料庫查詢程式和資料合併在一起，送給資料庫管理系統執行，因此有一段有害的程式碼被正常的程式碼包裝起來形成『隱碼』，直接對資料庫存取資料或進行破壞，進而造成資料庫損毀或資料流失。SQL injection 本身不僅可以由網頁上的欄位(HTTP POST method)發起攻擊，也可以直接從附加在 URL 變數以 HTTP GET method 送出。此外，更有相當多的變形(如子字串組合)與編碼手法(如 unicode)，攻擊者可以技巧性的繞過網站上之防護措施。因此針對未知之 SQL injection 攻擊字串之分析實為一大挑戰。

2.1.SQL injection 的弱點形式

(1) 未做適當的字元(串)過濾

當 web application 的設計者沒有對來自 user 的 input 做適當的字元(串)過濾時，它的 input 會經由 web application 合成 SQL 查詢字串的程式，合成非預期，甚至是有害的 SQL 查詢字串。

以下麵的 SQL 查詢字串的合成程式碼來說明：

```
statement = "SELECT * FROM users WHERE name = '" + userName + "';"
```

像這樣的程式，在未做適當的字元過濾的情況下，若以下列字串

```
a' or 't'='t
```

作為 input 來做 SQL 查詢字串合成的話，就會合成出下列的 SQL 查詢字串：

```
SELECT * FROM users WHERE name = 'a' OR 't'='t';
```

如果這樣的程式碼被用在對使用者的身分認證的話，由於't'='t'永遠是 true，將導致身份認證成功，且取得某一個合法使用者的身份及權限。

在許多的資料庫伺服器的軟體當中，允許一次的呼叫可以執行多道 SQL 的查詢命令，有些 SQL 的 API 有做到限制這樣的行為，像是 php 的 mysql_query 這個 SQL 查詢的 API，這樣可以避免攻擊者嵌入獨立且與原本程式無關的 SQL 查詢字串。

(2)不正確的變數型態處理

SQL injection 也有可能發生在沒有做正確的變數型態處理的 web application 中，例如：

```
statement := "SELECT * FROM data WHERE id = " + a_variable + ";"
```

這個 SQL statement 中，a_variable 應該要是一個正整數的值，以查詢與其相符的 id 的資料，但若程式設計者沒有對 a_variable 做是否為整數的檢查時，當下列的字串

```
1;DROP TABLE users
```

被輸入為 a_variable 變數的值時，就會執行 drop(delete) users 這個 table，產的 SQL 查詢字串如下：

```
SELECT * FROM DATA WHERE id=1;DROP TABLE users;
```

(3)資料庫伺服器內部的弱點

資料庫伺服器的軟體本身也有可能存在弱點，像是 MySQL server 的 mysql_real_escape_string()函式，能使攻擊者用 unicode 的編碼方式，繞過程式設計者的過濾，而成功執行 SQL injection。

(4)Blind SQL Injection

當 SQL injection 的結果不會呈現在網頁的頁面上時，攻擊者有可能使用以下的方式來做 injection 有沒有成功的檢查：

a. 條件的 respones

其中一種 blind SQL injection 的方法是在 select 敘述句中加上永遠成立，或是永遠失敗的條件式，如下所列：

```
SELECT booktitle FROM booklist WHERE bookId = 'OOk14cd' AND 1=1;
```

會產生原來的結果(不會影響)

```
SELECT booktitle FROM booklist WHERE bookId = 'OOk14cd' AND 1=2;
```

會產生與原來不同的結果，攻擊者可以藉此來判別是否有 SQL injection 的弱點或是 SQL injection 是否成功。

b. 條件的 errors

這個種類的 blind SQL injection 在 where 條件式成立的時候會產生 SQL 的 error，例如：

```
SELECT 1/0 FROM users WHERE username='Ralph';
```

者能夠藉此來找出可能被使用的 schema。

c. 時間延遲

攻擊者可以嵌入需要一段足夠長時間執行的 query，用 web server 回傳結果時間延遲來判斷是否有 SQL injection 的弱點或是 SQL injection 是否成功，快速回應可能表示 SQL injection 失敗，過一段時間才回應可能表示 SQL injection 的注入碼有被成功執行。

d. 一些可以用在 SQL injection 的 select 敘述命令

Version	SELECT @@version
Comments	SELECT 1; #comment SELECT /*comment*/1;
Current User	SELECT user();

	SELECT system_user();
List Users	SELECT user FROM mysql.user; -- priv
List Password Hashes	SELECT host, user, password FROM mysql.user; -- priv
Password Cracker	John the Ripper will crack MySQL password hashes.
List Privileges	SELECT grantee, privilege_type, is_grantable FROM information_schema.user_privileges; -- list user privs SELECT host, user, Select_priv, Insert_priv, Update_priv, Delete_priv, Create_priv, Drop_priv, Reload_priv, Shutdown_priv, Process_priv, File_priv, Grant_priv, References_priv, Index_priv, Alter_priv, Show_db_priv, Super_priv, Create_tmp_table_priv, Lock_tables_priv, Execute_priv, Repl_slave_priv, Repl_client_priv FROM mysql.user; -- priv, list user privs SELECT grantee, table_schema, privilege_type FROM information_schema.schema_privileges; -- list privs on databases (schemas) SELECT table_schema, table_name, column_name, privilege_type FROM information_schema.column_privileges; -- list privs on columns
List DBA Accounts	SELECT grantee, privilege_type, is_grantable FROM information_schema.user_privileges WHERE privilege_type = 'SUPER'; SELECT host, user FROM mysql.user WHERE Super_priv = 'Y'; # priv
Current Database	SELECT database()
List Databases	SELECT schema_name FROM information_schema.schemata; -- for MySQL >= v5.0

	SELECT distinct(db) FROM mysql.db -- priv
List Columns	SELECT table_schema, table_name, column_name FROM information_schema.columns WHERE table_schema != 'mysql' AND table_schema != 'information_schema'
List Tables	SELECT table_schema,table_name FROM information_schema.tables WHERE table_schema != 'mysql' AND table_schema != 'information_schema'
Find Tables From Column Name	SELECT table_schema, table_name FROM information_schema.columns WHERE column_name = 'username'; -- find table which have a column called 'username'
Select Nth Row	SELECT host,user FROM user ORDER BY host LIMIT 1 OFFSET 0; # rows numbered from 0 SELECT host,user FROM user ORDER BY host LIMIT 1 OFFSET 1; # rows numbered from 0
Select Nth Char	SELECT substr('abcd', 3, 1); # returns c
Bitwise AND	SELECT 6 & 2; # returns 2 SELECT 6 & 1; # returns 0
ASCII Value -> Char	SELECT char(65); # returns A
Char -> ASCII Value	SELECT ascii('A'); # returns 65
Casting	SELECT cast('1' AS unsigned integer); SELECT cast('123' AS char);
String Concatenation	SELECT CONCAT('A','B'); #returns AB SELECT CONCAT('A','B','C'); # returns ABC
If Statement	SELECT if(1=1,'foo','bar'); -- returns 'foo'
Case Statement	SELECT CASE WHEN (1=1) THEN 'A' ELSE 'B' END; # returns A
Avoiding Quotes	SELECT 0x414243; # returns ABC

Time Delay	SELECT BENCHMARK(1000000,MD5('A')); SELECT SLEEP(5); # >= 5.0.12
Make DNS Requests	Impossible?
Command Execution	If mysqld (<5.0) is running as root AND you compromise a DBA account you can execute OS commands by uploading a shared object file into /usr/lib (or similar). The .so file should contain a User Defined Function (UDF). raptor_udf.c explains exactly how you go about this. Remember to compile for the target architecture which may or may not be the same as your attack platform.
Local File Access	...' UNION ALL SELECT LOAD_FILE('/etc/passwd') -- priv, can only read world-readable files. SELECT * FROM mytable INTO dumpfile '/tmp/somefile'; -- priv, write to file system
Hostname, IP Address	Impossible?
Create Users	CREATE USER test1 IDENTIFIED BY 'pass1'; -- priv
Delete Users	DROP USER test1; -- priv
Make User DBA	GRANT ALL PRIVILEGES ON *.* TO test1@'%'; -- priv
Location of DB files	SELECT @@datadir;
Default/System Databases	information_schema (>= mysql 5.0) mysql

表格 16、SQL injection 的 select 敍述命令表

第三節、設計原理

3.1 安全等級提升：

防護等級	效果	規避方法
等級一	預設環境，對攻擊者無任何防範	特殊字元型攻擊 錯誤嘗試型攻擊 時差型攻擊
等級二	將特殊字元(‘, “, \)前加上 \	錯誤嘗試型攻擊 時差型攻擊
等級三	阻擋 MySQL 送出的錯誤訊息	時差型攻擊
等級四	過濾數字、字母 (0-9, a-z, A-Z)以外的字元	已知的 SQL Injection 攻擊手法皆無法繞過防護等級四

表格 17、誘捕網防護等級表

(1)防護等級一的設定標準：

防護等級一未做任何防範，開始捕捉攻擊事件，並將其紀錄。

(2)防護等級二的設定標準：

將輸入字串中的特殊字元 (‘, “, \)除去後，執行 SQL 的結果，與原始字串執行 SQL 的結果比對，若結果不相同，則設定防護等級為二。

(3)防護等級三的設定標準

測量 SQL 的執行時間並未超過設定的門檻值 且 繞過防護等級二之保護

(4)防護等級四的設定標準

測量 SQL 的執行時間是否超過設定的門檻值，若是超過，代表達到防護等級四。

3.2 自動設定防護等級

根據攻擊者輸入的 SQL injection 攻擊字串所達到技術等級的不同，在適當的時間後，給予不同的防護等級。

不同的防護等級會給予攻擊者不同程度的阻力，讓攻擊者以為網站管理者已對漏洞做修補，誘使他用進階方式攻擊網站。

輸入字串同時以四種防護等級作過濾並記錄，取最低無法通過之防護等級，作為防禦該攻擊者下次進入系統之等級。

3.3 誘捕未知型攻擊

將真實密碼存在另外的欄位，而原本存密碼的欄位放置假密碼，可捕捉到使用假密碼登入的攻擊者。

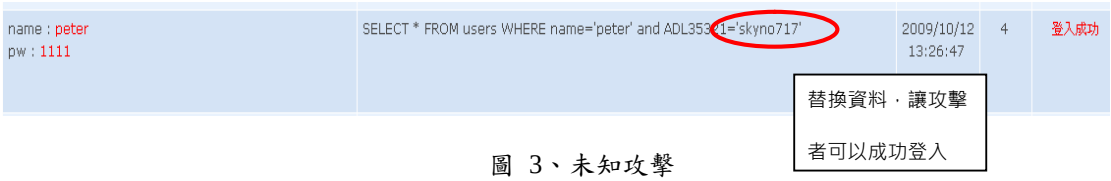


圖 3、未知攻擊

3.4 攻擊紀錄

在攻擊者對誘捕網測試到入侵成功的過程皆會被完整地記錄下來。

輸入欄位	SQL字串	Date	user_lv	用戶狀態	Input_Lv / 攻擊狀態	Comments
tid : 9 UNION ALL SELECT NULL,pw,NULL FROM users	SELECT * FROM thread WHERE tid=9 UNION ALL SELECT NULL,pw,NULL FROM users	2009/10/12 13:57:31	2	獲取資料成功	L1 : 獲取資料成功 L2 : 獲取資料成功 L3 : query失敗 L4 : query失敗	安全等級維持 L2
tid : 9 UNION ALL SELECT NULL,name,NULL FROM users	SELECT * FROM thread WHERE tid=9 UNION ALL SELECT NULL,name,NULL FROM users	2009/10/12 13:57:25	2	獲取資料成功	L1 : 獲取資料成功 L2 : 獲取資料成功 L3 : query失敗 L4 : query失敗	安全等級維持 L2
tid : 9 UNION ALL SELECT 1111,2222,3333 FROM users	SELECT * FROM thread WHERE tid=9 UNION ALL SELECT 1111,2222,3333 FROM users	2009/10/12 13:57:16	2	獲取資料成功	L1 : 獲取資料成功 L2 : 獲取資料成功 L3 : query失敗 L4 : query失敗	安全等級維持 L2
tid : 9 UNION ALL SELECT 1111, 2222, 3333	SELECT * FROM thread WHERE tid=9 UNION ALL SELECT 1111, 2222, 3333	2009/10/12 13:56:46	2	獲取資料成功	L1 : 獲取資料成功 L2 : 獲取資料成功 L3 : query失敗 L4 : query失敗	安全等級維持 L2
tid : 9 UNION ALL SELECT NULL, NULL, NULL	SELECT * FROM thread WHERE tid=9 UNION ALL SELECT NULL, NULL, NULL	2009/10/12 13:56:38	2	獲取資料成功	L1 : 獲取資料成功 L2 : 獲取資料成功 L3 : query失敗 L4 : query失敗	安全等級維持 L2
tid : 9 UNION ALL SELECT NULL	SELECT * FROM thread WHERE tid=9 UNION ALL SELECT NULL	2009/10/12 13:56:25	2	query 失敗	L1 : query失敗 L2 : query失敗 L3 : query失敗 L4 : query失敗	安全等級維持 L2

圖 4、攻擊記錄

第四節、Web Honeypot 系統架構

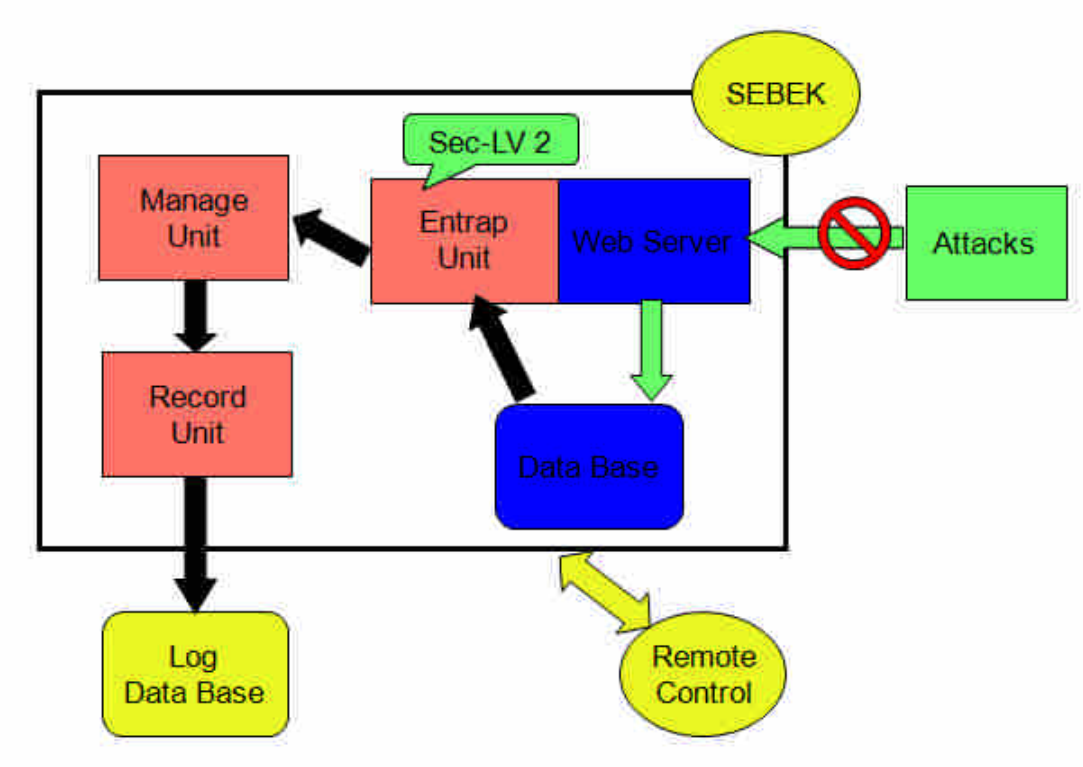


圖 5、誘捕網系統架構圖

Web honeypot 系統架構如上圖所示，主要分為幾個部分：

1. 含有 SQL injection 弱點的 web application & database

本研究計畫為建立一個網頁型態之 honeypot，以資訊安全討論區的形式呈現，在登入處理的頁面中設有 SQL injection 的弱點，引誘攻擊者入侵，藉以記錄其所做的行為。

2. Entrap unit

這個部份是用來做使用者 input 的檢查和 server response 的檢查以及 security level 的模式設定與切換的實作，若有攻擊者用 sql injection 的方式，成功執行惡意的 SQL 指令，則經過使用者 input 的檢查和 server response 的檢查，偵測到此一惡意行為，我們會將 security level 的等級調升。

3. Manage unit

這部份的設計，為因應可以做 remote control，從遠端能夠調整 security level。

4. Record unit

為了記錄攻擊者藉由 SQL injection 入侵的所有行為，需要一個記錄的機制，來完成記錄行為的動作，這個部分會將這些行為記錄下來存到 log database

中，且基於安全的考量，log database 須和 web application 所使用的 database 分開放置。

5. Log database

記錄攻擊者行為模式的資料庫，所記錄的欄位如下：

1. *Ip – 來源的 IP*
2. *Seclv – 目前所在的 security level*
3. *Time – 事件發生時的時間*
4. *Page – 事件發生時所在的頁面*
5. *Acc – 使用者輸入的 account*
6. *Pwd – 使用者輸入的 password*
7. *Sqlstr – 送到 MySQL server 執行的 SQL 查詢字串*

6. Sebek

Sebek 是一個現有的 kernel level rootkit，引入此一工具是為了防範受到非 SQL injection 的攻擊，使得 honeypot 被入侵，如此一來，我們在這個時間點之後所記錄的資料可能都是不可信的，為了找出這個時間點，以及記錄其行為模式，我們引入使用 sebek。

第五節、資料庫與程式碼設計

5.1 資料庫- *log*

用來紀錄系統所捕捉到的攻擊事件之資料，分別存入四個資料表中：attacker_log, attacker_log_input, attack_log_sec_level, black_list，各資料表內容如下：

5.1.1 資料表 attacker_log :

紀錄攻擊行為的主要資料表，每筆資料為一次攻擊事件。

No.	Field	Type	Null	Key	Default	Extra
1.1.1	Id	int(20)	否	PKI		auto_increment
1.1.2	Ip	varchar(50)	否		NULL	
1.1.3	user_sec_level	int(2)	是		NULL	
1.1.4	max_sec_level	int(2)	是		NULL	
1.1.5	Time	int(30)	否		NULL	
1.1.6	sql_string	mediumtext	否		NULL	
1.1.7	attack_page	varchar(50)	否		NULL	
1.1.8	user_authentic_type	int(1)	否		0	
1.1.9	log_comments	text	是		NULL	

1.1.1 id: 每筆攻擊紀錄之 unique id

1.1.2 ip: 攻擊來源 IP

1.1.3 user_sec_level: 攻擊發生時，此攻擊者的安全防護等級。

1.1.4 max_sec_level: 此次攻擊所能達到的最高防護等級。

(e.g., 3 代表：可以通過等級 1, 2 的檢查，會被等級 3 阻擋)

1.1.5 time: 紀錄攻擊的時間(單位：秒)

1.1.6 sql_string: 攻擊所產生的 SQL 字串。

1.1.7 attack_page: 受到攻擊的頁面。

1.1.8 user_authentic_type: 執行 SQL 指令後，Mysql 所回應的狀態

0 : query 失敗、1 : query 成功，但無法獲取資料

2 : 獲取資料成功、

3 : 登入成功 (login 檢查專用)

1.1.9 log_comments: 此筆攻擊紀錄的相關訊息

5.1.2 資料表 attacker_log_input

與 attacker_log 關聯的資料表，紀錄攻擊者所利用的字串接收欄位

No.	Field	Type	Null	Key	Default	Extra

1.2.1	Id	int(20)	否	PKI		auto_increment
1.2.2	log_id	int(20)	否			
1.2.3	Field	text	是		NULL	
1.2.4	Content	text	是		NULL	

1.2.1 id : 每筆 input 的 unique id

1.2.2 log_id : 存放 attacker_log 資料表的 id

1.2.3 field : 欄位名稱

1.2.4 content: 攻擊者輸入的內容、指令。

5.1.3 資料表 attacker_log_sec_level

與 attacker_log 關聯的資料表，紀錄攻擊分別在各個防護機制保護下的狀態。

安全防護：

等級 1 : 不做任何過濾

等級 2 : 對輸入字串做 addslash()

等級 3 : 關閉 error messages，並且對輸入字串做 addslash()

等級 4 : 去除輸入字串中所有的符號，並且關閉 error messages

狀態：

0 : query 失敗、1 : query 成功，但無法獲取資料、2 : 獲取資料成功、

3 : 登入成功 (login 檢查專用)

No.	Field	Type	Null	Key	Default	Extra
1.3.1	Id	int(20)	否	PKI		auto_increment
1.3.2	log_id	int(20)	否			
1.3.3	lv_1_sql	text	是		NULL	
1.3.4	lv_1_type	int(1)	是		NULL	
1.3.5	lv_2_sql	text	是		NULL	
1.3.6	lv_2_type	int(1)	是		NULL	
1.3.7	lv_3_sql	text	是		NULL	
1.3.8	lv_3_type	int(1)	是		NULL	

1.3.9	lv_4_sql	text	是	NULL
1.3.10	lv_4_type	int(1)	是	NULL

1.3.1 id : unique id of this table.

1.3.2 log_id : 存放 attacker_log 資料表的 id

1.3.3 lv_1_sql : 本次攻擊 SQL 在安全防護一過濾後的內容

1.3.4 lv_1_type : 未過濾 SQL 的執行狀態

1.3.5 lv_2_sql : 本次攻擊 SQL 在安全防護二過濾後的內容

1.3.6 lv_2_type : 輸入字串做 addslash()過濾，SQL 之執行狀態

1.3.7 lv_3_sql : 本次攻擊 SQL 在安全防護三過濾後的內容

1.3.8 lv_3_type : 過濾後，是否有出現 Time delay 之執行狀況

1.3.9 lv_4_sql : 本次攻擊 SQL 在安全防護四過濾後的內容

1.3.10 lv_4_type : 去除符號後，SQL 執行之狀況

5.1.4 資料表 black_list

紀錄曾經發起攻擊的 IP 黑名單。

No.	Field	Type	Null	Key	Default	Extra
1.4.1	Id	int(20)	否	PKI		auto_increment
1.4.2	Ip	varchar(50)	否			
1.4.3	sec_level	int(10)	是		NULL	
1.4.4	admin_setting_sec_level_time	int(30)	是		NULL	

1.4.1 id : unique id

1.4.2 ip : 攻擊來源 IP

1.4.3 sec_level : 上次攻擊所分配的安全防護等級。

1.4.4 admin_setting_sec_level_time :

系統管理人員可以手動提升 black_list .sec_level，此處紀錄調整的時間。

5.2 資料庫- msg_board

論壇網站(對外遭受攻擊)所使用的資料庫，包含 msg, thread, users 等資料表，各

資料表內容如下：

5.2.1 資料表 msg :

紀錄各篇文章的資料庫。

No.	Field	Type	Null	Key	Default	Extra
2.1.1	Mid	int(20)	否	PKI		auto_increment
2.1.2	Tid	int(20)	否			
2.1.3	Date	date	是		NULL	
2.1.4	Author	int(20)	是		NULL	
2.1.5	Subject	varchar(200)	是		NULL	
2.1.6	Content	mediumtext	是		NULL	
2.1.7	Reply	int(20)	是		NULL	

2.1.1 mid : 文章的 unique id

2.1.2 tid : 版面的 id (thread.id)

2.1.3 date : 文章發表日期

2.1.4 author : 文章發表人的 id (users.uid)

2.1.5 subject : 文章標題

2.1.6 content : 文章內容

2.1.7 reply : 回覆文章

5.2.2 資料表 msg :

各討論版的名稱、資訊。

No.	Field	Type	Null	Key	Default	Extra
2.2.1	tid	int(5)	否	PKI		auto_increment
2.2.2	name	varchar(20)	是		NULL	
2.2.3	info	varchar(100)	是		NULL	

2.2.1 tid : 討論版的 unique id

2.2.2 name : 討論版版名

2.2.3 info : 討論版的相關訊息、簡介

2.3 資料表 users :

紀錄註冊用戶的資訊。

No.	Field	Type	Null	Key	Default	Extra
2.3.1	uid	int(20)	否	PKI		auto_increment
2.3.2	name	varchar(50)	否			
2.3.3	pw	varchar(40)	否		NULL	
2.3.4	email	varchar(50)	是		NULL	
2.3.5	ADL35321	varchar(40)	是		NULL	
2.3.6	profile	text	是		NULL	

2.3.1 uid : users 的 unique id

2.3.2 name : 用戶帳號

2.3.3 pw : 假密碼(讓攻擊者竊取)

2.3.4 email : 用戶 email

2.3.5 ADL35321 : 真實密碼，採用特定名稱欄位讓攻擊者難以猜中，且無法直接存取這個欄位，只有透過特殊的程式檢查才能讀取。

2.3.6 profile : 用戶簽名檔

5.3 程式變數

5.3.1 module.php : 公用變數、資料庫連線的設定檔

若要移植誘捕系統至其他環境，需設定下列參數

No.	Var Name	Content	Comment
3.3.1	security_acc	name	HTML form 的帳號欄位名稱 (使用者輸入)
3.3.2	security_pw	pw	HTML form 的密碼欄位名稱 (使用者輸入)
3.3.3	database_security_acc	name	資料庫的帳號欄位名稱 (比對使用者輸入)
3.3.4	database_security_pw	ADL35321	資料庫的真實密碼欄位名稱 (比對使用者輸入)
3.3.5	database_for_user_pw	pw	資料庫的假密碼欄位名稱 (比對使用者輸入)
3.3.6	user_login_page	/board/index.php	Login 介面的程式名稱 (供使用者填入帳號密碼)
3.3.7	myip	140.115.53.35	網站的 IP Address 或是網站的網址(domain name)
3.3.8	WEBURL	http://'.myip.'/board/	首頁網址
3.3.9	time_to_upgrade_sec_level_for_demo	0	demo 用的開關，0 關閉, 1 打開
3.3.10	time_to_upgrade_sec_level	86400	提升安全防護的時段(前一次攻擊至本次攻擊相隔的時間) 單位：秒
3.3.11	send_to_mcc	true	是否利用 socket 送出訊息至 MCC server
3.3.12	mcc_ip	140.113.87.233	MCC server 之 IP
3.3.13	mcc_port	9876	MCC server 之 port
3.3.14	send_to_traceback	true	是否利用 JAVA 的 socket 程式

MySQL 連線程式：(module.php)

```
$link=mysql_connect('localhost', 'p1t1r', '1111');
                        (1)         (2)     (3)
```

(1) 資料庫的 IP，若是本地端設 localhost 即可

(2) 資料庫連線帳號

(3) 資料庫連線密碼

```
if (!$link) {
        die('Could not connect: ' . mysql_error());
}
if (!mysql_select_db('msg_board', $link)) {
                    //連線至資料庫 msg_board
        echo 'Could not select database';
        exit;
}
$link_for_log= mysql_connect('localhost', 'p1t1r_log', '1111');
                                    (1)         (2)         (3)
```

(1) 資料庫的 IP，若是本地端設 localhost 即可

(2) 資料庫連線帳號

(3) 資料庫連線密碼

```
if (!$link_for_log) {
        die('Could not connect: ' . mysql_error());
}
if (!mysql_select_db('log', $link_for_log)) {
                    //連線至資料庫 log
        echo 'Could not select database';
        exit;
}
```

※檢查專用連線：

檢查防護等級時，會將可疑的 SQL 過濾後，再進行資料庫存取，藉此找出此字串最高可達到的等級 。

```
$link_test1= mysql_connect('localhost', 'test1', '1111');
if (!$link_test1) {
	die('Could not connect: ' . mysql_error());
}
if (!mysql_select_db('msg_board', $link_test1)) {
	echo 'Could not test1';
	exit;
}
```

5.3.2 ODBC.php

將此函式從 lib.php 中獨立到 ODBC.php，避免遭受攻擊時，會顯示出 lib.php(重要函式)的名稱，增加重要檔案的隱密性。

```
function sec_fetch(&$result){
// $result : 執行 SQL query 之結果
本函式是提供假密碼所使用
將 mysql_fetch_assoc 做 hooking，檢查 SQL 抓取的資料中，是否具有真實密碼的欄位名稱(e.g. ADL35321)，若是出現，則取代為其他名稱，避免真實密碼被竊取出。
```

```
//本函式是防護等級一所使用
//將使用者輸入之資料中，具有「'」、「"」、「\」、「NULL」的字元前面加上\
```

```
//防護等級四所使用
//將所有符號去除

function symbol_strip(& $input) {
        //$input : 網站使用者所輸入的資料
        $pattern = "[-\#=\" !@$%^&*()_+,.]";
        if( is_array($input) )
        foreach ($input as $key => $child) {
            $child = ereg_replace($pattern, "", $child);
            $input[$key] = $child;
  }
```

檢查使用者 IP 是否為黑名單，若是存在，則依據此 IP 的防護等級，提升防範機制，讓此攻擊者先前所用的手法失效，藉此誘使攻擊者採用更複雜的攻擊方式。

Function　chk_time_to_upgrade_sec_level(){

檢查是否提升防禦等級的機制之一

檢查本次攻擊與上次攻擊相隔的時間,是否超過一定的區間 (可在 module.php time_to_upgrade_sec_level)，則會提升防禦。

檢查使用者本次輸入的資料，可否對目標 SQL 產生非正常結果，並判斷出惡意的輸入能規避哪些安全防護。

```
            $normal_user_result = mysql_query($normal_user_sql, $link);
                    if (mysql_num_rows($normal_user_result) > 0
                    &&    chk_mal_login($user_inputs['name'],$user_inputs['pw'],
$temp_strip['name'],$temp_strip['pw'])
```

```
$max_sec_level =0;
        $user_auth_type = 0;
        for($i=0;$i<5;$i++){
            $lv_sql[$i]='';
            $lv_type[$i]=0;
```

```
//能 delay 超過 0.1 秒 -> query 一定有成功
if($max_exec_time >=0.1){
        $lv_type[3]=2;
        //修正前面因為 query time 超過 0.1 秒被中斷的情況
```

```
//sec_level_2
            $addslash_sql=$sql;
        foreach ($user_inputs as $key => $child) {
            $addslash_sql = str_replace($user_inputs[$key], addslashes($child), $addslash_sql);
        }
```

```
$result = sec_query($sql_bl, $link);
                $user_sec_lv = $max_sec_level;
            }else{

                //檢查是否超過提升時間 : 是->提升安全等級， 否->維持
```

```
$fake_pw_sql    = " select * ";
            $fake_pw_sql .= " from ".sensitive_table_name." where
".database_security_acc."='".$temp_acc."' and ".database_for_user_pw."='".$temp_pw."' ";
            $fake_pw_result = mysql_query($fake_pw_sql, $link);
                  if (mysql_num_rows($fake_pw_result) > 0) {
```

將捕捉到的攻擊事件資訊傳給 MCC。

此處檢查 module.php 的 send_to_mcc，若是為 true 才會傳送。

```
Function    ssl_socket (){
```

```
                    insert_log_fields($log_id, $log_fields);
                    insert_log_sec_lv($log_id,$lv_sql,$lv_type);
                }
        }

    ini_set('display_errors', true);     //打開 error msg (對應　開始檢查前的關閉 error msg)
    chk_filter($_POST,$_GET);

  send_to_traceback();
  ssl_socket($user_sec_lv);
  return htmlspecialchars_decode($lv_sql[$user_sec_lv],ENT_QUOTES);
}
```

Function　insert_log_fields (){

將攻擊事件中，攻擊者輸入的資料(經過處理，避免傷害 log 資料庫)，再寫入 log 資料庫的 attacker_log_inputs 資料表。

Function insert_log (){
將攻擊事件寫入 log 資料庫的 attacker_log 資料表。

}

5.3.4 模組- 正常網站

論壇程式，設置漏洞讓攻擊者發揮。

程式名稱	功能描述
index.php	論壇首頁，提供欄位讓使用者輸入帳號密碼，也讓攻擊者有機會進行 SQL Injection。
board.php	討論區首頁，網址列有參數(id=1, 2, ...)代表不同的討論區，也是常出現的 SQL Injection 漏洞。
article.php, form.php, topic.php	論壇發表、觀看文章功能。會先檢查 login，所以攻擊者未成功 Injection 上述兩支程式前，是無法對此處進行存取。

5.3.5 模組- 正常網站

論壇程式，設置漏洞讓攻擊者發揮。

程式名稱	功能描述
Blist.php	控管介面，顯示攻擊來源的 IP 黑名單，並可手動調整防護等級。
log.php, log_list.php, log_search_form.php	顯示攻擊事件的 log 資料

第六節、功能測試

情境：

步五營侯上士在網路上習得 SQL Injection 相關知識，於是企圖運用此手法入侵架設在步一營的國軍入口網站，竊取他人帳號密碼，進一步取得機密資料。另一方面，楊上校登入中央監控系統監控。

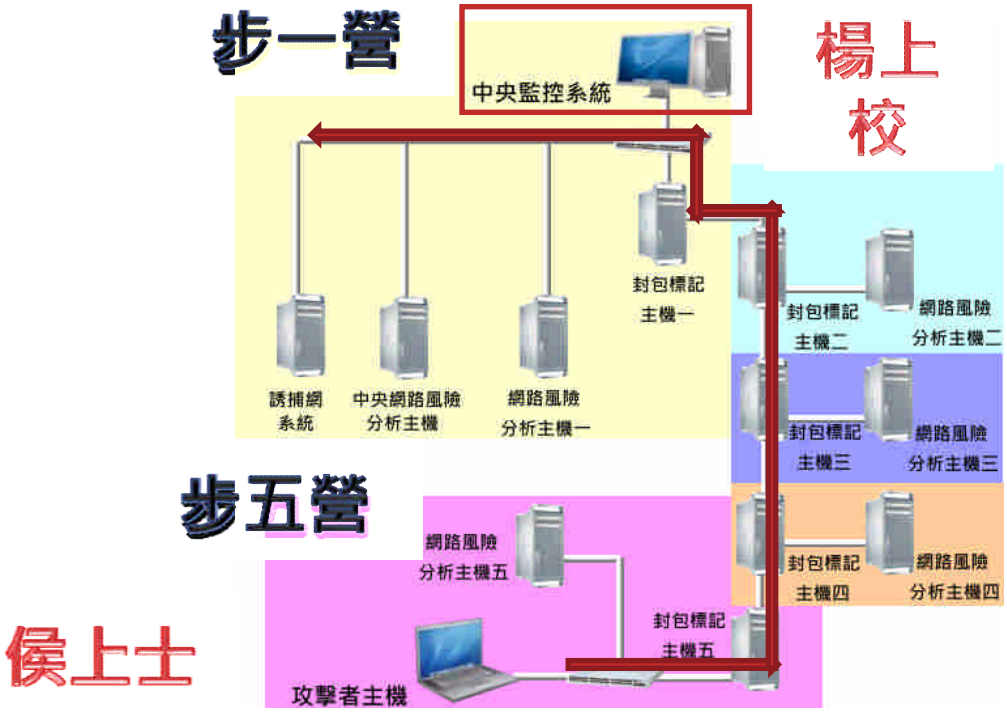


圖 6、攻擊模擬圖

6.1. 各式 SQL Injection 攻擊模擬

6.1.1 特殊字元型攻擊

- 攻擊原理及目的：

  透過特殊字元，使攻擊 SQL 成立，藉此規避檢查並且取得登入權限。

- 攻擊字串範例：『'or 1=1 # 』
- 攻擊成功：

圖 7、誘捕網受攻擊成功

- 攻擊失敗：

圖 8、誘捕網受攻擊失敗

6.1.2 錯誤嘗試型攻擊

- 攻擊原理及目的：

網站所提供的錯誤訊息，原本是讓開發者除錯用，但攻擊者同樣可以利用錯誤訊息來反覆詢問主機，順藤摸瓜地竊取出重要資訊，進而擁有登入權限。

- 攻擊字串範例：『UNION ALL SELECT NULL』
- 攻擊成功：

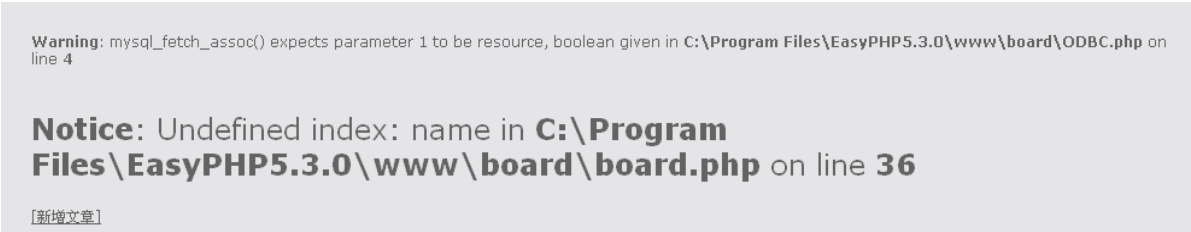


圖 9、誘捕網錯誤嘗試攻擊

- 攻擊失敗：

不論如何嘗試，畫面均不會提供任何訊息。

6.1.3 時差型攻擊

- 攻擊原理及目的：

原理同上，也是藉由反覆詢問來竊取資料，但是在錯誤訊息被屏蔽的情況下，攻擊者可以讓網站傳輸產生延遲，來達到如同顯示錯誤訊息般的效果。

- 攻擊字串範例：

『UNION ALL SELECT BENCHMARK(800000,sha1(111111)),NULL,NULL 』

- 攻擊成功：

圖 10、誘捕網時差型攻擊

- 攻擊失敗：

  不論如何嘗試，連線的延遲不受攻擊者左右。

6.1.4 未知型攻擊：

- 防禦機制：

  面臨到未知攻擊的考驗，我們的策略如下：

  所有進出資料庫的關鍵資料，皆會被攔截並且紀錄。

  檢查 輸入的帳號資料與所取得的帳號權限，是否相符

  製造誘餌資料(例如：帳號密碼)，若以此資料登入，視為攻擊。

6.2 各式 SQL Injection 攻擊紀錄

6.2.1 攻擊情景紀錄

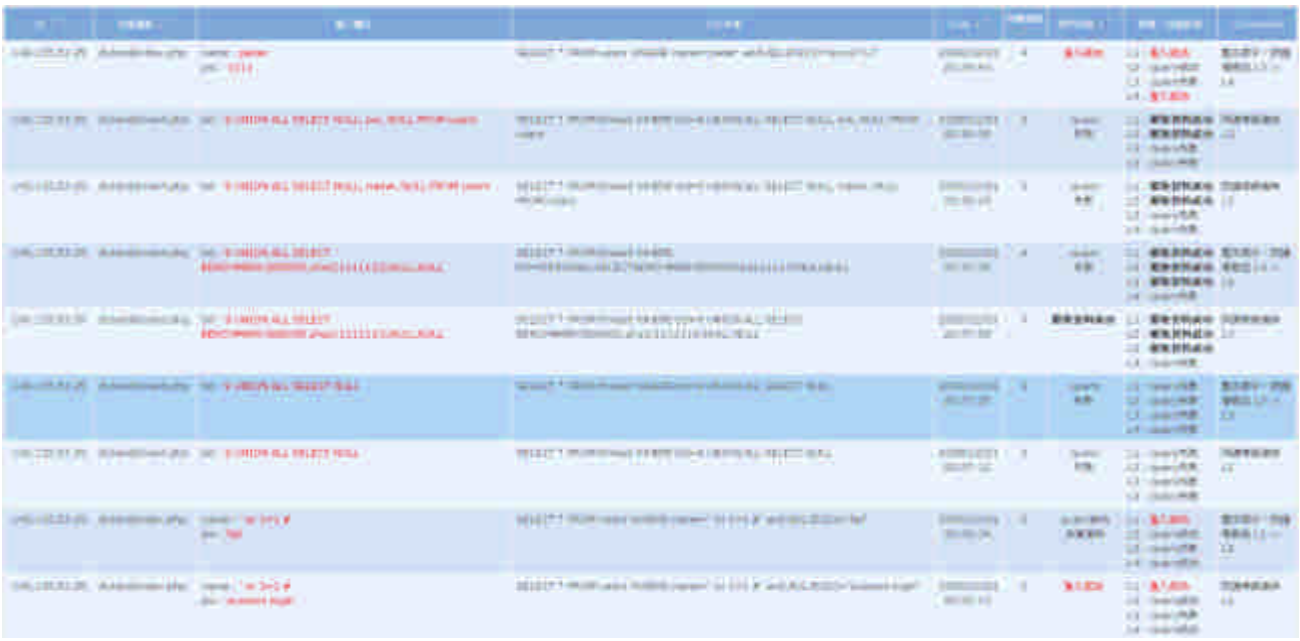


圖 11、誘捕網攻擊情境記錄

6.2.2 黑名單與遠端防護管理


Black list
共21筆
IP
防護等級
確認修改
140.115.50.233
Level 4
修改
140.113.24.204
Level 4
修改
140.113.24.119
Level 4
修改
140.115.220.113
Level 4
修改
140.115.53.10
Level 4
修改
140.115.53.35
Level 4
修改
127.0.0.1
Level 4
修改
140.115.220.228
Level 3
修改
78.142.140.194
Level 3
修改
122.116.5.170
Level 2
修改
118.160.184.87
Level 2
修改
122.116.5.38
Level 2
修改
140.113.216.142
Level 2
修改

圖 12、誘捕網黑名單

第七節、Sebek(Honeypot)

Sebek 是一個資料截取的工具。所有資料截取工具的目的都是用其截取的資料，準確地讓我們知道在 honeypot 上的事件。用它來確定 attacker 是什麼時候對系統做攻擊的，他們的攻擊手段以及攻擊成功後所做的事。

但有些惡意的行為是經過加密的，這樣的行為使用一般的 traffic filter 的方式是不能夠辨認出這些內容並加以記錄的。

我們知道，雖然訊息要加密，但在系統中某些地方不是被加密的，在這些地方可以截取到沒有經過加密的資料。一般攻擊者會加入一個木馬的 shell，用來記錄未經加密的輸入命令。

第一版的 sebek 設計成直接從 kernel 搜集按鍵的資料，參考了 Adore Rootkit，用替換 sys_read 的 system call 來截取按鍵資料，然後把這些資訊記錄到一個隱藏的文件並通過網路模擬成其它如 NetBIOS 的 UDP protocol 把它發送出去，這樣就能滿足我們 monitor 入侵者按鍵的需求了。但它易於被發現且效率不高是一個缺點。

到了第二版的 sebek，它不旦可以記錄按鍵，還記錄了所有通過 sys_read 的資料。收集到這些資料，我們就可以 monitor honeypot 的所有動作。比如有一個檔案被 copy 到 honeypot 裡，sebek 就能夠發現並記錄它，產生一同一個的拷貝。第二個重要的改變是 sebek 變得更難檢測出來，改進了日誌資料的傳輸，使得用 sniffer 檢測不到 sebek 的數據傳輸。

在 Web 誘捕系統中建構 Honey pot 有很大的好處：1. 方便與現有架構做整合，位於同一 Honey Wall 內的機器均為 Honey pot。2. 蒐集的 Log 傳輸更具隱蔽性，由於 Sebek 本身有防治 Sniffer 監聽的機制，誘捕系統的資料傳輸均透過 Sebek 則更加安全。

7.1 Sebek 結構

Sebek 的組成部分成二個：client 和 server。Client 端從 honeypot 截取資料並且輸出到網路讓 server 收集。Client 端有兩種方式收集資料：第一種是直接從網路活動的資料封包截取，第二種是從 tcpdump 格式保存的資料封包檔。當資料收集後可以上傳到相關的資料庫，也可以馬上顯示按鍵錄，且 sebek 使用 UDP protocol

進行通信。

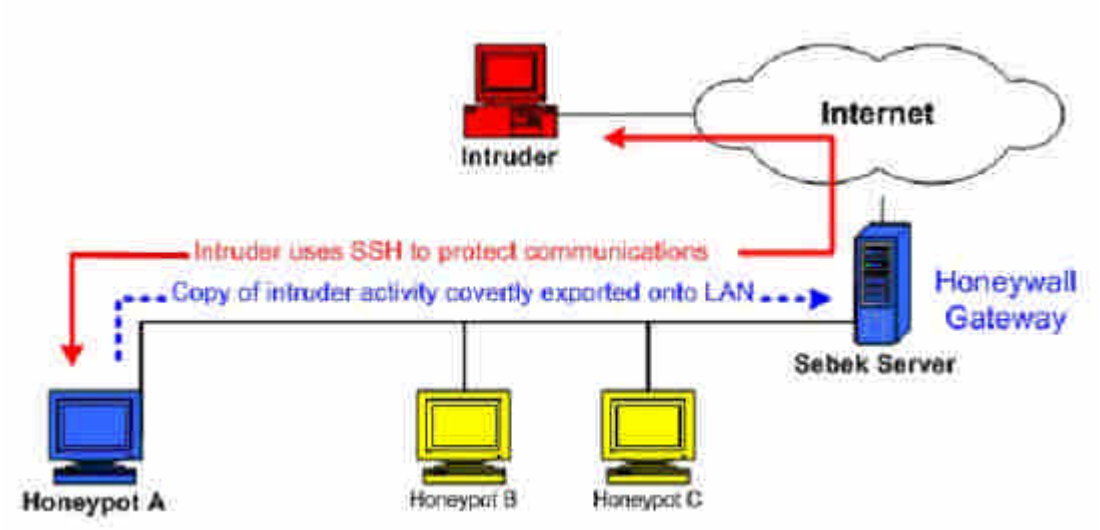


圖 13、Sebek 結構

上圖為典型的 sebek 部署。Client 模組安裝在 honeypot 裡，攻擊者行為被截取後發送到網路，並且由 honeywall gateway 收集。

Client 端完全在 honeypot 的 kernel space，根據 Linux 版本的情況以 LKM 的方式執行。Client 端可以記錄用戶通過 read() system call 的所有資料，執行 sebek 的 honeypot 以難以檢測的方式把這些資料輸出到 server 所在的網路，後 server 收集所有 honeypot 發送的資料。

7.2 Sebek 原理簡介

Client Data Capture：

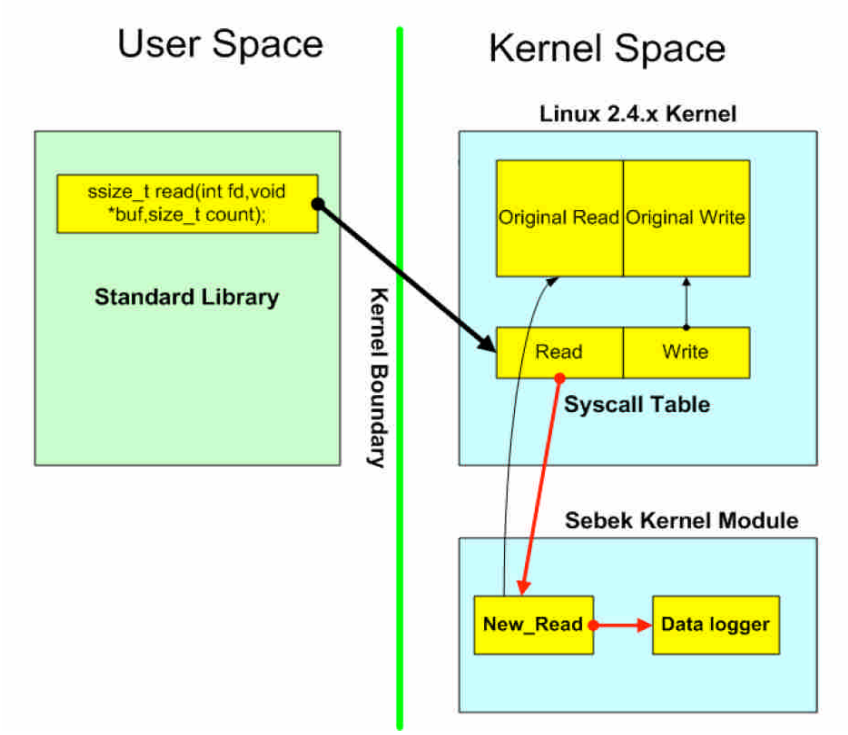


圖 14、Sebek 原理說明

上圖為 Linux kernel 2.4 的做法，在 kernel 2.6、Win32 環境也是採用相同方法：將原本的 System call Table 所指向的 system service routine address 改為自己寫好的 module routine address，藉此側錄(in New_read)有通過 system read 的內容，最後會回到原本的 service routine(Original Read)，在 Sebek 3 會側錄 read、write、socket、open 等 services。

Client Module Hiding：

Linux 版本是利用 rootkit 隱藏的技巧，將 Sebek 掛載的 module 從紀錄的 module linked list 中移除，所以重新開機後 Sebek 將無法執行。而 Win32 版本則是由於有 Hook 在系統的 Native API 中，所以隱藏手法與 Linux 有差異，將安裝時所用的 Driver 移除即可，也因此不會因為重新開機而無法正常使用 Sebek。

Client Packet Export：

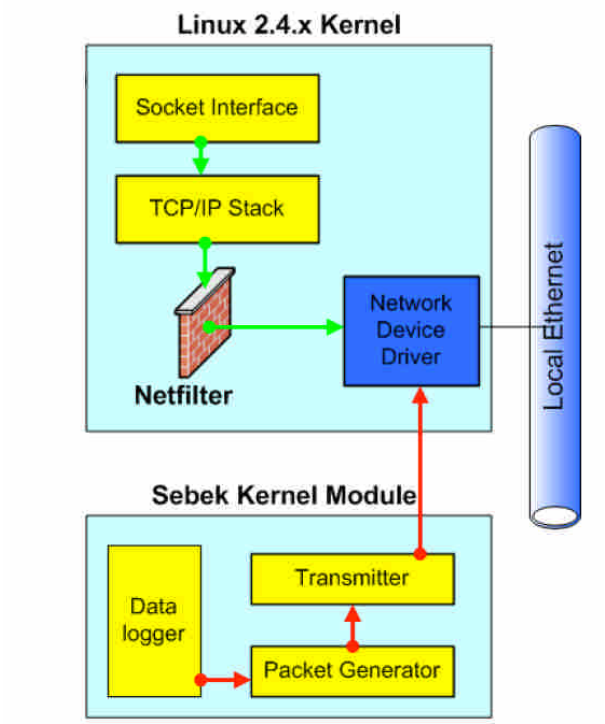


圖 15、Client Packet Export

為了避免誘捕主機遭到入侵後，入侵者可能藉由攔截與分析網路封包(sniffer)，進而偵測出 Sebek 的存在，所以此處採用秘密通訊的手法，將側錄到的 Log 資料轉為封包後，直接把封包傳至 Network Driver 發送至接收端主機；而不是以一般的傳輸流程：透過 Socket API 傳送封包。

同樣地，為了避免入侵者直接從 Network Driver 中提取資訊作分析，Sebek 會將 Linux 中的/proc/net/dev 內紀錄傳輸封包數量，扣掉 honey pot 使用的封包數，以免被偵測；Win32 則是直接 Hook 至 NDIS(Network Driver Interface Specification)，過濾到相關封包資訊。

在 Sebek version 3 中，重新設計了它傳輸的 protocol，與之前的版本並不相容，下圖是 Sebek version 3 中傳輸 protocol 的 header 詳細資料。

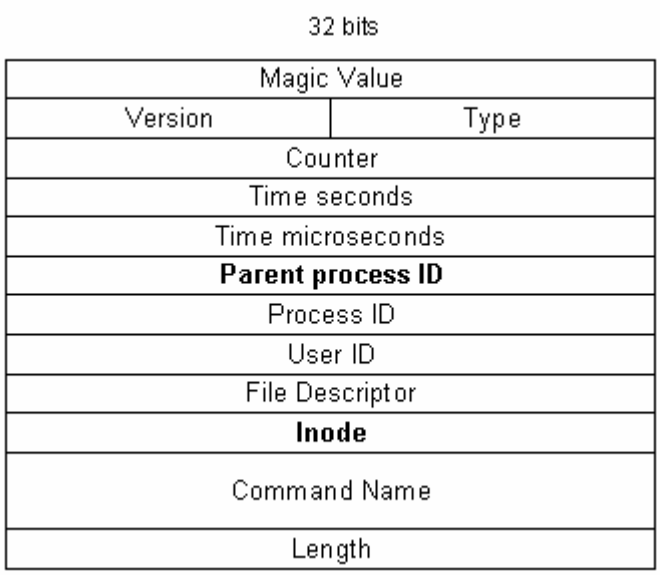


圖 16、Sebek Protool version 3 Packet header

新的 protocol header 包含了一些新的欄位，如上圖以黑色表示的欄位，parent process ID(PPID)，以及 inode。另外在 type 欄位中支援下列新的 system calls：read (0)，write (1)，socket (2)，open(3)。

欄位名	資料類型	描述
Magic	Unsigned 32 bit Int	前面是目標埠，Sebek 使用 Magic 來識別那些資料包需要隱藏
Version	Unsigned 16bit Int	Sebek 協議版本，當前版本是"1"
Type	Unsigned 16bit Int	記錄的類型。讀數據是 0，寫數據是 1。目前只實現了讀。
Counter	Unsigned 32bit Int	PDU 計數器，用來識別資料包什麼時候丟失。剛安裝時計數器是 0
Time_sec	Unsigned 32bit Int	蜜罐從 UNIX 紀元開始的秒數

Time_usec	Unsigned 32bit Int	剩餘的微秒數
PID	Unsigned 32bit Int	進程 ID
UID	Unsigned 32bit Int	用戶 ID
FD	Unsigned 32bit Int	檔描述符
Com	12 Character Array	記錄命令名字的前 12 個字元
Length	Unsigned 32bit Int	8 位元 PDU 實體的長度

發送到用戶端 Sebek 資料包的結構

當 Sebek 截獲 read()系統調用，它不但記錄讀的內容，而且注意到相關資訊。記錄頭 PID、UID、FD 和 Com 欄位的資訊來自內核處理進程產生的請求。Time_sec 欄位基於系統時間，當然這是容易被篡改的。

Length 欄位基於讀請求返回的長度。如果從 read()調用返回資料的長度大於局域網的 MTU，Sebek 會把讀到的資料分成多個分片以適應局域網傳輸。每個分片都包含了完整的 Sebek 記錄頭。

在 Sebek 用戶端裡決定對一個來自使用者空間的資料包需要隱藏依據下面的標準：資料包必須使用 UDP，UDP 目標埠必須和預先定義值匹配，並且 Sebek 頭的 Magic 欄位必須和預先設定的值匹配。如果只使用 UDP 目標埠的話將會被暴力破解方式識別出被隱藏的資料包，加上 Magic 值提高了暴 力檢測的難度，它和目標埠會產生 281,000,000,000 種組合。如果你有程式能夠每秒檢查 500,000 種組合，那麼它需要花費 6.5 天的時間 來測試。

7.3 Sebek 的限制

在蜜罐上使用的 Sebek 不能被入侵者檢測到，但是用於檢查基於內核模組技術 rootkit 的檢測工具可能會檢測到 Sebek，所以入侵者檢測到使用這種系統的可能性也是挺大的。

在 Linux 用戶端，這種風險來自內核模組支援的/dev/kmem 特性，它的功能

非常強大並且使 Linux 內核變得非常靈活。我們正是使用它們安裝 Sebek，入侵者也可以用它們來檢測並關閉 Sebek。幸運的是，當 Sebek 被關閉的時候，入侵者使用的代碼或程式以及關閉動作的記錄就發送到服務 端。在未來，一個減少被檢測到風險的方法是在服務端檢測 Sebek 什麼時候被攻擊並且馬上關閉用戶端。

7.4 警惕用戶端濫用

作為一個非常棒的監視工具，Sebek 可能被惡意用戶不正當使用。比如被入侵者用在被攻陷的系統上來獲取口令和監視合法使用者。為了減少 Sebek 潛在的危險，必須在結構設計上做改動。

早期版本的 Sebek 輸出的資料是加密的，並且資料包頭也不是統一的。我們拋棄了這種做法，並且使之變得很容易檢測不正當的使用。因為當前版本只對安裝了 Sebek 的蜜罐進行隱藏，而局域網上的其它主機都可以看到 Sebek 資料包。這意味著攻擊者如果試圖在自己攻陷的系統上安裝 Sebek，那麼管理員可以 在同網段其它主機上看到 Sebek 資料的傳輸。為了進一步減少被濫用的機會，我們還去除了 Sebek 的一些 rootkit 特性，所以入侵者沒法使用一個單 獨的 LKM 來嗅探密碼。

7.5 用戶端安裝

現在我們瞭解了 Sebek 用戶端的運行方式，接下來讓我們看一下如何安裝用戶端。用戶端的安裝必須要注意蜜罐系統上內核的運行版本，我們需要相應版本的內 核源碼來編譯 Sebek。當用戶端編譯好後，將生成一個二進位的 tar 包。這個 tar 包包含了 sbk_install.sh 腳本，這個腳本包含了 Sebek 安裝命令和配置參數，你只需簡單的修改選項，然後就可以在蜜罐執行安裝腳本，這樣 skb_install.sh 就按照指定的配置安裝內核模組。 這個腳本有八個配置參數，一般局域網上所有蜜罐都要使用相同的值。

介面：指定記錄哪個介面，預設是 eth0。不需要配置 IP 位址。

目標 IP：這個欄位定義了所有產生的 Sebek 資料包使用的目標 IP 位址。由於 Sebek 服務端收集資料包的時候不看目標 IP 位址，所以這裡無需配置成 Sebek 服務端

的 IP，而且也不推薦，否則入侵者看到這些資料包的時候有機會知道哪個主機是 Honeynet 裡提供資料控制功能的。

目標 MAC 位址：這個欄位指定蜜罐使用目標乙太 MAC 位址，目標 MAC 可以設置成缺省閘道的。把它設置成 FF:FF:FF:FF:FF:FF 將引起所有的包廣播到局域網上每個主機，這是測試的小技巧。

Magic 值：這個值結合目標埠，用來確定蜜罐上哪個包需要隱藏。強烈推薦局域網上所有安裝的 Sebek 都使用相同的 Magic 值和目標埠。

目標 UDP 埠：這個欄位定義 Sebek 資料包的目標 UDP 埠。這個值結合 Magic 值，用來確定蜜罐上哪個包需要隱藏。

源 UDP 埠：這個欄位定義了 Sebek 資料包的源 UDP 埠。

只記錄擊鍵：這個標記可以設定成 1 或 0。設置成 1，那麼 Sebek 只收集擊鍵，否則將收集所有讀到的數據。如果你想恢復 SCP 檔，必須設置成 0。

測試：如果設置了測試選項，那麼首先內核模組不會隱藏，第二，模組將產生的額外調試資訊，並且會發送到 syslog。

當在一個局域網裡配置蜜罐的時候，它們必須使用相同的 magic 值和目標埠值，這樣就能防止才一個蜜罐看到其它的 Sebek 資料包。配置 IP 和 MAC 地 址的時候，請記住 MAC 地址是最重要的。如果你配置了錯誤的目標 IP 位址，但是 MAC 位址配置正確了，並且 UDP 埠也是正確的，那麼 Sebek 記錄就能 發送到服務端。此外，當服務端運行在 Honeywall 閘道的時候，網卡介面沒有設置 IP 位址，所以必須把目標 MAC 位址設置成缺省閘道或 Honeywall 的 MAC 位址。

如果你要記錄遠端主機（不在同一個局域網內），那麼目標 IP 必須設置成主機的 IP，MAC 位址要設置成局域網缺省閘道的 MAC 位址。配置完後，執行 sbk_install.sh 將安裝 Sebek 用戶端，並且它將開始捕獲並發送資料。

7.6 服務端安裝

現在我們將討論如何恢復從用戶端來的 Sebek 資料，並且如何分析這些資料。如果你回憶一下圖 1，服務端一般會安裝在 Honeywall 閘道上。

服務端提取網路上 Sebek 用戶端資料有兩個來源，一個是捕獲所有網路資料的 tcpdump 日誌檔，另外一個是直接從網卡捕獲活動的資料傳輸。

服務端有三個組成部分。第一個部分叫 sbk_extract，它既可以像嗅探器一樣直接從網卡捕獲資料，也可以從 tcpdump 檔收集。使用這個工具你 可以用這兩種方法來恢復 Sebek 資料。用 sbk_extract 提取資料有兩種方法：第一種是把它發送給一個叫 sbk_ks_log.pl 的腳本，這個 Perl 腳本把攻擊者的擊鍵列印到標準輸出上；第二種是發送給一個叫 sbk_upload.pl，這個 Perl 腳本把 Sebek 資料裝載到 mysql 資料 庫。

sbk_extract 是一個從輸入資料（網卡或抓包檔）提取 Sebek 記錄的程式。sbk_extract 有三個選項：

-f 指定從 pcap 檔提取資料
-I 定義監聽的網卡
-p 指定 UDP 目標埠

把 Sebek 記錄導入資料庫

運行 sbk_extract，並且把輸出用管道傳給 sbk_upload.pl，如：
sbk_extract | sbk_upload.pl

監視命令列擊鍵
運行 sbk_extract，並且把輸出用管道傳給 sbk_ks_ log.pl，如：
sbk_extract | sbk_ks_ log.pl
定制分析

運行 sbk_extract，並且把輸出用管道傳給自己設計的程式來分析。

7.7 監視命令列擊鍵活動

sbk_ks_log.pl 可以讓你在服務端命令列查看運行 Sebek 用戶端主機的擊鍵活動，它沒有其它選項，需要讀 sbk_extract 的輸出，比如：

```
sbk_extract –i eth0 –p 1101 | sbk_ks_log.pl
```

在這個例子裡，sbk_extract 監聽 eth0，收集 UDP 埠 1101 上的資料，然後把這些記錄傳遞給 sbk_ks_log.pl 來提取擊鍵活動。sbk_ks_log.pl 的輸出如下面所示：

```
[2003-07-23 20:03:45 10.0.0.13 6673 bash 500]whoami
[2003-07-23 20:03:48 10.0.0.13 6673 bash 500]who
[2003-07-23 20:03:50 10.0.0.13 6673 bash 500]su
[2003-07-23 20:03:57 10.0.0.13 6886 bash 0]cd /var/log
[2003-07-23 20:03:57 10.0.0.13 6886 bash 0]ls
[2003-07-23 20:04:01 10.0.0.13 6886 bash 0]mkdir ...
[2003-07-23 20:04:20 10.0.0.13 6886 bash 0]tcsh
[2003-07-23 20:04:20 10.0.0.13 6921 tcsh 0]0
[2003-07-23 20:04:20 10.0.0.13 6920 tcsh 0]vt
[2003-07-23 20:04:20 10.0.0.13 6920 tcsh 0]en
[2003-07-23 20:04:20 10.0.0.13 6920 tcsh 0]en
[2003-07-23 20:04:27 10.0.0.13 6920 tcsh 0]cd /tmp
[2003-07-23 20:04:28 10.0.0.13 6920 tcsh 0]ls
[2003-07-23 20:04:42 10.0.0.13 6920 tcsh 0]cd /usr/lib
[2003-07-23 20:04:42 10.0.0.13 6920 tcsh 0]ls
```

sbk_ks_log.pl 的輸出和使用者在終端看到的一樣，不過我們只能看到輸入的命令，那些命令的輸出資訊是看不到的。控制字元會被轉義，比如回格鍵會替換成[BS]。sbk_ks_log.pl 的每行輸出按照如下格式：

[時間戳記 IP 位址 進程 ID 命令 使用者 ID] 文本

時間戳記顯示擊鍵動作的時間。

IP 地址是蜜罐的地址。

進程 ID 是運行的進程 ID。

命令是運行命令的前 10 個字元。

使用者 ID 是這個進程擁有者的使用者 ID。

7.8 把 Sebek 資料導入資料庫

sbk_upload.pl 是一個把記錄導入到 mysql 資料庫的 perl 腳本，這個腳本有以下幾個選項：

-u 資料庫使用者
-s 資料庫伺服器，預設是 localhost
-d 資料庫庫名
-p 資料庫口令
-P 資料庫使用的埠

執行例子：

sbk_extract –i eth0 –p 1101 | sbk_upload.pl -u Sebek –p secret – d Sebek

上面的例子 sbk_extract 監聽 eth0 上 UDP 埠是 1101 的 Sebek 資料包，提取出來的資料發送給 sbk_upload.pl，它把這些數 據插入到用戶名是“Sebek”、口令是“secret”、資料庫名字叫“Sebek”的本地主機資料庫。附錄 A 定義了資料表結構，當資料記錄插入到 mysql 資料庫後，就可以用 Web 介面來方便的查看這些 Sebek 資料。

7.9 Web 介面

sbk_upload.pl 是一個把記錄導入到 mysql 資料庫的 perl 腳本，這個腳本有以下幾個選項：

-u 資料庫使用者

-s 資料庫伺服器，預設是 localhost

-d 資料庫庫名

-p 資料庫口令

-P 資料庫使用的埠

執行例子：

sbk_extract –i eth0 –p 1101 | sbk_upload.pl -u Sebek –p secret – d Sebek

上面的例子 sbk_extract 監聽 eth0 上 UDP 埠是 1101 的 Sebek 資料包，提取出來的資料發送給 sbk_upload.pl，它把這些數 據插入到用戶名是“Sebek”、口令是“secret”、資料庫名字叫“Sebek”的本地主機資料庫。附錄 A 定義了資料表結構，當資料記錄插入到 mysql 資料庫後,就可以用 Web 介面來方便的查看這些 Sebek 資料。

7.10 總結

Sebek 是一個基於內核的資料捕獲工具，它設計用來隱蔽的捕獲蜜罐上的所有活動。我們通過加密資訊在內核空間非加密形式的活動可以得到擊鍵、恢復密碼， 並且可以監視任何通信，包括 IRC 聊天、郵件和 SSH/SCP 活動。總的來說，Sebek 提供了查看蜜罐內部活動的優秀功能。

當前版本的 Sebek 有它的局限性,一個具有作業系統知識的有經驗入侵者還是有一些方法可以檢測到存在的 Sebek，比如使用 kstat 和 chkroot 等工具。我們必須繼續努力，讓 Sebek 足夠狡猾，儘量不引起入侵者的懷疑，還要預見到入侵者會開發出工具或叫本來自動檢測主機上的 Sebek，把各種可 能都考慮到才能把 Sebek 被檢測到的機會降到最低。

第四章、封包標記與路徑追蹤系統

此章節我們將詳細介紹在本計劃所開發的封包標記系統。

第一節、前言

去年的計畫主要是架構整個封包標記實作環境，並且提出一個適合整個系統架構和環境需求的封包標記方法，在封包標記的實做，主要是將標記內容記錄在封包表頭的 Options 欄位，將所要進行的標記內容放置到自行設計的 Options 內，使路徑資訊以解析 Options 欄位就能夠取得，使用 Options 欄位會使封包的大小增加，但不影響目前的通訊協定是它的優點，而相反的會使網路流量增加。而去年以不影響目前的通訊協定為目標，因此使用了 IP Protocol 中的 Options 欄位，雖然增加了 IP 標頭的大小，但也實現將標記的資訊帶入封包。在封包所經過的路徑上加入進行封包標記的機器，機器將封包進行標記後並進行傳遞，同時機器也會將標記的資訊記錄在記錄檔內，最後在由一台主機向機器下達查詢的指令，查詢攻擊封包的路徑，達到路徑追蹤的效果。

去年的計畫主要有項缺點就是有許多防火牆或路由器都防止 IP Option 的使用，導致封包在進行傳送時，遭受到丟棄，為了解決這個問題，去年計畫採用網域式的標記，使封包在經過網域裡的最後一台 MPC 後，會將封包標記內容拿掉，然後再進行轉送到其它網域，而使連線能順利成功，也使進來的封包能夠進行標記並且記錄路徑，達成封包傳輸的路徑追蹤，但因為今年計畫路徑重建呈現方式的更改，所以即使採用網域式的標記，也無法呈現出我們想要的成果，所以在標記方式做了變更。

今年計畫的目標則著重在路徑的重建呈現，但由於去年計畫的標記設計是使用 IP 表頭的 Option 欄位，由於在此欄位做標記，封包在傳送的途中，可能被防火牆阻擋下來，而導致封包遺失，無法成功完成標記的工作，且做路徑重建，也會導致資料封包無法在營區間傳遞，為了達到跨營區的目的，今年將標記封包的欄位改成在 IP 表頭的 Identification 欄位，且以 gateway 呈現整條路徑。

考慮利用每一個IP封包表頭內的Identification 欄位作為marking的儲存空間，其原因有兩點：第一、經過切割的封包，對網路的效能有不利的影響，所以大部分路由器都具備有自動偵測MTU 的機制，送出封包時便已符合MTU 的規定，封包表頭並不需要用Identification 欄位進行封包切割，據最近的研究，只有

低於0.25%的網路總封包量才會經過切割而必須使用Identification 欄位[1]。第二、利用現存的IP 封包表頭欄位，而非Option等自創欄位，才能無接縫的適用於所有現存的路由器[1]。

至於經過切割過的封包，因為不能使用 Identification 欄位，所以無法放入標記內容，但是會在出口端將相關資料寫近資料庫內，進而做到出入口端的查詢，也希望藉由判斷 TCP 連線是否完成三向交握程序，以達到再次縮減資料庫的資料量，以上為今年大致上要完成的目標。

第二節、文獻探討

封包標記的機制，最早提出的是 Probabilistic Packet Marking (PPM) [02]，主要的設計是由收集所標記的封包內容去建立出攻擊端到受害端之間的整條攻擊路徑，在這個概念被提出後，根據不同的標記方式分為三種實作。

第一種是 PPM-node append[03]，與其他 PPM 實作方式比較是最簡單也是最基礎的作法，標記方法是將每一台路由器的位址都寫入傳送的封包中，當一個封包到達接收端時，接收端可依據封包路由器資訊所寫入的順序及位址，直接建立出封包所傳送的整條路徑。它所使用的標記是將路由器的位址寫入 IP Option 欄位，每傳送到一台路由器，就接續上台路由器標記位置添加上去。這樣的作法經過大量的路由器時，會造成封包增加的量太多，而造成網路傳輸額外的負擔增大，並且無法判定是否還有空間可寫入，攻擊者也可在 IP Option 中增加僞造的位址而誤導路徑的建立。

第二種是 PPM-node sampling[03]，從 PPM-node append 的實作進行了修正，標記方法是在封包的標頭保留一個 node 欄位(32-bit)，每經過一台路由器時，會由一定的機率 P，當 P 大於一個臨界值時，才會將路由器的位址寫入 node 欄位，將空間不足的問題解決及網路傳輸的負擔減少。但是由於是採用機率 P 進行標記，因此接收到距離為 d 的路由器的位址機率為 $P(1-P)^{d-1}$，每次接收端只能接收到其中一台路由器的位址，因此需要收集大量的封包才能達到路徑追蹤的效果。而路徑追蹤則是需要傳送 sample 封包來進行排序動作，來達到建立有次序的整條路徑。這個方式針對單一攻擊者擁有很大的成效，但是不適用於有多條攻擊路徑的攻擊者，而且也需要大量時間去進行排序的動作。

第三種是 PPM-edge sampling[03]，再為 PPM-node sampling 進行改進，增加

了 edge 的資訊在封包上，變更為新的 start、end 和 distance 三個欄位。Distance 欄位是記錄所標記的路由器到受害端的距離，start 欄位是記錄所標記的路由器的位址，而 end 欄位是記錄標記後的下一個路由器的位址，所以當一個封包被決定標記時，會將路由器 A 的位址寫入 start，distance 設為 0，到達下個路由器 B 時，會將路由器 B 的位址寫入 end 欄位，而 distance 欄位每經過一個路由器就會加一，因此就能夠推出路由器的距離。這次改良變成可以針對多重路徑攻擊者，除了由 distance 欄位了解它在攻擊路徑上的位置，也可由 start 及 end 建出路由器前後的關順序，加強了排序的時間也繼承了 PPM-node sampling 的優點。

相對於建立完整的路徑，也有以找出最靠近攻擊者的路由器為目標，而不建立出完整的路徑，而 Deterministic Packet Marking（DPM）[04]就是以此為目標，DPM 相對於 PPM，它是最容易實作，也是最不增加路由器負擔的一種作法，它不需額外保留欄位來進行標記，而是使用現有的 IP 標頭內的 17bits(ID field and the reserved 1-bit flag)，當封包一進到網路時就讓它進行標記，標記的方法是將原本的路由器位址切割成兩部份(每個 16 bits)，每次標記時，放入不同的部份，而放入的內容可以由 flag 欄位內的 0 或 1 來判定，當兩個部份都收集到後，就可以組合出原本的路由器位址，推導出最靠近攻擊者的那台路由器的位址，而不找出整條路徑。優點是不用每台路由器都進行標記的動作，大大減少路由器的負擔，但是多個攻擊者使用同一來源位址或每次攻擊封包設定不同的來源端位址時，無法有效找到攻擊者。

根據 DPM 的缺點，DPM-with address digest[05]則改進了 DPM，將原本的 DPM 概念加上雜湊函數與 ingress 路由器來區別攻擊者，欄位變成 Address fragment、Hash digest 以及 Index 三個，將原本的 IP address 切割成更多區段，而使用雜湊函數讓所有封包通過相同的路由器擁有相同的 identity，而受害端利用這個 identity 去進行 IP address 組合。延展了 DPM 的優點，增加能夠區分多個攻擊者，但必然的摘要碰撞將導致有些合法來源端會被誤判，成了它的缺點。

在前兩個主要目標都是在於建立路徑，然而也有封包標記主要是為了阻擋 DDoS 攻擊而進行設計，主要目的在於區分出是來自相同的傳輸路徑，Pi Marking Scheme[06]就是以此為目標，使用了 IP header 內的 ID field 欄位(16 bits)，將 ID 欄位分為 16/n 個區段，當封包每經過一台路由器，就會將路由器的 IP address 中的 n bits(n=1~2)寫入 ID 欄位其中一個區段，因此 ID 欄位一次可記 8 台或 16 台

路由器的個別 n bits，而放入的區段是由 TTL 的數值除以 16/n 的餘數所計算出來的，之後藉由 ID 欄位的值，就可判斷出是否來自不同的來源封包攻擊，若是判斷出是惡意攻擊的封包，就由相同的 ID 欄位去進行封包過濾的動作，使惡意攻擊的封包無法傳送，達到防止 DDoS 攻擊。但是只要中途有一台路由器沒有支援 Pi Marking Scheme，而 TTL 經過這台時，會進行遞減，而使標記的動作跳過 n bits，而造成誤判的結果。

接續 Pi Marking Scheme 的方法，StackPi Marking Scheme[07]為改進 TTL 缺點，而變成採用堆疊的方式將資訊標記上去，以達到改善不支援的路由器問題。其它的設計方式都與 PI 相同，而在 ID 欄位滿時，將舊的標記丟棄，加入新的標記進去。也就是放置的方式就像排隊一樣，先來的在前，後來的在後，而滿時就會將前面的丟棄，而補進後面的，如此一來在 Pi Marking Scheme 的 TTL 缺點就改善了。

Router Interface Marking (RIM) [08]，RIM 也是源於 PPM 的概念，去建立出攻擊者到受害端的攻擊路徑，RIM 機制在每個封包經過每台路由器時，會根據某一個機率 *P* 之下，去進行標記的動作，標記的動作主要是將路由器進入的介面 (Interface)、相對應的距離 (Hop 數)以及收集這條路徑上其他路由器的資訊，在以不增加封包額外的 overhead，額外的訊息封包，以及影響路由器的負擔前提之下，達到接收端能夠由這些所收集的封包，建立出不同的惡意攻擊的封包路徑，以樹狀的方式去呈現出來，因此可以針對 DDoS 的攻擊，做出攻擊路徑的重建，並且能夠採取額外的措施去防止攻擊。

第三節、架構說明

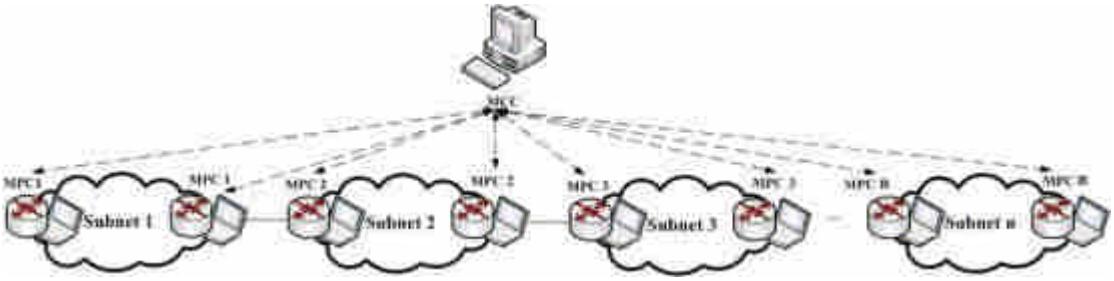


圖 17、封包標記與追蹤系統架構圖

(1) 我們所設計的主機稱為 Marking PC(MPC)

(2)整個架構會有一台中央控管主機 MCC，可以與所有的 MPC

進行資料傳輸，在做路徑追蹤時，即由 MCC 向特定 MPC 開始進行查詢。

(3) MCC 與 MPC 間用虛線相連，想表示的是兩者間並非真的有一條專線，而是 MCC 可以與所有的 MPC 溝通。

(4)雲圖為一個子網域，可以代表一個營區，或者更小的區域。

(5)由於今年的路徑改以呈現途中所經過的 gateway，所以 MPC 的放置需要放在各子網域的 gateway 附近。

第四節、封包標記

4.1 Identification 封包標記

標記設計是使用目前已有的 IP Protocol，不更改它的欄位設定，並且使用 Identification 欄位和 Flags 欄位的保留設定位元。IP Protocol 中，IP 標頭欄位的設定如圖 18，前 20 bytes 為每一個封包必定要有的資料，而 Options 是在需要特定的控制時，才會利用到，Padding 則是在 IP Header 不為 32 bits 倍數時，進行補足的位元。

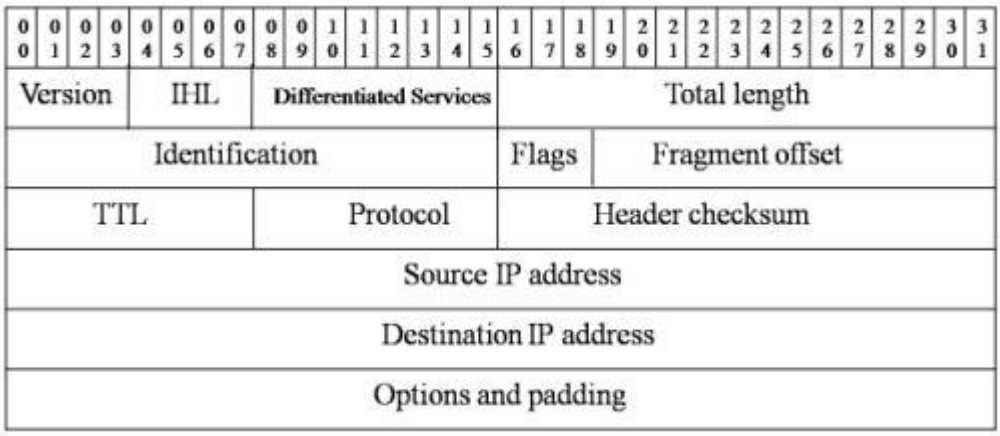


圖 18、IP 表頭

我們的設計是使用 Identification 欄位的 16 bits，做為我們標記資訊放置的位置，且使用 Flags 欄位的保留設定位元，有 1bit 來識別是否以經經過第一台 MPC 標記過。Flags 總共佔 3bits，如圖 19，主要與 IP 封包的切割與重組有關，第一個 bit 為保留用途用，預設值為 0，第二個 bit 為 DF(Don't Fragment)，用來定義 IP 封包是否可以加以切割，第三個 bit 為 MF(May Fragment)，用來定義此 IP fragment 是否為原始封包的最後一個 IP fragment。我們針對欄位的設定結果如

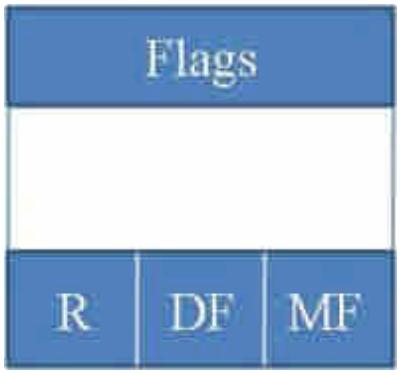


圖 19、Flags 欄位設計

圖 20、Identification 欄位

欄位說明:

(1) Identification: 16bits，只有對未切割過封包，才會將此欄位分成兩欄，分別記錄第一個經過的 MPC 的 IID 值，和目前所處的 MPC IID 值，範圍 0~255[註 1]。

(2) IID：8bits，標記內容，只有對未切割過封包才會在 Identification 欄位標記 IID 值，每經過在 gateway 附近的 MPC，就會將 MPC 特定的 IID 值放置在這個欄位，在相同網域內的 MPC 會有相同 IID 值，不同網域間，設定不同的 IID 值，IID 的範圍是從 0~255，然而因為 IID 要有未使用時的數值，因此將 0 保留下來，當作還未填 IID 資料進去，因此真正能夠使用的 IID 就只 1~255，如圖 20。

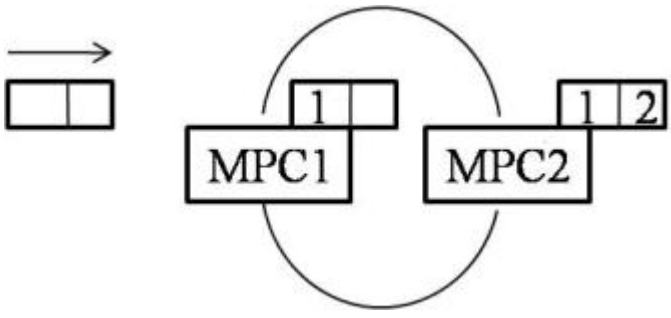


圖 21、標記示意圖

(3) R：有兩個功用，對於未切割過封包是用來判斷是否已經經過第一個 MPC 標記過，R 等於 0，表示無，IID 值放到 IID1，R 等於 1，表示有，IID 值放到 IID2；對於切割過封包則是用來讓監聽程式判斷是否需要將相關資料寫進資料庫，R 等於 0，需要 ，R 等於 1，不需要。

在為每一個封包進行標記時，除了更改 IID 值外，也要針對特定欄位進行修改。Header Checksum 欄位代表標頭檢驗值，當我們修改了 IID 欄位的內容後，檢驗值必需要重新計算，放入一個修改過後的新值，避免在後續的傳送中，封包遭到丟棄。

註 1: 最早在 IP 設計上所提供的 QoS 層級解決方案, 其中之一就是 IP 標頭中的 Type of Service 欄位(TOS byte)。改變這些數值, 我們可以選擇一個 高/低 等級的吞吐量、延遲、或是可靠度。但是這並不能提供足夠的靈活性, 以滿足較新服務(如 real-time 應用程式、互動程式、和其它)的需求。有鑒於此, 新的架構出現了。其一就是 DiffSserv, 它會保留 TOS bits , 同時重新命名 DS 欄位。

4.2 標記偵測與記錄

封包偵測必須要針對每一個封包的 Flags 的保留設定位元，進行解析動作。對於未切割封包，Flags 的保留設定位元是用來判斷封包是否已經經過第一個 MPC 標記過，若保留設定位元值為 0，表示目前所處的 MPC 是此封包經過的第一台 MPC，會在標記後，再將標記內的所有 IID 及相關資料，記錄到資料庫內；若保留欄位值為 1，會在標記前，將標記內的所有 IID 和本身 IID 及相關資料，記錄到資料庫內。對於切割過封包，Flags 的保留設定位元是用來判斷封包是否已經經過出口端 MPC 的監聽程式將所需資料寫進資料庫，若保留欄位值為 0，表示目前所在的 MPC 是出口端，監聽程式需將本身 IID 及相關資料，記錄到資料庫內；若保留欄位值為 1，則監聽程式無須將任何資料寫進資料庫。

在每一台 MPC 上，我們都會架設一個 MySQL 資料庫，並且設計一個 Table，使監聽所偵測到的標記內容寫入到資料庫的 Table 內，而 Table 欄位設定如圖 21。

STime 與 ETime 兩個欄位的格式型態是 DATETIME，用來記錄日期以及時間，STime 全名為 Start Time，記錄這個標記內容的起始時間，而 ETime 全名為 End Time，記錄這個標記內容的結束時間，SIP 為來源 IP，DIP 為目的 IP，Protocol 為 IP 通訊協定，DPORT 為目的 Port，IIDNUM 為封包內標記的 IID 數量，IID1、

IID2、IID3 代表的是經過的 MPC 的各個 IID。

欄位名稱	型態	欄位名稱	型態
(1)STime	DATETIME	(6) DPORT	SMALLINT UNSIGNED
(2)ETime	DATETIME	(7) IIDNUM	SMALLINT UNSIGNED
(3)SIP	INT UNSIGNED	(8) IID1	SMALLINT UNSIGNED
(4)DIP	INT UNSIGNED	(9) IID2	SMALLINT UNSIGNED
(5)Protocol	SMALLINT UNSIGNED	(10)IID3	SMALLINT UNSIGNED

表格 18、MySQL 的 Table 設計

每當一個封包經過 MPC，具有標記的內容如果直接寫入資料庫，會使記錄資料大量產生，短時間內擁有數筆相同的數據資料，為了將這些重覆資料在短時間內能夠集合成同一筆資料，我們在監聽的部份加入 Buffer，使標記的資料能夠在短時間保留在 Buffer 內，直到 Buffer 滿載或者一段時間後，再寫入資料庫，以減少短時間內相同資料的產生。Buffer 是使用動態產生，並且擁有一個數值限制 Buffer 的最大量，並且可以設定保留在 Buffer 的時間。

當一個標記資料進入 MPC 後，它會先將這筆標記資料記錄放到 Buffer，並且會在 Buffer 內的 STime 與 ETime 寫入現在的時間與其它數值，當下一筆標記資料進入時，在 Buffer 中如果也擁有這筆相同資料時，它會將 ETime 更新為目前時間，直到 STime 超過我們所設定的時間限制，就會寫入資料庫。或者是目前的 Buffer 已達到最大量，我們就將 Buffer 中 STime 最舊的那一個，寫入資料庫。

假如是切割過的封包，即使沒做標記，出口端 MPC 的監聽程式依舊要將其寫入資料庫，作類似 logging 的動作，且 IIDNUM 為 1，IID1 為本身 IID，而 IID2、IID3 皆為 0。

4.3 資料庫記錄縮減

為了將標記內容寫入資料庫的量縮減，我們會在監聽時，藉由檢查 TCP 封包是否完成三向交握程序，再決定是否將其記錄到 Buffer。將沒完成三向交握的連線，記錄為需要寫進 Buffer 的連線封包；若完成三向交握程序的連線，在此連線傳送的封包將不予以記錄到 Buffer，藉此可減少寫入資料庫的資料量，演算法

見圖 22，修改 mylistener.c 程式，以達到資料庫量再縮減的目標，mylistener.c 會記錄哪些 flow 已經完成三向交握程序，並將此 flow 的 source IP、source port、destination IP 和 destination port 存入陣列，以供 mylistener.c 判斷封包是否需要先記錄到 Buffer 的依據。

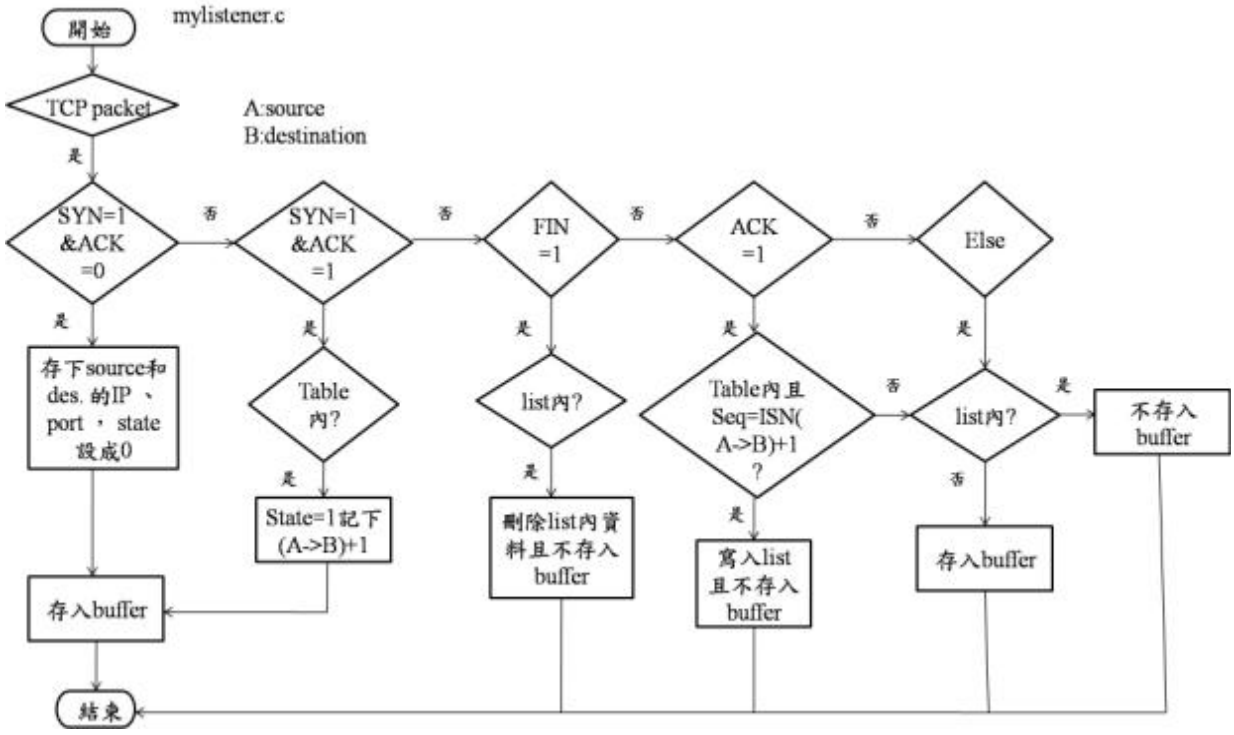


圖 22、mylistener.c 演算法修改

Table

Source IP	Destination IP	Source port	Destination port	ISN(A->B)+1	state

list

SIP	DIP	SPORT	DPORT

表格 19、Table 和 list

監聽封包程式:

(1)若監聽到的為 SYN 封包，則將 Source 和 Destination 的 IP 、port 存入 table(由六個一維陣列組成，見圖 22)，寫入 Buffer 暫存區。

(2)若監聽到的為(SYN+ACK)封包，則先檢查在 table 內是否有記錄，若有，則修改相對應的欄位，將 state 設為 1，且記錄下此次的 ACK 值在 ISN(A→B)+1 欄位，寫入 Buffer 暫存區。

(3)若監聽到的為 FIN 封包，則先檢查在 list 內是否有記錄，若有，則刪除 list 內資料，且不存入 Buffer。

(4)若監聽到的為其它 ACK 封包，則先檢查在 table 內是否有記錄， 若有，再檢查 Seq 是否等於記錄在 ISN(A→B)+1 的值，若相等，則將 source 和 destination 的 IP、 port 記錄在 list 內，且不存入 Buffer，若不相等，則檢查 list 有無記錄，若無，則寫入 Buffer，若有，則不寫入 Buffer。

(5) 若監聽到的為其它封包，則檢查 list 有無記錄，若無，則寫入 Buffer，若有，則不寫入 Buffer。

4.4 路徑追蹤

路徑追蹤的方式由一台中央控管主機(MCC)發出查詢，在 MCC 會建一個資料庫，有所有 MPC IID 和 subnet、IP、Gateway 的對照表，由輸入的條件之一—封包的 destination IP，由 destination IP 得知向特定 subnet 裡的 MPC 做資料查詢，MPC 依照中央控管主機所設定的查詢條件，由本身的資料庫找到前一個 MPC IID 值，再跟前一個 subnet 裡的 MPC 作查詢，陸續往回查詢相關的 MPC，直到此封包經過的第一台 MPC，即可將資料回傳至 MCC 呈現，路徑的呈現方式是以 gateway 呈現。

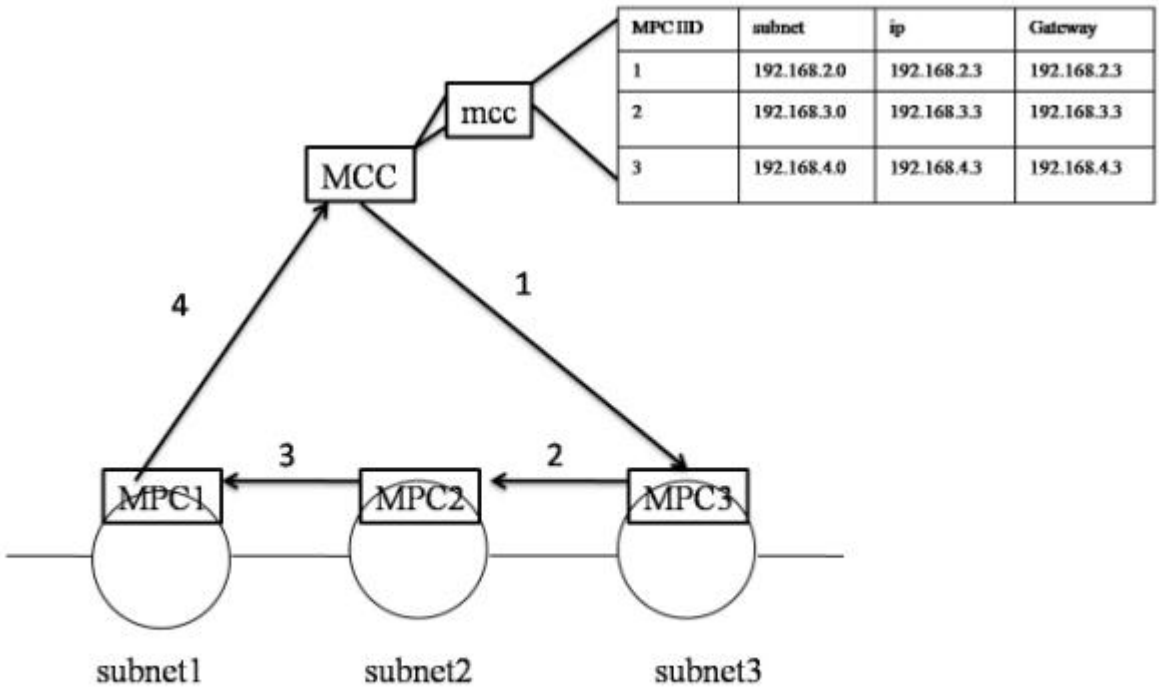


圖 23、Traceback 方式

從圖 23 中，了解路徑重建的步驟。

Step1:MCC 依據查詢條件 destination IP，知道要向 subnet3 裡的 MPC3 做查詢。

Step2~3:MPC3 根據資料庫內的資料，知道要向 MPC2 做查詢，MPC2 根據資料庫內的資料，知道要向 MPC1 做查詢，因為 MPC1 是第一個經過的 MPC，所以查詢做到此即可。

Step4: MPC1 作回傳資料的動作，回傳路徑給 MCC。

找出整條路徑的 IID 後，需要將 IID 轉換為 gateway，這時藉由中央控管主機內的 MPC IID 與 gateway 相對應的記錄，將 IID 轉回所處的 gateway。

有關於切割過的封包，則是由MCC向全部MPC查詢是否有此筆記錄，找出出口端gateway，在MCC呈現出入口gateway。

第五節、系統設計

5.1 系統整體架構

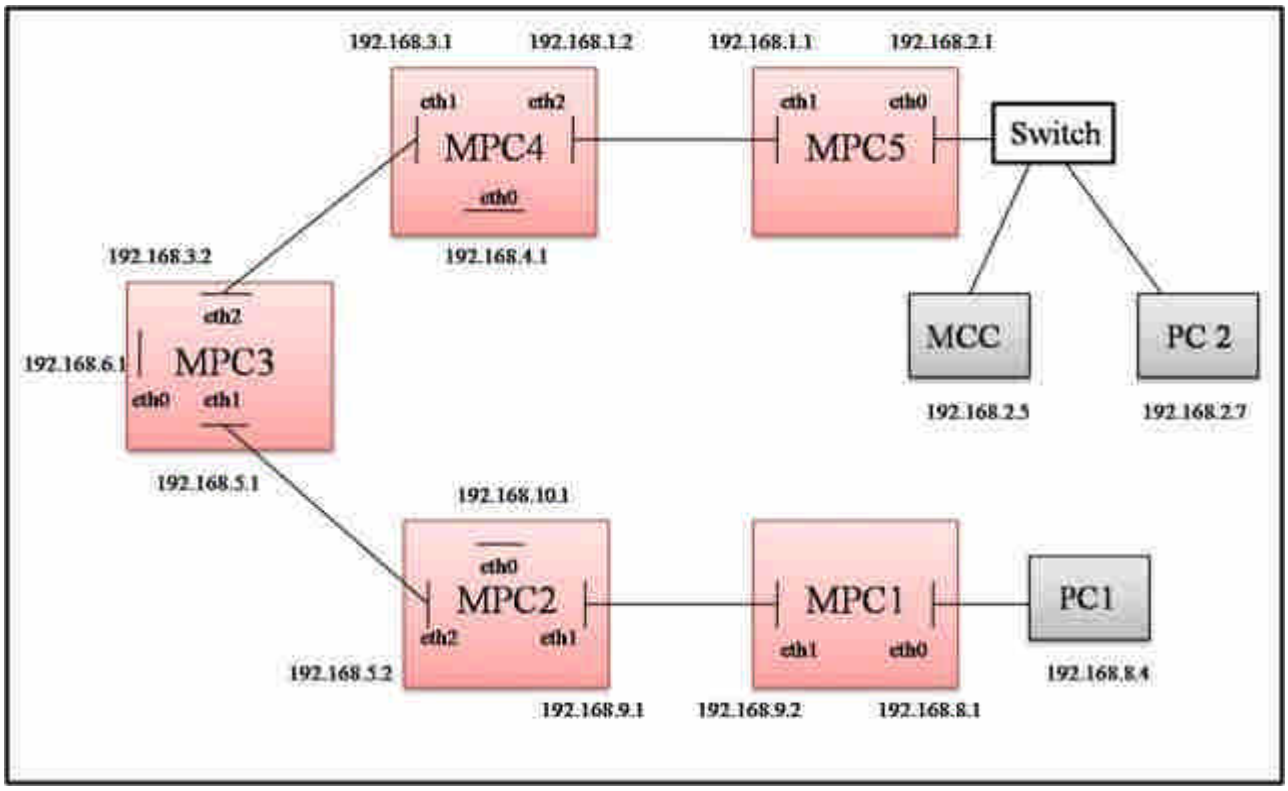


圖 24、封包標記與追蹤系統硬體架構圖

系統硬體組成：

5 台 MPC、2 台 PC 和一台 Switch。

系統內容：

	軟體	• Linux – Ubuntu 8.04
		• Java SE Development Kit (JDK) 6
		• MySQL Database
		• Quagga
MPC	硬體	• Intel E7200
		• MD Gigabyte EP35-DS3L P35/ICH9
		• 4GB DDRII 800 RAM
		• VGA GeForce 7200 series (128M)
		• 640GB HD

		• PSU 350W
		• DVD-ROM
		• Network Interface Card(NIC) x 2
	軟體	• Microsoft Windows XP
		• Java SE Development Kit (JDK) 6
	硬體	• Intel E7200
		• MD Gigabyte EP35-DS3L P35/ICH9
		• 4GB DDRII 800 RAM
MCC		• VGA GeForce 7200 series (128M)
		• 640GB HD
		• PSU 350W
		• DVD-ROM
		• Network Interface Card(NIC) x 1
	軟體	• Microsoft Windows XP
	硬體	• Intel E7200
		• MD Gigabyte EP35-DS3L P35/ICH9
		• 4GB DDRII 800 RAM
PC		• VGA GeForce 7200 series (128M)
		• 640GB HD
		• PSU 350W
		• DVD-ROM
		• Network Interface Card(NIC) x 1

5.2 各主機功能

MPC
- 封包標記
- 封包轉送
- 記錄封包標記
- 資料庫存取
- 自動路由功能

MCC　　・Java 程式執行

・路徑查詢

PC　　・上網功能

封包標記：將傳送進來的封包進行封包標記。

封包轉送：將標記過後的封包進行轉送。

記錄封包標記：將傳送進來的封包標記內容讀取出來，記錄到資料庫。

資料庫存取：資料庫存取只能由本地端進行存取。

Java 程式執行：路徑追蹤 Java GUI 介面的執行。

5.3 軟體架構

MPC 軟體架構，在安裝好的 Ubuntu 8.04 中，核心部份改由修改過後的 Kernel 2.6.24.6 版本，並安裝修改過後的程式碼 Bridge modules 與 MySQL 資料庫，最後再安裝 Apache Server，軟體整體架構圖如圖 25 所示。

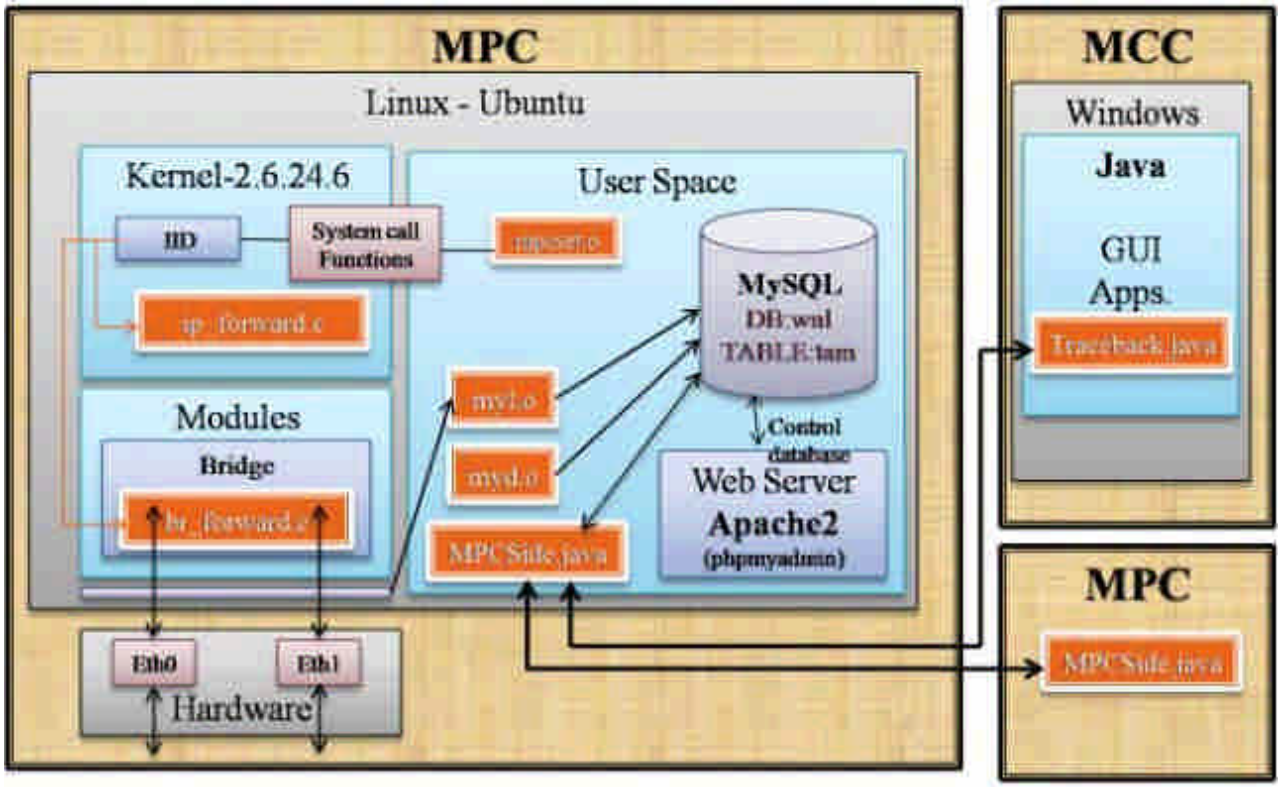


圖 25、MPC 軟體架構圖

Kernel-2.6.24.6 部份，IID 為核心的變數，mpcset.o 是在使用者環境下所使用的程式，mpcset.o 程式可進行變更核心的變數。Modules 部份，使用修改過後

模組，使封包經由 Bridge 模組進行標記。如果加入自動路由功能，則是在核心檔案 ip_forward.c 進行標記。在使用者方面，myl.o 是常駐執行程式，進行封包標記的記錄，myd.o 同樣是常駐程式，進行資料庫定時清理功能。

MCC 透過 Traceback.java 的圖型介面，進行路徑追蹤，由 Socket 將查詢的內容傳送至 MPC，之後由 MPCSide.java 陸續向其它 MPC 進行路徑回追，再將路徑重建結果回傳給 MCC。

第六節、封包標記實作

6.1 硬體設備

每一台 MPC 的硬體設備都採用相同規格，其設備如表六-1。一共使用了五台當 MPC，以及使用者主機。

硬體名稱	規格內容
CPU	Intel E7200
Motherboard	Gigabyte EP35-DS3L P35/ICH9
RAM	威剛 DDRII 800 2GB x2
VGA	GeForce 7200 series (128M)
HDD	WD 6400AAKS 640GB
PSU	全漢 藍晶鑽 350W APFC
NIC	2 Card

表格 20、MPC 硬體規格

6.2 硬體需求與安裝設定

軟體主要需求是作業系統，使用 Ubuntu 8.04 LTS，其 Linux 核心為 2.6.24.6 版本，將 Ubuntu 8.04 進行系統安裝，系統安裝完畢後，將後續需要使用到的也安裝上去，所需要的有 g++、libncurses5-dev、bridge-utils、mysql-server-5.0、phpmyadmin、libmysqlclient15-dev 和 quagga，由指令視窗下答安裝指令，所要安裝的指令如下：

 # apt-get install g++

 # apt-get install libncurses5-dev

 # apt-get install bridge-utils

 # apt-get install mysql-server-5.0

 # apt-get install phpmyadmin

 # apt-get install libmysqlclient15-dev

#apt-get install sun-java6-jdk

#apt-get install quagga

將上述程式安裝完畢後，後續才能進行安裝、修改與設定。

6.3 系統核心修改及編譯

修改核心的目的，主要是為了新增自訂的變數，並且讓使用者自由修改變數內容，因此需要在核心新增變數與讀取的函式，還有因為 MPC 如果加入動態路由的功能當路由器，封包不會經過 bridge module 標記，所以需要去修改核心的 ip_forward.c，使封包經過核心時做標記。要加入的變數有一個，就是 MYIID，每一台 MPC 所具有的編號。

增加這一個變數，可以在修改 IID 時，不需要重新編譯核心，達到有彈性的效果。

增加變數後，同時也要增加函式，讓使用者在 user space 能夠進行資料的讀取與修改，因此就需要編寫 system call 的函式，讓它能夠由這些函式修改這個變數內容。

首先要取得核心程式碼，上 www.kernel.org 進行 kernel 的下載，我們所下載的核心為 2.6.24.6 的 full source，將 source code 下載完後解壓縮，並放置在資料夾為/usr/src/linux-2.6.24.6 底下。

在資料夾/usr/src/linux-2.6.24.6/include/linux/內新增一個自訂的標頭檔”mysyscall.h”，主要的變數宣告，其內容如下：

```
#ifndef __MYSYSCALL__
#define __MYSYSCALL__
   unsigned int MYIID;
#end if
```

找到原本存在的檔案/Linux-2.6.24.6/arch/x86/kernel/syscall_table_32.S，在最後加上我們要新增的 function 名稱：

```
.log sys_setIID          /* 325 */
.log sys_getIID          /* 326 */
```

再找到另二個檔案/Linux-2.6.24.16/include/asm-x86/unistd_32.h 和 /usr/include/asm/unistd_32.h，找到有一行#define __NR_fallocate 324 後加上我們剛自訂的 function 上去，內容如下：

```
#define __NR_setIID    325
#define __NR_getIID    326
```

函式的宣告部份設定好後，開始編寫我們自訂的函式，在資料夾

/Linux-2.6.24.6/arch/x86/kernel/底下，新增檔案”mysyscall.c”，然後編寫函式：

```
#include <linux/mysyscall.h>
#include <linux/linkage.h>
asmlinkage void sys_setIID(unsigned int theIID)
{    MYIID = theIID; }
asmlinkage unsigned int sys_getIID(void)
{    return MYIID; }
```

然後再修改/Linux-2.6.24.6/arch/x86/kernel/Makefile_32，加入指令，” obj-y += mysyscall.o”，使 c 檔能夠進行 compiler 後出現 mysyscall.o 檔。

為了能夠讓 module 能夠直接存取所設定的變數，要將變數 EXPORT 出來，在資料夾/Linux-2.6.24.6/arch/x86/kernel/，修改 i386_ksyms_32.c，在這支程式碼最底下，加入：

EXPORT_SYMBOL(MYIID);

之後再編寫初始化的內容，修改/Linux-2.6.24.6/init/main.c，在全域部份宣告 extern int MYIID，而在 start_kernel 的函式內的 check_bugs();之前加入 MYIID=1 使變數能夠擁有初始值，到這兒變數的宣告與函式設定就修改完成了。

在 kernel 的 source code 修改完畢後，我們要進行核心編譯，將目前目錄改至/usr/src/linux-2.6.24.6/資料夾底下，核心編譯需要以下步驟：

(1) # make　clean

將原本已經 compiler 過的檔案清除，以確保舊檔案仍保留住。

(2)# make　menuconfig

主要的核心設定，必須根據硬體設備去進行設定，不同的硬體會有不同的設定，因為支援的不同，這兒的設定若沒有設好，會造成無法開機或者其它的一些問題。

(3)# make　bzImage

完全編譯內核。

(4)# make　modules

編譯模組

(5)# make　modules_install

安裝所編譯的模組

(6)# mkinitramfs　–o　/boot/initrd.img-2.6.24.6 2.6.24.6

生成 initrad 到/boot 資料夾下

(7)# cp /usr/src/Linux-2.6.24.6/arch/i386/boot/bzImage /boot/vmlinuz-2.6.24.6

複製編譯後的 bzImage 到/boot 資料夾並改名為 vmlinuz-2.6.24.6

(8)# cp /usr/src/Linux-2.6.24.6/System.map　/boot/System.map-2.6.24.6

複製 System.map 至/boot 資料夾

(9)# update-grub

更新開機選單

(10)# reboot

重新開機

以上步驟完成後，重新開機後，在開機選單上會多一個剛所新增的 kernel 選單，如果沒有出現，就必須要確認一下/boot/grub/menu.lst 這一個檔案，自行將開機選項加入。

6.4 路由安裝與設定

為了測追蹤路徑的功能，我們將 MPC 加入動態路由的功能，使用軟體 Quagga 達此目的，使用 RIP v2 來作為路由協定，以下為 Quagga 的設定程序。

(1)先安裝 Quagga

安裝完 quagga 之後，他會有好幾隻 daemon，其中 zebra 負責去修改系統裡面的 routing table，ripd 負責接收和發送 routing update，並且告知 zebra 如何去修改 routing table。

(2)#vim /etc/quagga/daemons

進入編輯模式輸入 i

離開編輯模式輸入 Esc

存檔離開輸入 :wq

編輯/etc/quagga/目錄下的 daemons 將 zebra ，ripd 改為 yes 存檔

zebra=yes

bgpd=no

ospfd=no

ospf6d=no

ripd=yes

ripngd=no

(3)

#cp /usr/share/doc/quagga/examples/zebra.conf.sample /etc/quagga/zebra.conf

#cp /usr/share/doc/quagga/examples/ripd.conf.sample /etc/quagga/ripd.conf

複製設定檔 zebra.conf.sample 和 ripd.conf.sample 到/etc/quagga 目錄下，並各別存檔為 zebra.conf、ripd.conf，不要更改檔名。

(4)修改設定檔

#vim /etc/quagga/ zebra.conf

用來設定主機網卡的 ip 位址和 netmask，如果有要修改 ip，只要修改 interface ethX 下面那一行指令即可，若要刪除或新增某個網卡只要刪除或新增

interface ethX

ip address X.X.X.X/X

ipv6 nd suppress-ra

!

!

! Zebra configuration saved from vty

! 2009/08/12 09:57:34

!

hostname Router

password zebra

enable password zebra

```
!
interface eth0
 ip address 192.168.6.1/24 //設定 eth0 的 ip 和 netmask
 ipv6 nd suppress-ra
!
interface eth1
 ip address 192.168.5.1/24
 ipv6 nd suppress-ra
!
interface eth2
 ip address 192.168.3.2/24
 ipv6 nd suppress-ra
!
interface lo
!
ip forwarding
!
!
line vty
```

#vim /etc/quagga/ ripd.conf

設定哪些網卡負責哪些網域的動態路由，若要修改只要刪除或新增

```
network X.X.X.0/24   //網卡 X 負責的網域
network ethX
```

```
!
! Zebra configuration saved from vty
!     2009/08/12 10:00:06
!
hostname ripd
password zebra
log stdout
!
```

```
router rip
  version 2
  network 192.168.3.0/24
  network 192.168.5.0/24
  network 192.168.6.0/24
  network eth0
  network eth1
  network eth2
!
line vty
!
```

不只要修改設定檔，還要進行 GUI 介面的網路設定，可點選右上的網路設定(用紅線框起來)，或由左上點選系統→管理→網路，進行網卡介面設定，如圖 26。

圖 26、網路設定

出現網路設定介面後，再點選解除鎖定，輸入密碼，點選認證，即可進行每張網卡的設定，如圖 27。

圖 27、網路連線設定

再點選要設定的網卡→屬性，設定選固定 IP 地址，輸入要給網卡的 IP 地址和子網域遮罩，按確定，完成網卡的設定，如圖 28。

圖 28、網路 IP 設定

有時 GUI 介面雖有設定網卡的 IP，但是網卡實際上沒被分配到 IP，會導致路由設定無法成功，這時可從終端機登入，輸入 ifconfig，檢查在主機上所有網卡介面的相關資訊，若是 eth0 IP 設定沒成功，可輸入指令 ifconfig eth0 X.X.X.X，分配 IP 給 eth0，如圖 28。

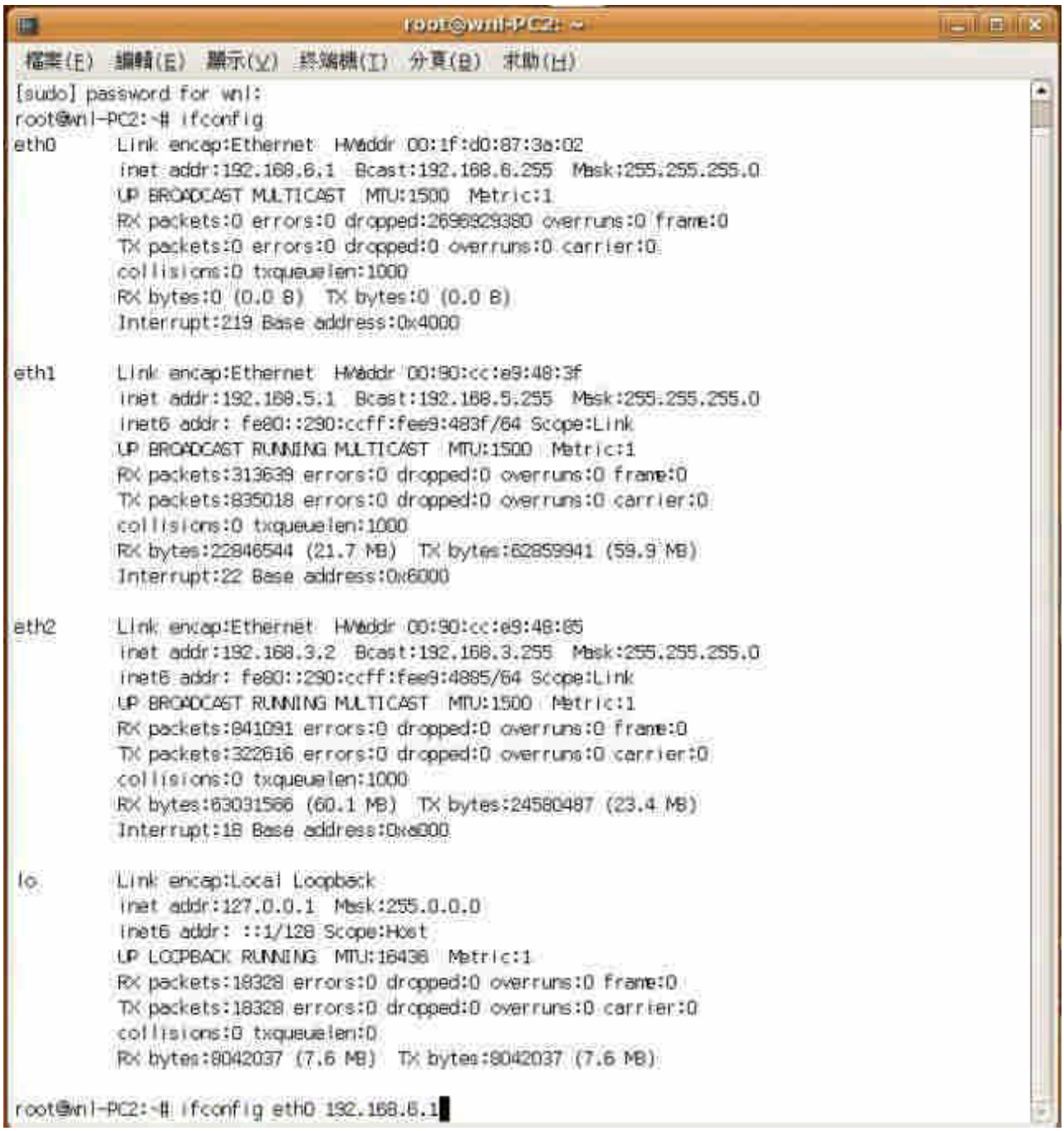


圖 29、網路資訊

(5)#/etc/init.d/quagga restart

啟動 quagga 軟體，進行路由封包的交換，主機重開機後，quagga 會自動執行，若是想暫停可輸入#/etc/init.d/quagga stop。

(6)設定完後，可用 ping 指令，檢查設定是否完成，也可輸入 route –n 檢查各主機內的路由表。

(7)若是想要主機當 bridge，則在 GUI 網路介面設定，每張網卡的網路設定皆點選啟用漫遊，如圖 29，並暫停 quagga 軟體(#/etc/init.d/quagga stop)，進行 bridge 指令設定。若要從 bridge 換回 router，讓主機重開機即可，請不要把建 bridge 指令寫進/etc/rc.local。

圖 30、eth0 屬性

6.5 核心修改

在 MPC 當作 router 傳送的時候，是經由內核所運作的，因此我們要修改傳送封包內容時，就要在傳送封包的流程中更改程式，讓它在經過 MPC 後，能夠增加封包標記內容。

在核心中，封包進行傳遞時，會經過 ip_forward.c 檔案內的 ip_forward 這個函式，利用它所經過的途徑，將所要標記的程式碼寫入這個函式內，就可以達到封包修改的目的。

程式執行的順序：

(1) 檢測封包是否經過切割

(2) 若是切割封包，則檢查 Flags 欄位的 R-bit 是否為 1，若是，則直接離開，若不是，則將 R-bit 設為 1，重新運算 checksum，然後離開。

(3) 若是未切割封包，檢測封包是否已經擁有第一筆標記記錄，若無，則加入第一筆標記內容到 IID1，並將 R-bit 設為 1，重新運算 checksum，然後離開，若有，則在 IID2 加入標記內容。

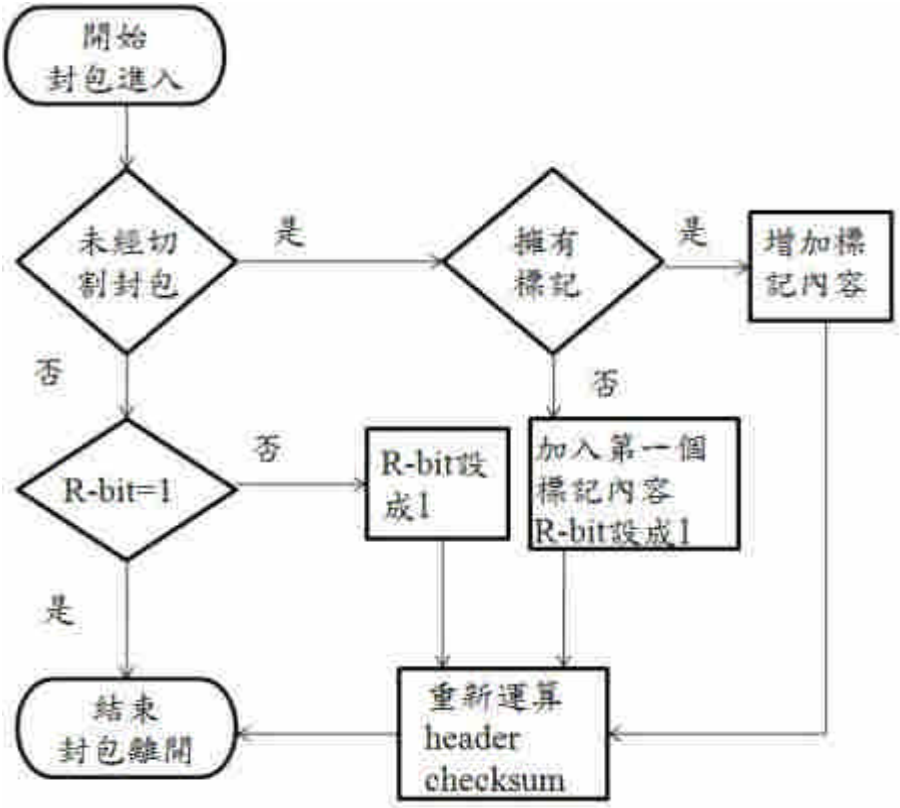


圖 31、標記演算法流程

ip_forward 函式的程式碼：

```
int ip_forward(struct sk_buff *skb)
{
      struct ethhdr *ethh;      //marking
      struct iphdr *iph;  /* Our header */
      struct rtable *rt;    /* Route we use */
      struct ip_options * opt  = &(IPCB(skb)->opt);


        ethh = eth_hdr(skb);      //marking
```

```
/***      skip      ***/

        //additional marking code
        if(ntohs(ethh->h_proto) == ETHERTYPE_IP&& ((iph->protocol)==6 ||
(iph->protocol)==17)){
            iph = ip_hdr(skb);
                if(((ntohs(iph->frag_off))&0x3FFF)==0x0000){ //MF=0, offset=0
                  if(((ntohs(iph->frag_off))|0x7FFF)==0xFFFF){ // reserved=1
                                iph->id=(iph->id&0x00FF)|htons(MYIID) ;//add
marking content in IID2
                                ip_send_check(iph);


                  }
                     else if((ntohs(iph->frag_off)&0x8000)==0x0000){ //reserved=0
                                iph->id=(iph->id&0x0000)|MYIID;//add first
marking in IID1
                                iph->frag_off=htons(0x8000)^iph->frag_off;
//reserve=1
                                ip_send_check(iph);
                     }
                }
                else
if((((((ntohs(iph->frag_off))&0x2000)==0x0000)&&(((ntohs(iph->frag_off))&0x1
FFF)!=0x0000)))
                                &&((ntohs(iph->frag_off)&0x8000)==0x0000)){
                                //fragmented packet { (MF=0 offset!=0)} and
reserved=0
                                iph->frag_off=htons(0x8000)^iph->frag_off;
//reserve=1
                                ip_send_check(iph);
                }
      }
```

```
        // additional marking code end
            /***      skip      ***/
    }
```

6.6 模組修改及安裝

在 MPC 當作 bridge 傳送的時候，是經由模組內的 bridge module 所運作的，因此我們要修改傳送封包內容時，就要藉由所經過的模組，在傳送封包的流程中更改程式，讓它在經過 MPC 後，能夠增加封包標記內容。

在 bridge 模組中，封包進行傳遞時，會經過 br_forward.c 檔案內的 __br_forward 這個函式，利用它所經過的途徑，將所要標記的程式碼寫入這個函式內，就可以達到封包修改的目的。

程式執行的順序：

(1) 檢測封包是否經過切割

(2) 若是切割封包，則檢查 Flags 欄位的 R-bit 是否為 1，若是，則直接離開，若不是，則將 R-bit 設為 1，重新運算 checksum，然後離開。

(3) 若是未切割封包，檢測封包是否已經擁有第一筆標記記錄，若無，則加入第一筆標記內容到 IID1，並將 R-bit 設為 1，重新運算 checksum，然後離開，若有，則在 IID2 加入標記內容。

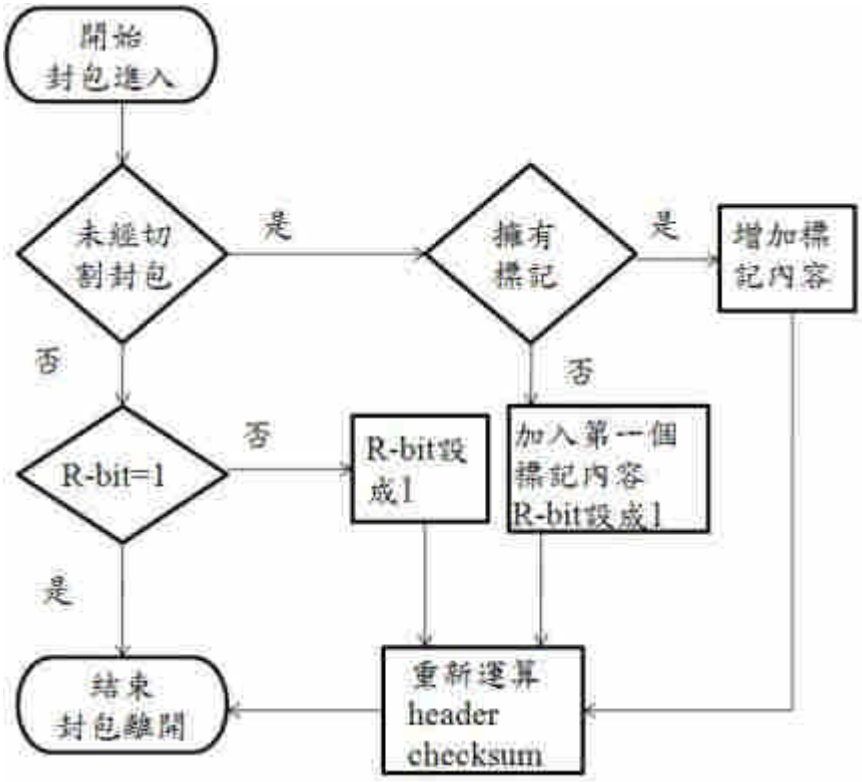


圖 32、標記演算法流程圖

__br_forward 函式的程式碼：

```
static void __br_forward(const struct net_bridge_port *to, struct sk_buff *skb)
{
        struct net_device *indev;
        //add my thing
        struct ethhdr *ethh;
        struct iphdr *iph;

        indev = skb->dev;
        skb->dev = to->dev;
        skb_forward_csum(skb);

        ethh = eth_hdr(skb);
        if(ntohs(ethh->h_proto) == ETHERTYPE_IP){      //ip packet
                iph = ip_hdr(skb);
                        if(((iph->protocol)==6 || (iph->protocol)==17)){ //tcp or udp packet
                                if(((ntohs(iph->frag_off))&0x3FFF)==0x0000){    //MF=0,
```

```
offset=0
                    if(((ntohs(iph->frag_off))|0x7FFF)==0xFFFF){    //
reserved=1
                        iph->id=(iph->id&0x00FF)|htons(MYIID) ;    //add
marking content in IID2
                        ip_send_check(iph);
                    }
                    else
if((ntohs(iph->frag_off)&0x8000)==0x0000){    //reserved=0
                        iph->id=(iph->id&0x0000)|MYIID;    //add first
marking in IID1
                        iph->frag_off=htons(0x8000)^iph->frag_off;
//reserve=1
                        ip_send_check(iph);
                    }
                }
            else
if((((((ntohs(iph->frag_off))&0x2000)==0x0000)&&(((ntohs(iph->frag_off))&0x1
FFF)!=0x0000)))
                        &&((ntohs(iph->frag_off)&0x8000)==0x0000)){
                        //fragmented packet { (MF=0 offset!=0)} and
reserved=0
                        iph->frag_off=htons(0x8000)^iph->frag_off;
//reserve=1
                        ip_send_check(iph);
                }
            }
    }

    // end of add


    NF_HOOK(PF_BRIDGE, NF_BR_FORWARD, skb, indev, skb->dev,
```

br_forward_finish);

}

修改完模組後，再進入/usr/src/linux-2.6.24.6 的資料夾底下，然後執行 make modules 以及 make modules_install，就能夠將修改好的模組安裝進去，最後再重新開機就能夠執行新的模組，也就能夠修改封包標頭。

模組修改完後，接下來是需要設定，假如我們要將目前的網路卡 eth0、eth1 與 eth2 設為相同的傳輸介面，設定的執行指令：

(1)# ifconfig eth0 0.0.0.0

eth0 設為廣播

(2)#ifconfig eth1 0.0.0.0

eth1 設為廣播

(3) #ifconfig eth2 0.0.0.0

eth2 設為廣播

(4)#brctl addbr mybridge

新增一個 mybridge 橋接器介面

(5)#brctl addif mybridge eth0

將 eth0 加入 mybridge

(6)#brctl addif mybridge eth1

將 eth1 加入 mybridge

(7)#brctl addif mybridge eth2

將 eth2 加入 mybridge

(8)#ifconfig mybridge up

啟動 bridge

MPC 由上面的指令執行後，mybridge 就能夠當 bridge 啟動封包轉送，另外需要注意的是要確認”/proc/sys/net/ipv4/ip_forward”的值為 1，否則也無法進行轉送的動作。

而若要讓這台 MPC 也能夠讓其它台電腦進行存取，我們就要設一個 IP 在這台 MPC 上，而且是在它所傳送的這個網域上的 IP，設定的方式則是在第八項修改為”#ifconfig br0 192.168.0.2 netmask 255.255.255.0 up”之後這台 MPC 的 IP 位址就為 192.168.0.2。

6.7 系統核心與使用者介面的溝通

之前在核心部份，新增了變數以及讀寫變數的函式，使用者使用這些讀寫的函式，我們稱為使用”system call”，system call 所指的是由 user space 去讀取 kernel space 系統變數的意思，而使用者要去讀取變數，首先要在程式碼內加入 system call function 的所在位置及 library，然後由函式取得變數進行控制。

要控制的變數有一個—IID。IID 是用來控制所要標記的數值。

為了使用上的方便，我們編寫了一個使用者介面的程式，原始檔為”mpcset.c”，編譯過後為”mpcset.o”，我們使用 mpcset.o 進行修改及設定 IID。在修改的過程中，我們也將修改的結果寫入設定檔”mpc.config”，當我們改變 IID 內容時，設定檔也會跟著將內容改變。

這個執行檔所能執行的指令如圖 34，使用 help 可以查詢參數使用方式。

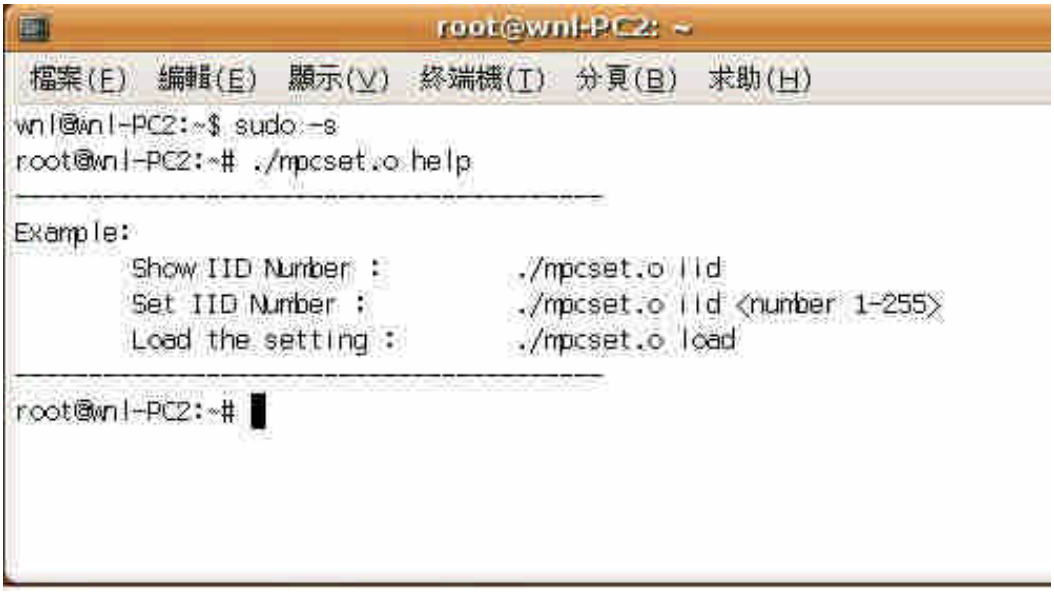


圖 33、執行 mpcset.o help 查詢可執行指令

執行 IID 參數內容的修改使用如圖 34。

圖 34、執行參數 IID

執行 load 參數使用如圖 35，主機若是重開機，需執行./mpcset.o load，才能載入上次的 IID 設定值，否則會是預設值 1。

圖 35、執行參數 load

在程式碼執行的部份，首先是依所下的參數決定之後所要處理的項目，之後藉由參數的傳遞，將所要改變的數值，使用 system call 的 function，進行核心變數的修改，達到使用者能夠自由變更標記的 IID。

mpcset.c 的部份程式碼：

```
/***      skip      ***/

void loadfile(){
    FILE *input;

    int iid;

    if((input=fopen("/home/wnl/mpc.config","r"))==NULL){
```

```
            fprintf(stderr,RED"File mpc.config is not found!\n"NORMAL);
            exit(-1);
        }

        // read IID first
        if(fscanf(input,"%d\n",&iid)==EOF){
            fprintf(stderr,RED"File mpc.config is not correct
context!\n"NORMAL);
            fclose(input);
            exit(-1);
        }
        if(iid<1 || iid>255){
            fprintf(stderr,RED"File mpc.config is not correct
context!\n"NORMAL);
            fclose(input);
            exit(-1);
        }
        syscall(__NR_setIID,iid);
        fclose(input);
        printf(GREEN"Load mpc.config is finished!\n"NORMAL);
}

void IID(unsigned int iid){
        if(iid < 1 || iid > 255){
            fprintf(stderr,RED"IID Number out of range (1-255)\n"NORMAL);
            exit(-1);
        }
        syscall(__NR_setIID,iid);
        printf(GREEN"IID = %d\n"NORMAL,syscall(__NR_getIID));
        savefile();
}
int main(int argc, char **argv)
{
```

```
    /***      skip      ***/
    return 0;
}
```

6.8 資料庫設定

MySQL 資料庫設定主要是在於權限以及遠端存取的控制，我們使用 phpMyadmin 去進行網頁式設定。在先前的安裝設定，已經將 apache 以及 phpMyadmin 安裝完畢，所以我們可以從網頁輸入 http://127.0.0.1/phpmyadmin/ 進入的 MySQL 資料庫管理頁面，如圖 36 所示。

圖 36、phpMydmin 的登錄畫面

進入登錄畫面後，我們首先要新增我們要使用的資料庫 wnl，之後在資料庫 wnl 內新增 table 名稱為 tam，然後將欄位資料設定好，最後再進行權限的設定。

設定的順序如下：

(1) 首先建立資料庫，我們在此新增 wnl 資料庫，圖 37。

圖 37、建立新資料庫

(2) 進入 wnl 資料庫後，我們新增 table，名稱為 tam，圖 38。

圖 38、新增資料表

(3) 設定 table 的欄位，圖 39。

圖 39、欄位設定

(4) 接下來是進入權限設定，之後新增使用者權限設定，圖 40。

圖 40、進入權限設定與新增使用者

(5) 權限新增完畢後，因為新增了使用者，但並未讀取入權限區，所以再執行重新讀取權限讓它更新，圖 41。

圖 41、重新讀取權限

在資料庫、資料表和使用者都建立完畢後，要讓遠端電腦存取資料庫，還必須要修改一個設定檔，/etc/mysql/my.cnf 的這個設定檔內，我們要修改裡面一行為”bind-address = 127.0.0.1”，由於這行限制只有本機能夠存取資料庫，因此我們必須將它進行註解，使遠端可以存取，變為” #bind-address = 127.0.0.1”，整個資料庫的設定就全部完成了，如圖 42。

```
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
#bind-address           = 127.0.0.1
#
# * Fine Tuning
#
```

圖 42、註解原本的程式碼

6.9 監聽程式設計與運行

監聽的部份，當我們每一個封包都進行標記，會發現每經過一個封包就擁有一筆記錄，這樣的結果，除了增加記錄的延遲時間，也造成有很多重複性的資料，因此為了減少記錄，我們可以將相同的傳輸內容，先記錄在暫存區，以一個相同連結的傳輸封包來說，我們首先將它記錄到暫存區的 Queue 內，在六十秒間，相同的標記資料，只有更新相同記錄的最後時間，而不再產生新的一筆記錄，在六十秒後，將這筆記錄寫入資料庫，因此每一筆記錄，都是具有六十秒的暫存時間，而超過六十秒後的記錄，則成為一筆新的記錄。

在資料結構方面，我們使用兩個 linking list 同時為同一筆資料做 link，第一個 linking list 是用來做 Queue 使用，第二個 linking list 為了判斷相同資料的搜尋，如圖 43，上方是以時間為主的 Linking list，下方是以 IID 數量為主的 Linking list，在這兩個 linking list 中的資料，所有的資料在第一個 linking list 是相同於所有的資料在第二個 linking list，所以當資料寫入資料庫後，那筆資料在兩個 linking list 同時都會消失。

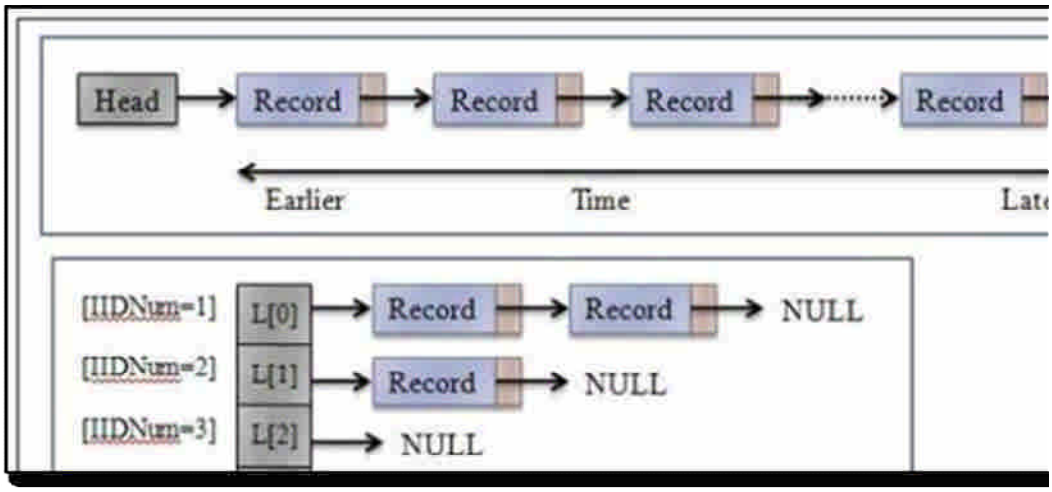


圖 43、封包監聽流程

在執行的時候，每次執行只能監聽一個網路卡，因此要監聽多個網路卡，必須要執行多次相同的程式並指定不同的網路卡，才能夠達到所有標記封包的監聽。在監聽的過程當中，只有當標記封包進入到網卡時，在未標記之前，將標記記錄寫到暫存區，然後在要記錄到資料庫之前，會將本身的 IID 再寫入到最後一個 IID 位置，因此所能夠監聽到的 IID 數量，最多為 3 個，監聽結果如圖 44。

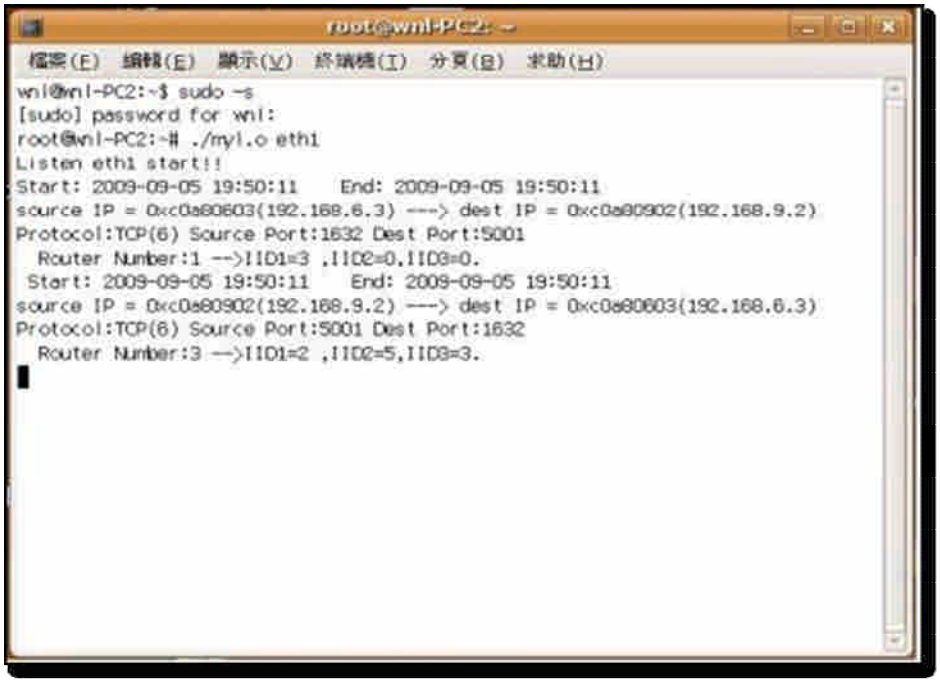


圖 44、網路卡封包監聽

6.10 結果呈現

執行的結果有兩部份，第一部份就是監聽程式結果寫入資料庫，第二部份是路徑自動回追的結果呈現。

在第一部份，監聽結果寫到資料庫，我們從 phpmyadmin 進入資料庫的資料表內，可以發現到監聽封包的結果寫入到資料庫內，如圖 45 所示。

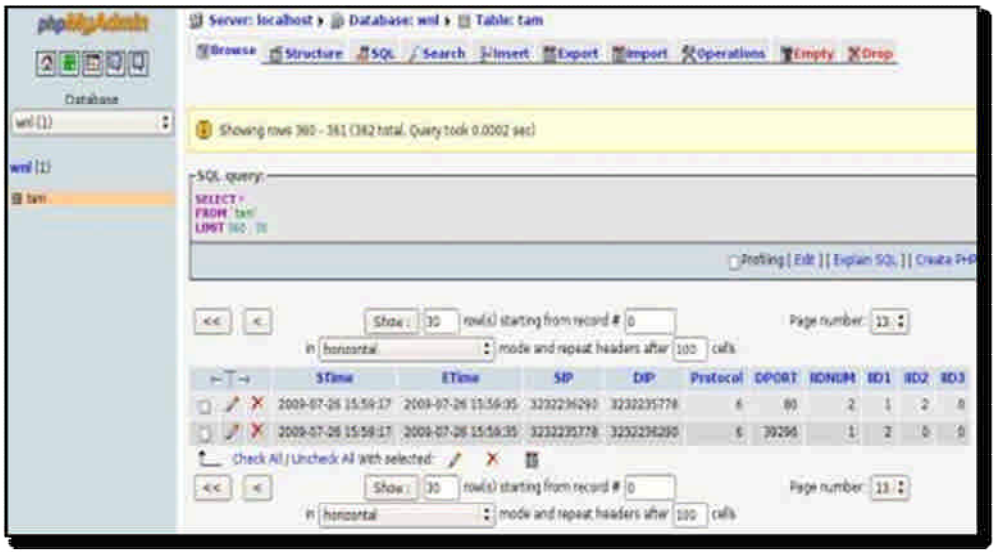


圖 45、資料表內容

在第二部份，是使用 Java 所編寫的 GUI 介面，如圖 46，輸入欲查詢封包的相關資訊，其中 Source IP、Destination IP、Destination Port 和 IP Protocol 為必

要輸入的查詢條件，而 Fragmentation、Date 和 Time 為可勾選項目，按下 Search，系統即會自動回追，重建封包經過的整條路徑，呈現畫面如圖 47。

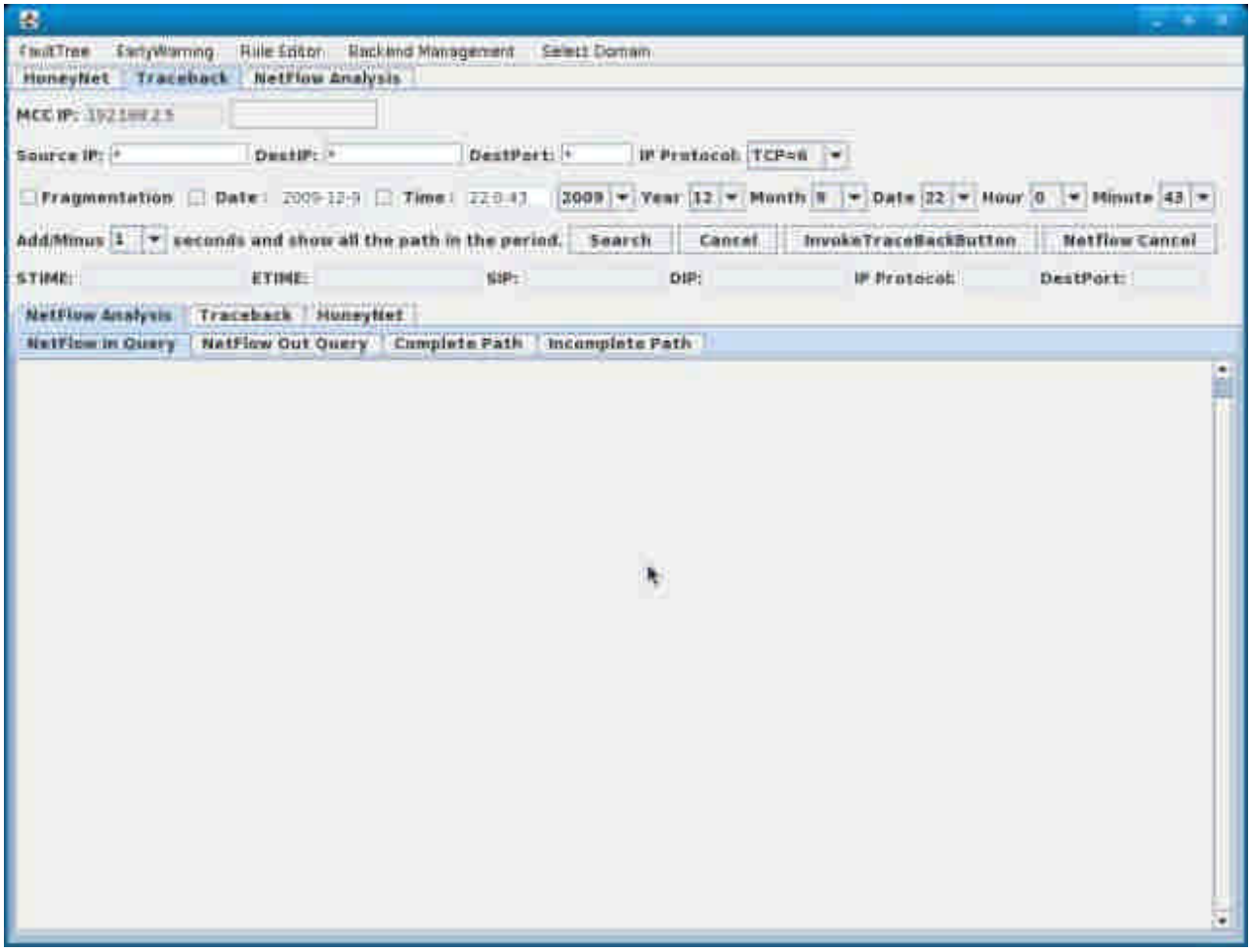
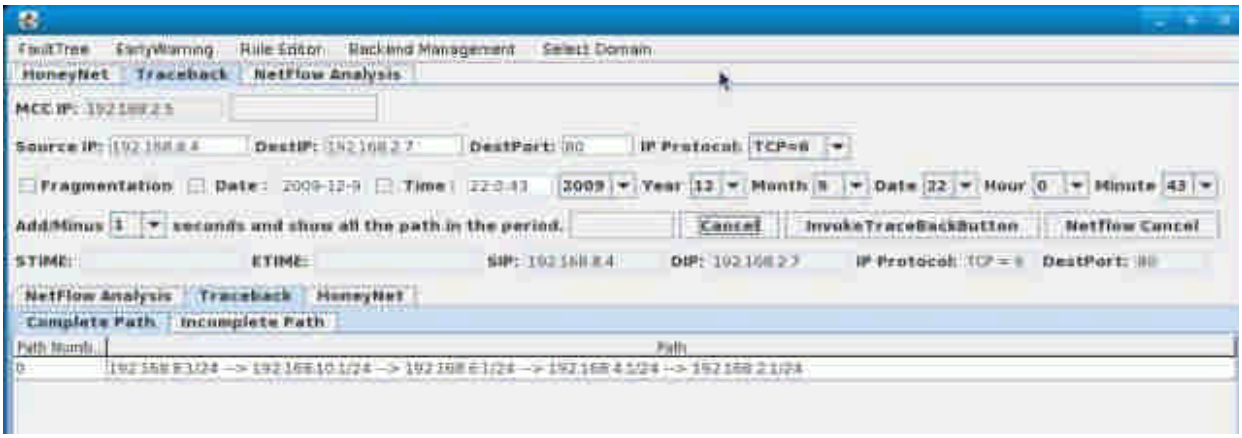


圖 46、Traceback GUI



1

圖 47、回追結果

第七節、未來展望

現有的通訊協定所進行的路徑追蹤，都是額外使用封包，針對傳送來的封包來源位址，去進行回送查詢的動作，但是傳送過來的封包有可能與傳送過去的封包是不同的路徑，當攻擊者是偽造 IP 攻擊時，我們則無法使用這種方法進行路徑追蹤，因此產生出封包標記的方法，以進行路徑追蹤。而我們使用 Identification 欄位去進行封包的標記，以不影響現有的網路通訊協定，不更改現有的網路設備，達成隱密性的效果，並使用 IID 取代原本的 IP address，以減少標記所需要的空間，同時也需要相對應的 IID 與 IP address 的對換表，找出完整的傳送路徑。

在設計過程中，我們考慮目前網際網路上，有許多路由器或防火牆都防止 IP Option 的使用，而導致封包在進行傳送時，遭受到丟棄，使網站與 host 端無法進行連線，為了解決這個問題，今年變更標記方式，使用 IP 表頭的 Identification 欄位和 Flags 的保留設定位元以解決此問題而使連線能順利成功，並達到標記的功能，達成我們封包傳輸的路徑追蹤。因為切割過的封包需要利用 Identification 欄位進行重組，所以不能動用 Identification 欄位，所以我們藉由 Flags 的保留設定位元，使得就算切割過的封包雖然不能進行標記，也能做到出入口端的查詢。

今年為了能夠自動回追路徑，我們也重新設計 GUI 介面和回追程式，使得路徑能夠根據查詢條件自動進行路徑回追，並呈現結果於介面上。且進一步跟其它兩個子計畫進行整合，根據其它子計畫的查詢請求，替其進行路徑回追，重建完整路徑，已達到去年未達成之目標。

第五章、網路風險分析與預警系統

此章節我們將詳細介紹在本計劃所開發的網路風險分析與預警系統。

第一節、前言

過去對於網路攻擊及異常行為之偵測大多需要事先找到攻擊或異常的樣式(pattern)或特徵(signature)，對於已知的攻擊行為可有效的隔絕，但對於新型態的攻擊手法，因為事先並無相對應的 pattern 及 signature，便無法在第一時間進行偵測；另一方面，大多數的 IDS 系統皆是被動的偵測，而無法預先得知可能的攻擊行為而進行主動預警。因此，本計畫將利用主機連線行為的規律性(regularity)，來進行偵測，當主機之行為異於平常之規律性，便有可能是遭受攻擊或行為異常。

為網路上一伺服器之連線數量統計圖，由圖中可以看出主機在連線上具有一定的規律性。而對於預警方面，我們則將分別利用失誤樹(fault tree)及分群法(clustering)來計算主機或群組的風險程度，對可能遭受攻擊的主機進行預警。

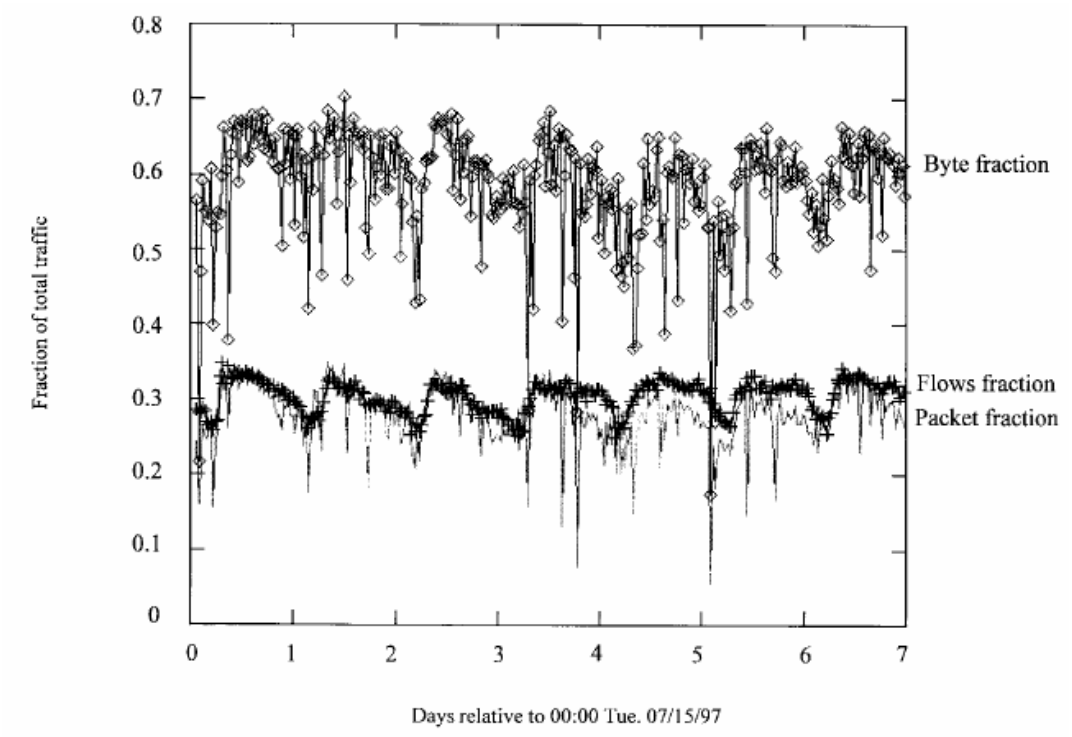


圖 48、主機連線數量統計圖

本年度計畫開發網路威脅風險分析與威脅預警之核心技術與軟體單元，將以實作去年度之計畫為主，設計一套以履歷為基礎之輔助系統，藉由防火牆的連線紀錄，可以不需要事先得知攻擊 pattern 及 signature 便能偵測出異常行為，同時藉由風險分析，我們可以在主機遭受攻擊或入侵前，事先加以預警。同時，我們也將加強法則編輯，讓使用者可以自行定義法則，透過人性化的使用者介面，輸入自定規則，並且在法則瀏覽器中顯示法則及異常連線記錄。

第二節、系統架構

本系統之目標為製作一套輔助管理者決策暨制定資安規則之系統，並且利用視覺化的顯示介面來達成網路威脅分析及風險量化，將防火牆日誌分解成包括 Source、Destination、Port、Time 等欄位，系統根據給定之規則防護並監測主機的安全。使用者可利用查詢工具和網路連結履歷(Profile)將詳細連線資料取出分析，配合時序資料之輔助來找出難以發現的攻擊，判斷是否有異常行為發生。

圖 49 為本系統的整體架構圖，其中包含：資料來源擷取系統、資安法則定義與編輯系統、資料庫與資料探勘系統以及資安法則驗証系統。以下將分項逐一說明介紹。

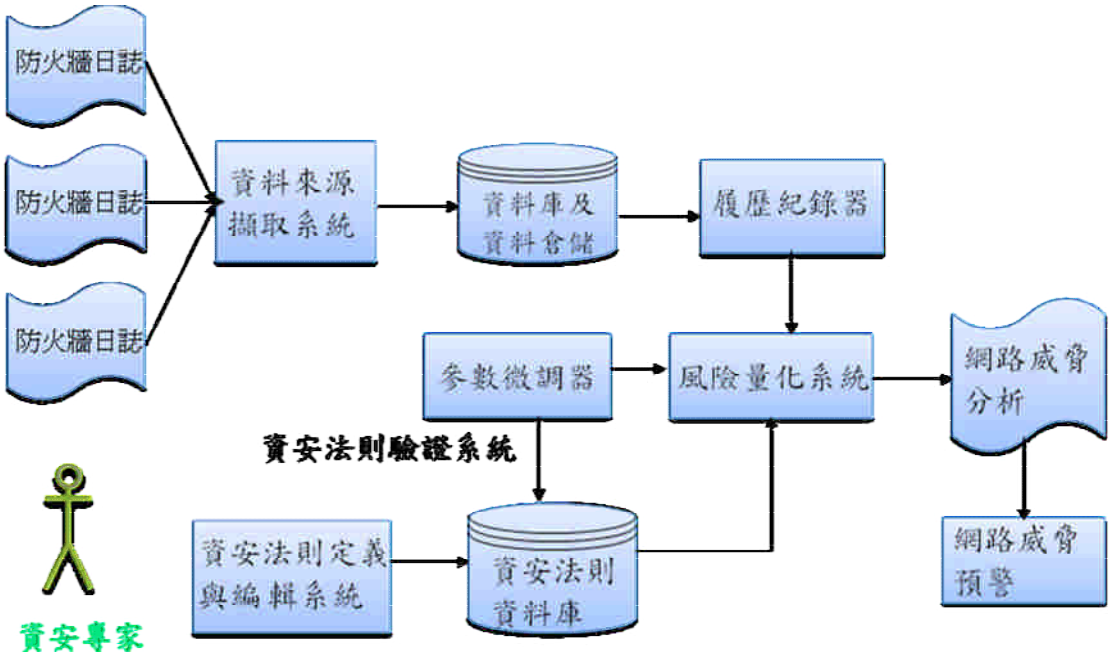


圖 49、系統架構圖

2.1 資料來源擷取系統

為了減少在傳輸防火牆日誌時所需花費的成本(cost)，包含網路頻寬、傳送時間及磁碟空間，我們利用 2-tier 的方式來進行達到較高的效能(performance) 圖 50 為此架構之示意圖，各營區(battalion)有自己的 SIM 伺服器，收集各自營區的防火牆日誌，再將分析後的異常資料，包含主機及連線清單，上傳至 master SIM 伺服器，並且我們在上傳的過程中加入壓縮及加密的機制，一方面可以進一步減少資料量，另一方面可以防止資料在傳送過程中被惡意的攻擊者竄改。

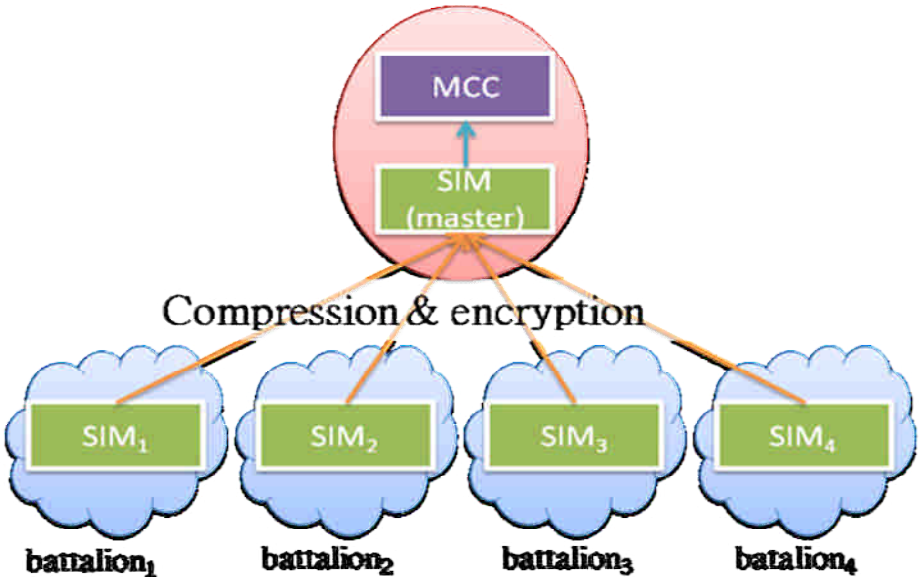


圖 50、2-tier 之系統架構

資料擷取後將存入 Log DB，並且進一步利用資料倉儲(data warehouse)將連線依各種不同維度(如：時間、IP、Port)進行整合(aggregation)，以利未來查詢時能更快速的回應。

2.2 資安法則定義與編輯器

資安法則定義及編輯器提供一圖形化的介面讓使用者可以輸入及設定自行定義之資安規則。我們將規則拆解成各種子規則的組合，如表 21 為一利用兩項子法則(S1 及 S2)組合而成的法則(R1)，S1 及 S2 利用 Union 的方式組合。而相對的，

Sub Rule	Source ip	Source port	Destination ip	max	min	Un safe port	Src_ip _gid	dst_ip _gid
s1	Null	Null	Null	100	2	Null	1	Null
s2	Null	Null	Null	1000	1	Null	null	5
s3	Null	80	Null	null	null	null	null	Null
s4	Null	Null	Null	Null	Null	100	null	null

表格 21、資安法則

為多層次的子法則組合，其中，S2 和 S3 先以 Sub 的方式組合成 R4，接著 R4 再以 Union 的方式和 S1 組合成 R3，最後 R3 再和 R3 以 intersect 的方式組合成 R2，而 R2 即為我們最終的法則。利用這樣的拆解及組合的方式，我們可以讓使用者自行設計規則，而非只受限於已事先定義的法則項目。

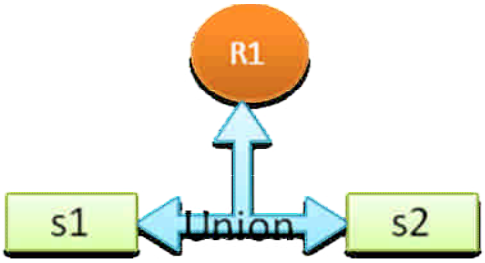


圖 51、子法則拆解範例

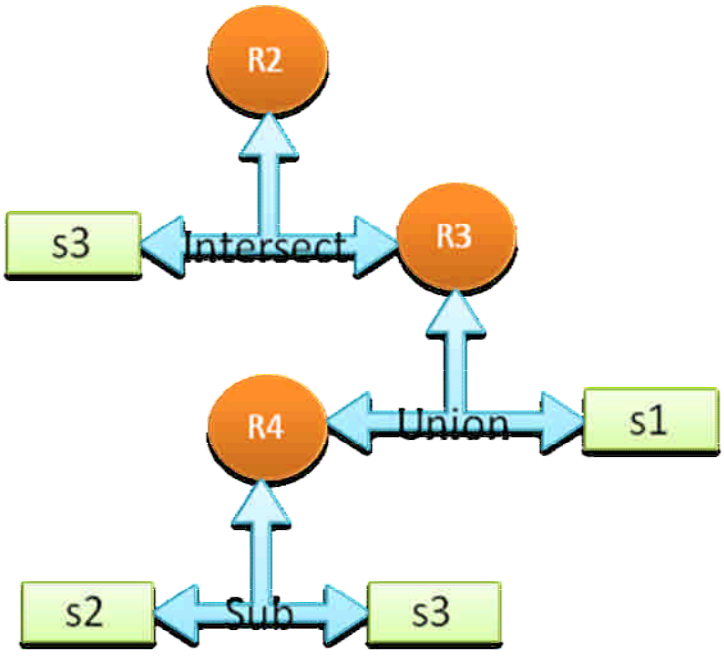


圖 52、多重子法則拆解範例

在實作方面，為了達成以上的目標，資料庫的表格設計也將配合子法則拆解的需求，如表格 22、法則表格

及表 22 分別為法則表格及子法則表格。其中每個 internal node 皆存在法則表格，而 leaf node 則存在子法則表格。再透過法則表格中的 Operator 欄位來記錄組合的方式。

Rule	Root	Left	Right	Operator	Comment
R1	Yes	S1	Null	Null	列管主機規則
R2	Yes	S2	Null	Null	提供軟體更新的伺服器規則
R3	Yes	S3	null	null	申請使用特定 port
R4	Yes	S4	null	null	使用不安全組合服務之主機

表格 22、法則表格

雖然利用子法則拆解方式可以有效率的讓使用者自定法則，但由於一條規則被拆成好幾個部份，因此，在每次讀取規則時，皆需要進行一次組合，traverse 整個 rule tree 才能將整個規則組合起來，對於較長或結構較複雜的法則來說，需要花費的時間也會較多。因此，我們利用 XML 文件同屬樹狀結構的特性，將每一條規則以 XML 的方式來描述，之後在每次讀取法則時，只需要讀取 XML 文件，而不需要組合整個 rule tree。表 23 為一以 XML 文件描述更新頻率異常法則的範例。

Sub Rule	Source ip	Source port	Destination ip	max	min	Un safe port	Src_ip_gid	dst_ip_gid
s1	Null	Null	Null	100	2	Null	1	Null
s2	Null	Null	Null	1000	1	Null	null	5
s3	Null	80	Null	null	null	null	null	Null
s4	Null	Null	Null	Null	Null	100	null	null

表格 23、子法則表格

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<rule_table rule="R1">
        <subrule_table subrule="s1">
                <max>100</max>
                <min>100</min>
                <source_ip_gid>
                        <group_id gid="1">
                                <ip>140.113.166.21</ip>
                                <ip>140.113.166.22</ip>
                                <ip>140.113.166.23</ip>
                                <ip>140.113.166.25</ip>
                        </group_id >
                <source_ip_gid>
        </subrule_table >
</rule_table>
```

圖 53、更新頻率異常法則之 XML 文件

2.3 資料倉儲與履歷系統

為了替每一台主機建立個別的網路連結履歷(profile)，並利用這份履歷來偵測該台主機是否出現異常的行為並提出警告，我們將利用資料倉儲來計算各維度及不同解析程度資料，使未來在資料的存取上能更加快速。而使用者便能由 profile 來判定主機異常的可能性，在後續小節中會說明網路連結履歷(profile)的設計與設計流程並且介紹如何利用資料探勘中分群(clustering)技術來為網路中的主機進行分群，以輔助資安專家在系統參數上的設定。

1. 資料倉儲

由於 Log DB 僅存放各別的連線記錄，當我們要查詢某一主機在某段時間內的連線總數時，資料庫需要針對不同的需求來進行查詢，對於程式的效率性較差，因此，我們利用資料倉儲來預先將可能會用到的數值計算出來，我們將計算各別維度，如:IP、Port 及時間在不同解析程度，如：日、週、季、月、年，的連線總數，將來需要用到這些資料時，便能即時取用，而不需要再額外進行計算。

2. 網路連結履歷(Profile)

我們所建立的履歷為一階層式的結構，初始的網路連接履歷為空，接著從防火牆日誌(firewall log)中取得每個主機的 IP 位置和網際網路服務後，會先建立出

如圖 54 之樣示，並在 others 紀錄此 IP 所有連線數目和連線型態(如 Http、Ftp)。

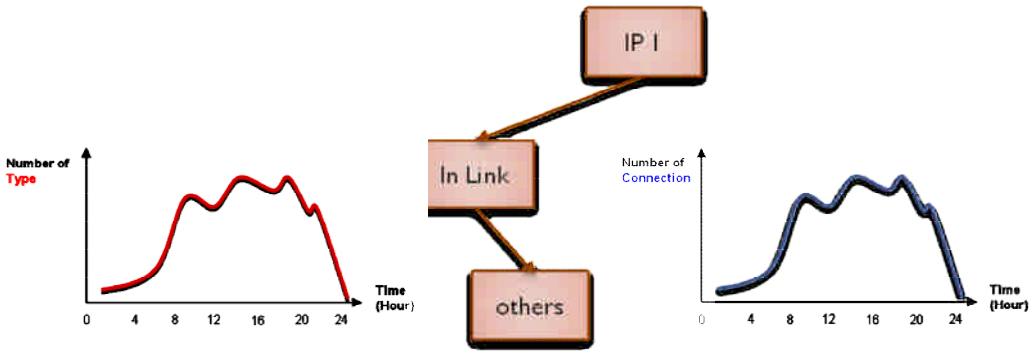


圖 54、網路連結履歷初始圖

然而當某項服務的連線數值(count)大於使用者所定義的門檻時(Count Threshold)，會在網路連結履歷(profile)新增一項個別的節點(node)如圖 55 所示，將 FTP 服務從 others 分離出來。經過一段使用者所定義的時間(Time window size)後，整體的網路連結履歷(profile)會變成如

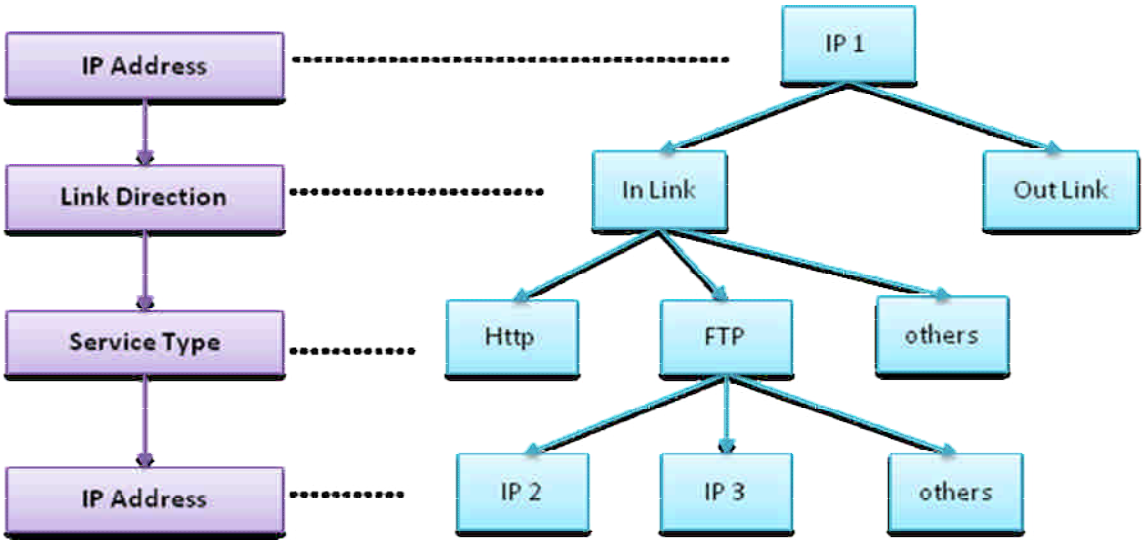


圖 55、連結履歷連結

其中因為排版空間的限制，我們只顯示 In link 的分支，另外在 Out link 的分支其結構和 In link 是相同的，而每一層的定義將一一說明如下：

第一層：針對該主機所有的連結建立履歷。

- 格式: (IP, Pattern)
- 範例: (140.113.6.2, Pattern)

第二層：針對該主機的連出連結及連入連結分別建立履歷。

- 格式: (IP, Direction, Pattern)
- 範例: (140.113.6.2, IN, Pattern)

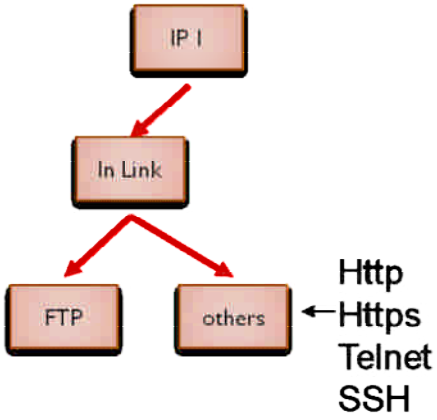


圖 56、新增個別服務項目

第三層：除了連結方向外，還會根據服務類別建立各自的履歷。

- 格式: (IP, Direction, Service, Pattern)
- 範例: (140.113.6.2, IN, http, Pattern)

第四層：除了連結方向及服務類別外，則會根據對方的 IP 位址建立個別履歷。

- 格式: (IP, Direction, Service, Domain, IP, Pattern)
- 範例: (140.113.6.2, IN, http, 140.112.*.*, 140.112.172.46, Pattern)

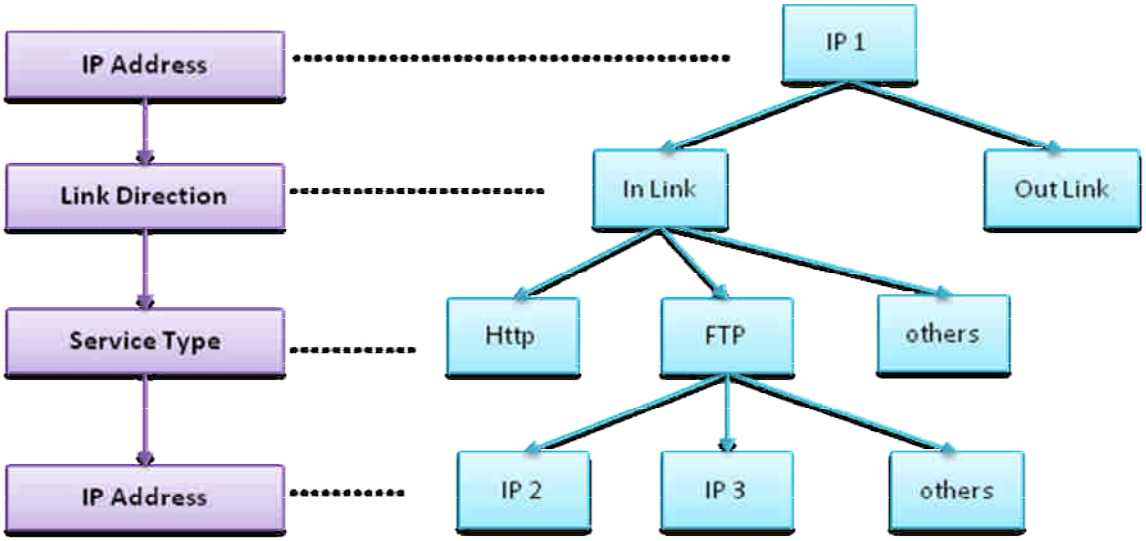


圖 57、主機履歷格式

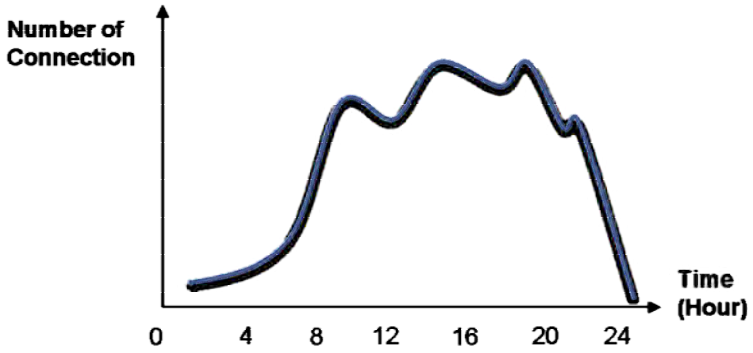


圖 58、時間序列樣式

在每一層的履歷中，我們都會利用資料倉儲的技術來建立時間序列式的樣式(time series pattern)，如圖 58。

2.4 風險分析及預警

在風險分析方面，我們將針對各別主機利用失誤樹分析(Fault tree analysis)來進行風險量化分析，而另一方面，我們也將利用資料探勘中的分群法(clustering)來將連線行為相似的主機進行分群，再依各群體中主機異常數量，來判斷該群組的風險程度。

首先是失誤樹的部份，同樣的我們將每天利用失誤樹計算一次風險值，並將結果上傳至 master SIM，以供未來 MCC 查詢，因此，每台主機每天會有不同的失誤樹的風險值。由於對於使用者新增的規則，其相對應的失誤樹並無法完全由資料倉儲中取得所需的資料，亦即，無法完全得知每個節點的機率值，因為，我們僅針對五項已知的資安規則進行計算，將這五棵失誤樹內進在程式中。

在群組風險上，我們將取各主機當日之履歷，做為資料物件(data object)，進行分群。主要可以分為兩部份，其一為特徵選擇(feature section)，其二為相似度比對(similarity measurement)。在特徵選擇方面，初步我們將利用後序排列(post order)的方式將樹狀結構的 profile 轉成序列(sequence)，而此 post order sequence 即為該 profile 的特徵；而在相似度比對方面，我們將利用 editor distance 的方式來計算兩個 sequence 的相似度，editor distance 為一利用 dynamic programming 的概念設計的 pseudo-polynomial algorithm，在效率上有較好的表現。

當為各主機進行分群後，如圖 59 所示，當群組內的主機發生異常，我們便能計算其異常的比例，做為群組的風險程度，當愈多的主機發生異常，其風險程

度也愈高。

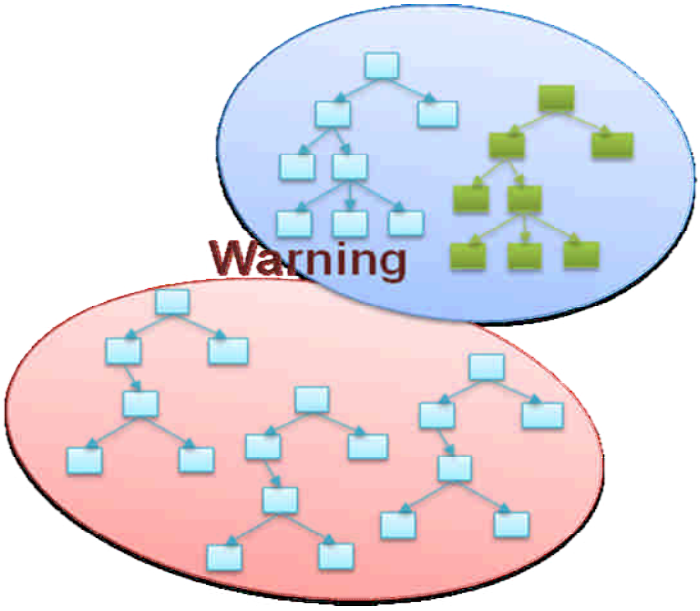


圖 59、主機分群圖

第三節、軟體設計說明

3.1 系統綜觀

本系統可以分為四大部份：履歷(profile)、法則編輯、資料倉儲及風險分析預警，如圖 60 所示。履歷部份用來為各主機建立其歷史連線紀錄並且比對已知的資安法則找出可能為異常的主機，履歷部份又可分為：profile builder、anomaly detector；資料倉儲將收集到的連線資料依各種維度進行整合，以利將來存取時能即時的拿到資料；法則編輯主要提供使用者介面來讓使用者輸入資安法則，並且將資安規則拆解成子規則(sub rule)存入資料庫，可分為：rule editor、rule browser 及 rule database；風險分析預警為主機計算其分險程度，進而做為預警之依據，其中又可分為：fault tree analysis 及 profile clustering。

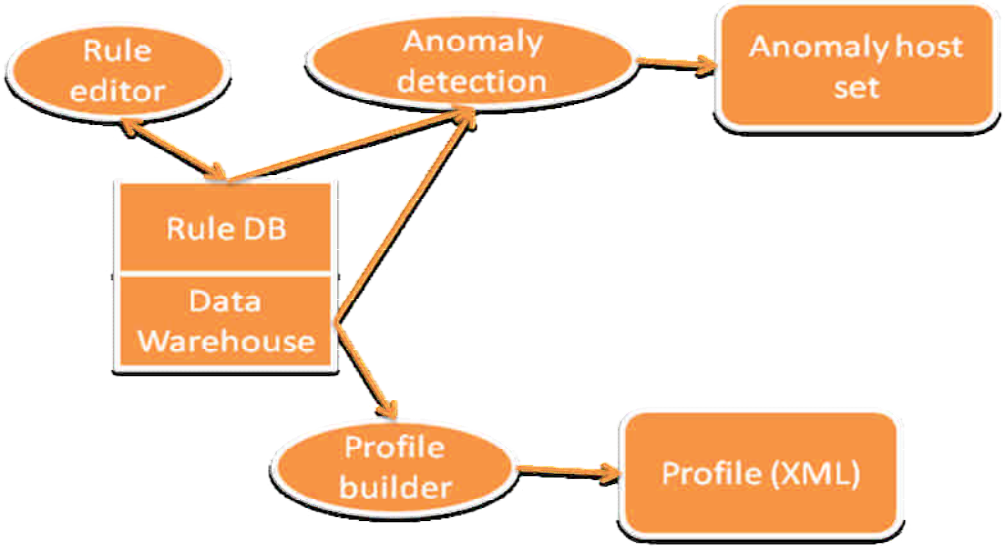


圖 60、系統架構

3.2 電腦軟體構型項目層面設計決策

1. 電腦軟體發展環境選用決策分析

本系統基於可移植性(portable)使用 Java 開發程式，作業系統則為 Debian GNU/Linux，在資料庫方面使用自由軟體(open source)的 MySQL Database，firewall 則參考[35]技術報告中推薦的項目，使用 Netscreen-5GT。

2. 電腦軟體構型項目行為設計決策分析

名稱	所屬單元	功能
Log parser	資料來源擷取系統	將 Log 存入資料庫
Profile builder	profile	利用 data warehouse 預算的資

		料建立 profile
Profile clustering	風險分析	依據各主機之 profile 進行分群
Anomaly detection	profile	比對 profile 和 rule，找出異常主機
Rule editor	法則編輯	提供使用者輸入自訂法則
Rule browser	法則編輯	提供使用者瀏覽已定義之法則
Create data warehouse table	資料倉儲	依據資料庫中的 log 建立未來將使用到的資料倉儲表格
Data warehouse	資料倉儲	將資料庫中的 Log 依據不同維度進行整合
Fault tree analysis	風險分析	計算各主機之風險程度

表格 24、電腦軟體構型項目行為設計決策分析

3.3 電腦軟體構型項目架構設計

1. 電腦軟體構型項目組件

Log DB

SNo	序號
Datatime	日期時間
Source IP	來源位址
Source Port	來源埠號
Destination IP	目的位址
Destination Port	目的埠號

表格 25、Log DB

Host group

id	序號
IP	主機位址
gid	群組編號

表格 26、Host group

Group

id	序號

Name	群組名稱

表格 27、Group

Combination Port

Id	序號
cid	組合編號
Port	埠號

表格 28、Combination Port

Available Port

Id	序號
Port	埠號
Gid	群組編號

表格 29、Available Port

Port group

Id	序號
Pgid	群組編號
Port	埠號

表格 30、Port group

Rule

Rule	法則編號
Root	是否為 Root
Left	左子法則編號
Right	右子法則編號
Operation	運算方式
Description	法則描述

表格 31、Rule

Sub-rule

Sub-Rule	子法則編號
Source ip	來源位址
Source Port	來源埠號
Destination IP	目的位址
Destination Port	目的埠號
Datetime	日期時間

max	最大值
min	最小值
Unsafe Port	不安全的埠號組合
Source ip gid	來源群組
Source Port gid	來源埠群組
Destination IP gid	目的群組
Destination Port gid	目的埠群組

表格 32、Sub-rule

2. 執行的概念

名稱	功能
Log parser	每天將 Log 存入 Log DB
Profile builder	將每個小時的 log 轉成 snapshot，並且將至多一個月的 snapshot 建立為 profile
Profile clustering	將 tree 轉為 feature vector，進行分群
Anomaly detection	每天比對 profile 和所有的 rules，將異常主機 list 傳給 master SIM
Rule editor	GUI 介面提供給使用者編輯、修改、新增資安規則
Rule browser	提供使用者瀏覽已定義之法則
Create data warehouse table	依據資料庫中的log建立未來將使用到的資料倉儲表格
Data warehouse	將資料庫中的 Log 依據不同維度進行整合
Fault tree analysis	計算各主機之風險程度

表格 33、執行概念表

3. 介面設計

名稱	介面及參數
Log parser	無可用參數
Profile builder	無可用參數
Profile clustering	無可用參數

Anomaly detection	無可用參數
Rule editor	GUI 介面提供給使用者編輯、修改、新增資安規則
Rule browser	提供使用者瀏覽已定義之法則
Create data warehouse table	無可用參數
Data warehouse	無可用參數
Fault tree analysis	無可用參數

表格 34、介面設計

3.4 電腦軟體構型項目細部設計

1. 物件導向分析(OOA)

a. Use Case

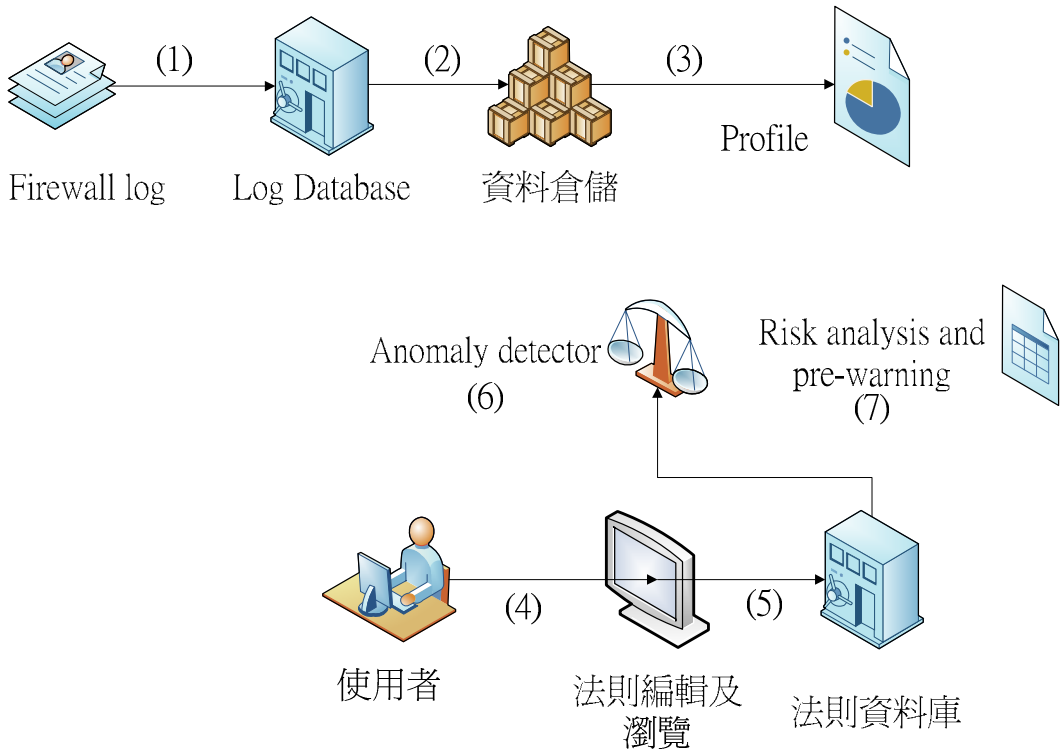


圖 61、物件導向分析

Log parser 依照欄位每天將 Firewall log 讀入 Log database

(1) 資料倉儲將一天的 Log 由 Log database 中讀出來，根據不同的維度建立不同時間長度的 data warehouse 表格，並且將資料加總後存入 data warehouse。

(2) Profile builder 利用資料倉儲的資料建立 profile。

(3) 使用者登入 MCC 利用法則編輯器新增、修改或刪除法則。

(4) 將法則拆解成子法則存入法則資料庫

(5) Anomaly detector 將 profile 及 rule 進行比對，將異常的主機存成 list 並且上傳給 Master SIM。

(6) Risk analysis 每天計算所有主機的風險值並且建立各主機之群組，計算其風險值。

2. 法則編輯子系統使用案例

a. 內容描述

使用者經由滑鼠及鍵盤設定自定之規則，並透過 rule browser 選擇檢視違反該規則之主機連線狀態

b. 功能與性能需求流程(Flow of Events)

使用者由 rule browser 選定規則後，可得到 rule id，透過傳送 rule id 給 master SIM 可以得知使用者欲檢視之規則編號(rule id)。

c. 活動圖(Activity Diagram)

圖 62、活動圖

d. 循序圖(Sequence Diagram)

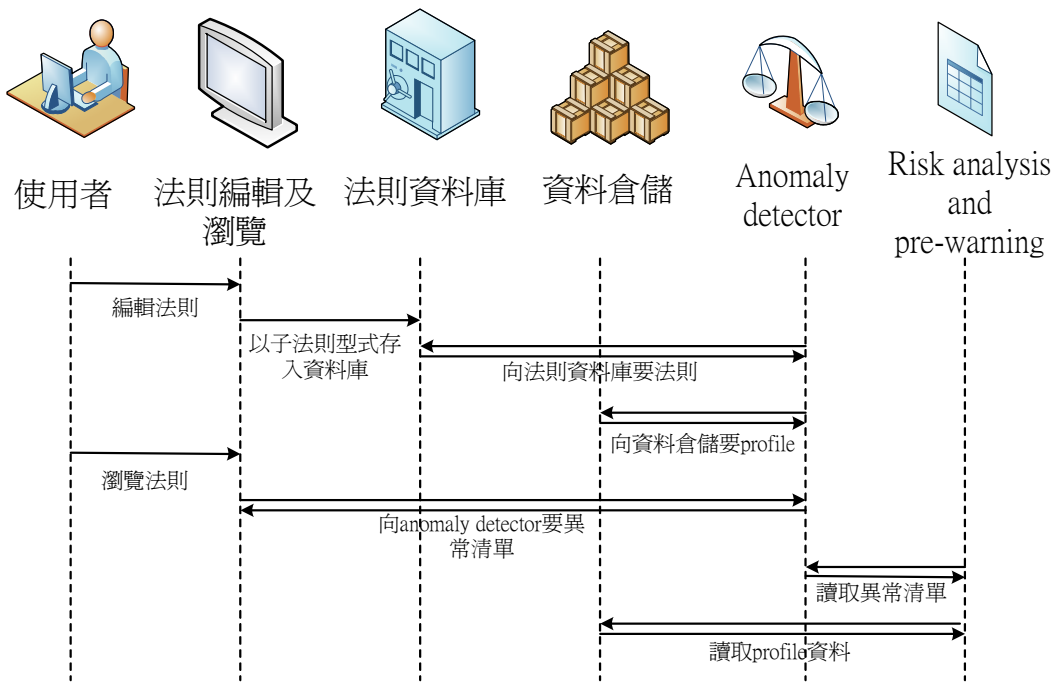


圖 63、循序圖

e. 合作圖(Collaboration Diagram)

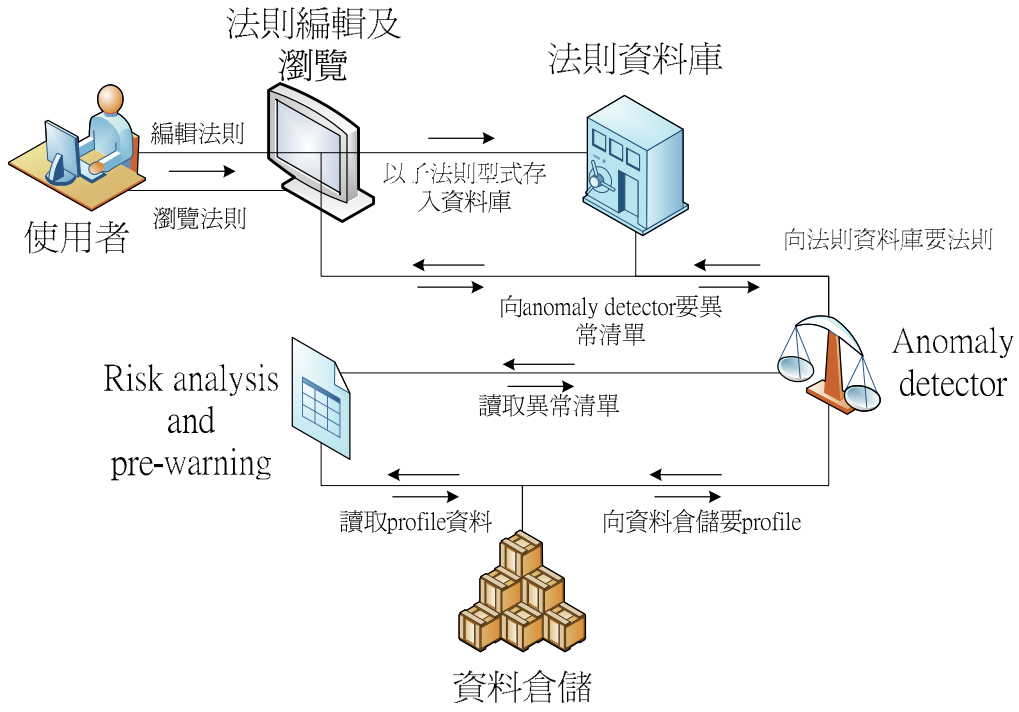


圖 64、合作圖

3.5 註記

中英文名辭對照與縮寫

Profile 履歷

Anomaly detection 異常偵測

Clustering 分群

Fault tree 失誤樹

第四節、網路威脅分析及預警技術之研究

在資安法則上，我們將針對去年的五項資安法則進行格式轉換，將舊的法

則格式轉換為新的 XML 型式，同時我們也將針對三種攻擊行為，分別為：port scan、warm 及 trojan，進行偵測，以下分別介紹這幾種異常及攻擊行為：

4.1 預先定義之資安規定

1. 未更新之主機(No update PCs)

為避免已知的漏洞被有心人士利用，每部電腦必需至 update server 下載並安裝 patch，若發現有某些電腦長時間未連線至 update server，則視為違反資安規則，亦即這些電腦極有可能存在資安漏洞。

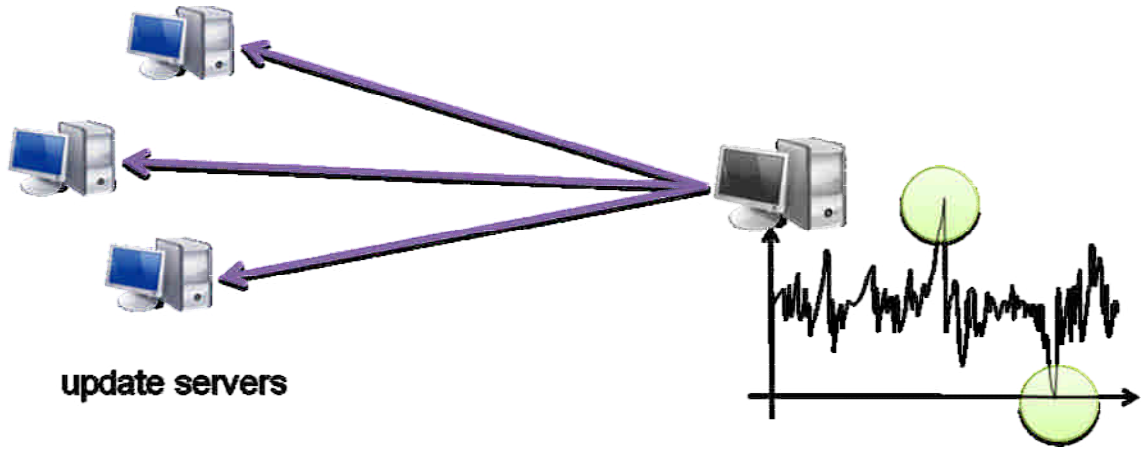


圖 65、違反未更新之主機法則

2. 未回報之主機(No report PCs)

為避免稽核程式(agent)回報資訊，有些電腦或營區會安裝防火牆(firewall)阻檔回報的封包。因此，若發現有某些電腦或營區長時間未進行回報，則視為規避稽核，可能內部正進行違法的行為。

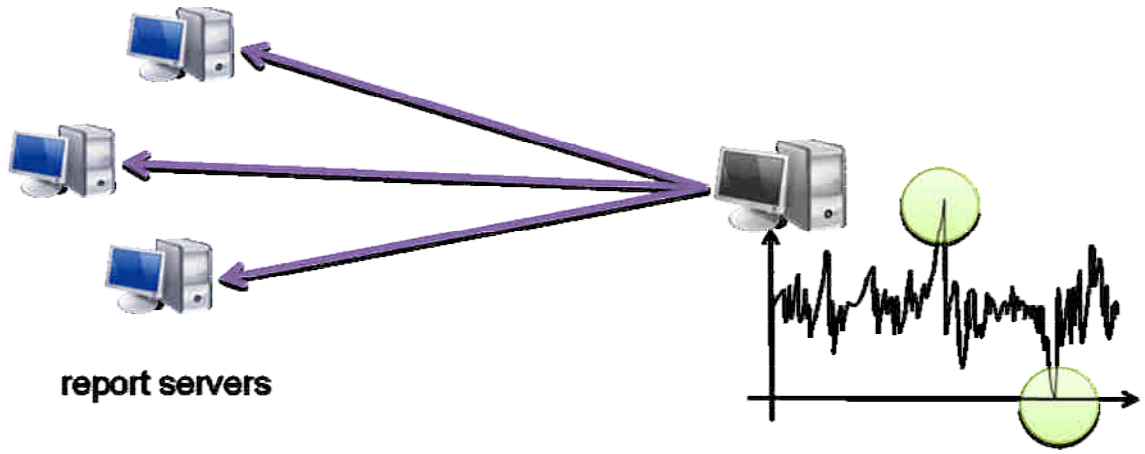


圖 66、違反未回報主機法則

3. 具有不安全服務(unsafe service)組合之主機

由資訊安全專家定義，哪些服務(service)不能同時存在一台伺服器(server)上。

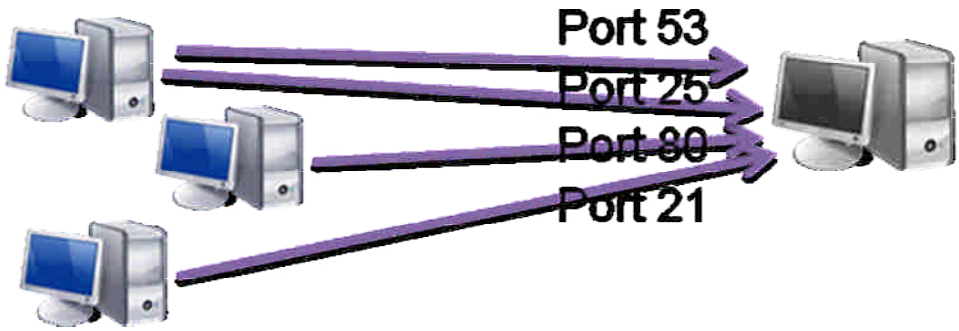


圖 67、違反具有不安全服務組合法則

4. 未列管伺服器

營區內的所有伺服器都必需登記列管，不能私自架設伺服器。

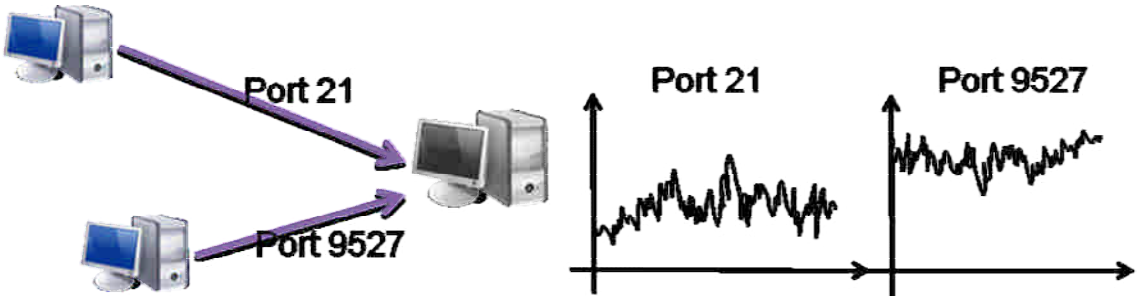


圖 68、違反未列管伺服器法則

5. 使用未申請的 Port

當使用者連往遠方不知名的 Port 時，有可能是遭到攻擊而將資料外洩。

圖 69、違反使用未申請的 Port 法則

4.2 惡意攻擊行為

1. 掃描通訊埠攻擊 (Port Scan)

掃描通訊埠攻擊透過對目標主機傳送封包藉以探查目標主機可能存在的漏洞。這種攻擊通常是入侵的第一步，如果偵測到掃描攻擊將對維護資訊安全將有很大的助益。

遭受掃描攻擊時，主機會收到針對各種不同服務類型的連線，這些連線所要求的服務與履歷中所記錄的服務種類往往會有很大差異，如圖 70 所示。我們的系統藉由比對履歷記載的服務與被要求的服務，便可發現掃描通訊埠攻擊。另外來自掃埠工具的攻擊會產生大量的連線，當超過 InLink 的安全限度時我們就將他視為遭受攻擊。

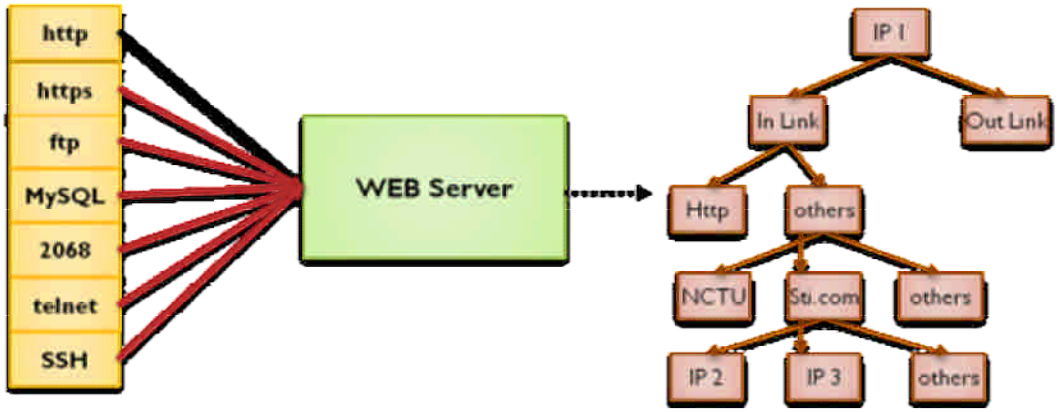


圖 70、掃描通訊埠攻擊

來自駭客的掃埠攻擊相較之下複雜許多。他們會有計劃地針對某些重點埠進行掃描並分散掃描時間降低被發現的可能，此種攻擊單是觀察連線數目難以發現，必須利用連線所要求的埠種類來輔助偵測。若是針對單一主機多個埠的攻擊，可以利用 InLink 所記錄的連線種類來判斷，當掃埠攻擊發生時，觀察時間內所累積的連線種類會比平常多，且不屬於此主機服務的要求也會上升。若是針對多主機單一埠的掃描，我們可觀察全域的被要求服務統計，若是某一個服務或是某一個埠被大量的要求就有可能是此類攻擊。

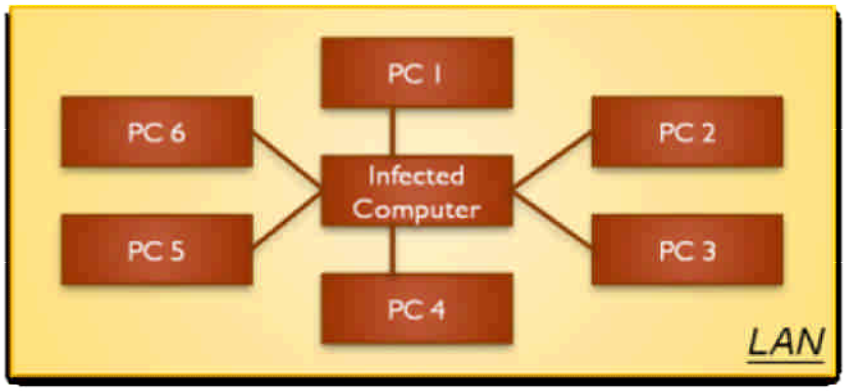


圖 71、透過網路上的芳鄰傳染的蠕蟲

2. 電腦蠕蟲(computer worm)

電腦「蠕蟲」跟病毒一樣，可將自身從一台電腦複製到另一台，但蠕蟲會藉由控制電腦上可以傳送檔案或資訊的功能自動複製。一旦您的系統中有蠕蟲存在，它就會自動蔓延。在蠕蟲蔓延的過程當中，它會嘗試進行大量的連結，圖 71 所示為一利用網路上的芳鄰進行傳染的蠕蟲，它會自動偵測區域網路上的其他主機，並嘗試透過網路分享的資料夾進行傳染。感染蠕蟲的主機之連結數量會遠大於履歷中所記錄的行為，故我們的系統可以發現感染蠕蟲的主機，並將其歸類為異常行為。

3. 木馬程式(Trojan horse)

一個完整的特洛伊木馬套裝程式含了兩部分：服務端（伺服器部分）和用戶端（控制器部分）。植入對方電腦的是服務端，而駭客利用用戶端進入運行了服務端的電腦。運行了木馬程式的服務端以後，會產生一個有著容易迷惑用戶的名稱的進程，暗中打開埠，向指定地點發送資料，透過木馬程式，惡意攻擊者可以掌握受害者的電腦。

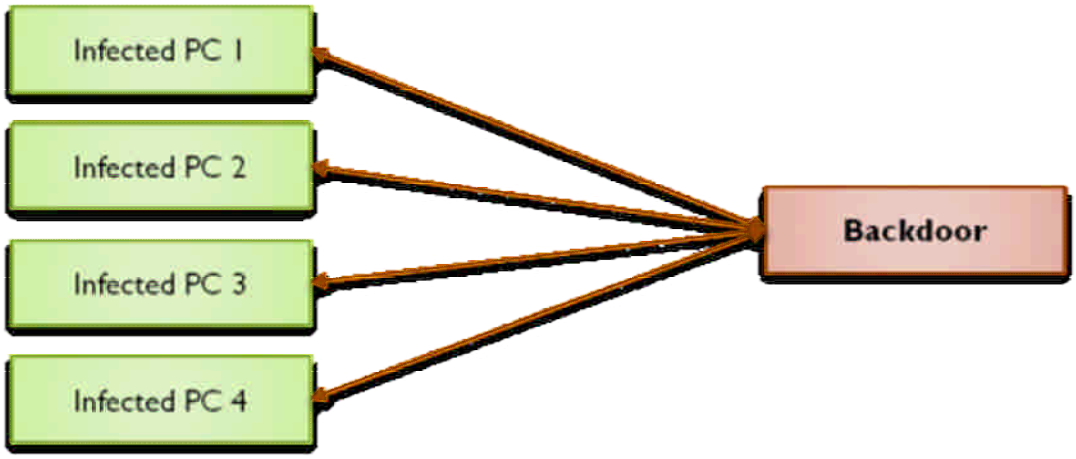


圖 72、同時遭受木馬攻擊的電腦

如果多台電腦皆感染同一隻木馬程式，則當惡意攻擊者要操控這些電腦時，會有許多主機同時收到不在履歷中的連線，如圖 72 所示，因為這與履歷中所記錄的有很大差異，因此我們的系統可以發現並將其歸類為異常行為。

第五節、失誤樹分析

5.1 簡介

失誤樹(fault tree)是 1962 年由貝爾實驗室(Bell Telephone Laboratories)發表，從系統的失效現象做出發，根據這些失效現象，分析失效發生的原因及造成系統失效的可能部位。而為了使失誤樹能順利運作，我們需要輸入精確的錯誤資料和適當的模型(model)，才能計算出失誤發生的風險。圖 73 為一失誤樹的範例，用來表示 Top event 發生的風險，其中每一個圓形的節點都代表一個事件，而失誤樹則透過一些邏輯的組合來描述這些事件對 Top event 的影響。例如圖 73 的失誤樹告訴我們：「當 X2 和 X3 同時發生或 X1 發生時，top event 變會發生」。

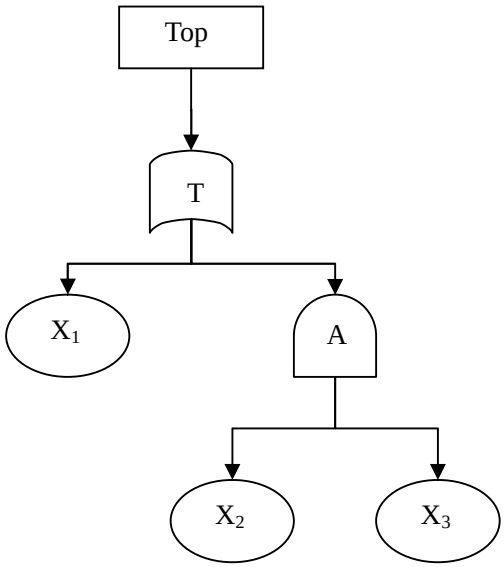


圖 73、Example of fault tree

我們也可以透過下面的式子來計算 top event 的風險值：

$T = X_1 \cup A$

$A = X_2 \cap X_3$

因此，我們可以得到 $T = X_1 \cup (X_2 \cap X_3)$，若以 P_{X_i} 來代表 X_i 發生的機率，則 $P_T = 1 - ((1 - P_{X_1})(1 - P_{X_2} P_{X_3})$，此即為 top event 可能發生的風險。

5.2 應用 Fault Tree Analysis 進行違反資安規則之風險量化分析

我們利用失誤樹分析法(fault tree analysis)來為特殊的環境設定進行風險量化，並且應用在這次的計畫當中。

1. 更新頻率異常

更新頻率異常的情形可以分為頻率太高或低，其中頻率的高低是和本身平常的連線數比較，當高過平常的連線數或低於平常的連線數太多則視為異常。其中左子樹為更新頻率過高的情形，為考慮整個網域總連線數的影響，其機率為發生更新連線的機率乘上發生一般連線的機率。圖 74 為計算更新頻率異常之風險的失誤樹。

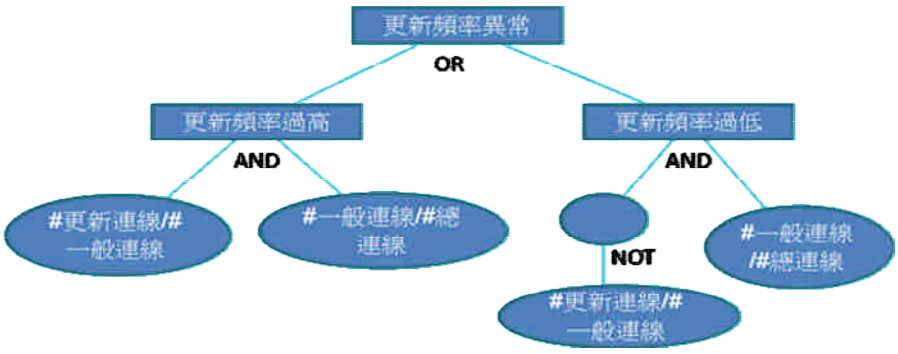


圖 74、更新頻率異常之失誤樹

2. 回報頻率異常

和更新頻率異常類似，回報頻率異常的評估方式以回報的連線數和該主機本身的一般連線數比較，若太高或太低都將會被視為異常，詳細的失誤樹顯示在圖 75 中。

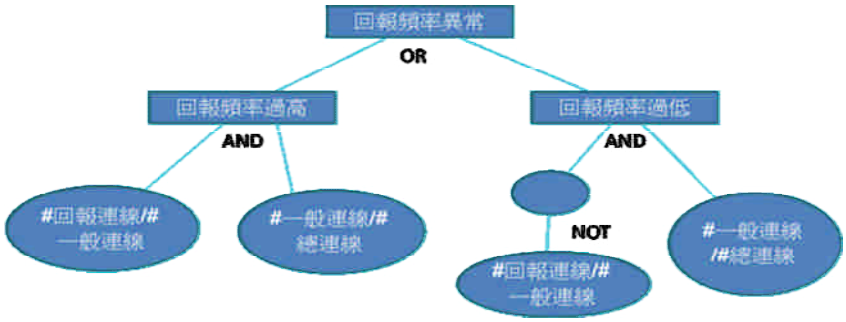


圖 75、回報頻率異常之失誤樹

3. 不安全的服務組合

由於不安全的服務組合需要由資安專家來定制，因此我們在這裡只列出兩種可能的不安全服務組合，分別為 DNS+任何服務及 HTTP 服務+FTP 服務，而當主機存在的不安全服務組合愈多，其存在的風險就愈高，圖 76 為使用不安全服務組合之失誤樹。其中，左子樹為 DNS 和其他服務共存的機率，計算方式為出現 DNS 連線的機率乘上其他服務連線的機率，當兩者機率愈高時，該項異常發生的機率也愈高。

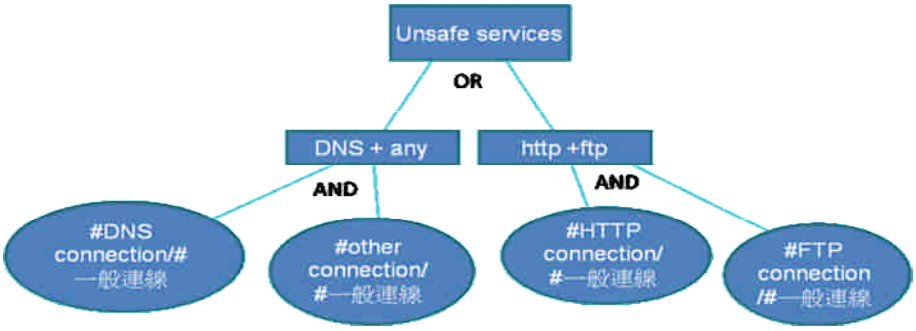


圖 76、不安全的服務組合之失誤樹

4. 使用未申請的服務

為了找出主機是否開設了某種不在申請內的服務，我們會檢查該主機各個 port 的 in-link 是否超出所有 port 的平均 in-link 太多，以及該 port 是否為 well-known port，若該 port 為 well-known port 則背後存在服務的風險則較高，圖 77 為使用未申請服務之失誤樹。其中右子樹的計算方式為：沒有在伺服器清單的連線數/連進所有 well-known port 的連線數，即為發生連進不在伺服器清單中的非 well-know port 連線的機率。

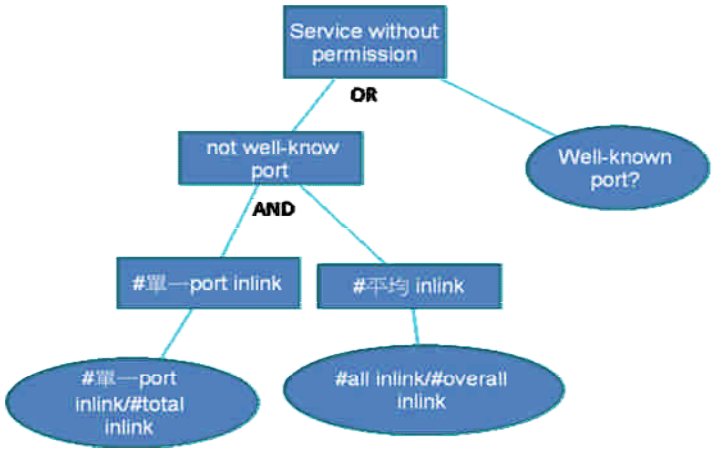


圖 77、使用未申請服務之失誤樹

5. 使用未申請的 port

使用未申請 port 的風險計算方式和使用未申請服務類似，但因為使用的 port 都必需經過申請，因此只要有不該開設的 port 出現，即會被視為異常，圖 78 為使用未申請 port 之失誤樹。和未申請服務不同的是，其右子樹的計算方式為：非 well-known port 的連線數/所有連線數，即為發生非 well-known port 連線的機率。

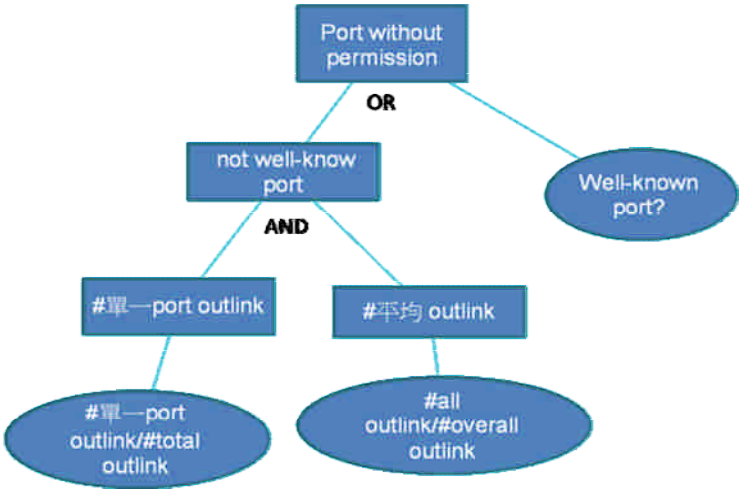


圖 78、使用未申請 port 之失誤樹

5.3 應用防火牆日至計算失誤樹風險

透過防火牆日誌(firewall log)我們可以得到主機的連線履歷(profile)，並且針對失誤樹的每一 leaf node 計算出其機率，例如：假設有 A、b 兩主機，其的 outlink 連線數分別為 100 及 200，其中 A 主機有 20 個 outlink 為連往 update server 的連線；而 B 主機則為 30 個，則主機 A 存在未更新異常的風險為：1-(1-(20/100)*(100/300))(1-(1-20/100)*(1-100/300))=0.564；而主機 B 存在未更新異常的風險則為：

1-(1-(30/200)*(200/300))(1-(1-(30/200))*(1-(200/300)))=0.355，因此，我們可以判斷 A 主機的風險較高。由於風險計算的結果和失誤樹的結構有密切的關系，我們必需考慮所有可能影響的因素，並且將這些因素正確的組織起來，如此才能避免計算出錯誤的風險。

5.4 系統實作說明

在實作上，可以分為三個部份，失誤樹編輯器(fault tree editor)、失誤樹分析器(fault tree analysis)及風險瀏覽器(risk browser)。

```
<?xml version="1.0" encoding="UTF-8"?>
<Fault-Tree>
<Intermediate-Event>
<Title>AbnoReport</Title><Info></Info>
<Or-Gate>
<Intermediate-Event>
<Title>High</Title><Info></Info>
<And-Gate>
<Basic-Event>
<Title></Title><Info></Info><Numerator>6</Numerator>
<Denominator>1</Denominator>
</Basic-Event>
<Basic-Event><Title></Title><Info></Info>
<Numerator>1</Numerator><Denominator>0</Denominator>
</Basic-Event>
</And-Gate>
</Intermediate-Event>
<Intermediate-Event>
<Title>Low</Title><Info></Info>
<And-Gate>
<Basic-Event>
<Title></Title><Info></Info>
<Numerator>19</Numerator><Denominator>1</Denominator>
</Basic-Event>
<Basic-Event>
<Title></Title><Info></Info><Numerator>1</Numerator>
<Denominator>0</Denominator>
</Basic-Event>
</And-Gate>
</Intermediate-Event>
</Or-Gate>
</Intermediate-Event>
</Fault-Tree>
```

圖 79、回報數異常之 XML

1. 失誤樹編輯器

圖 80 為一編輯回報數異常之畫面，此編輯器修改自 faultCat，我們提供十一種 events 做為將失誤樹的節點。並將編輯完成之失誤樹存成 XML 格式，如圖 80 為回報數異常之 XML 檔案(請參考圖 75 之失誤樹)，以做為分析時用。其中十一種 events 分別為：

a. 所有連線數

b. 更新連線數

c. HTTP 連線數

d. DNS 連線數

e. FTP 連線數

f. 回報連線數

g. Well-known port inlink

h. Well-known port outlink

i. 全域 well-known port inlink

j. 全域 well-known port outlink

k. 全域總連線數

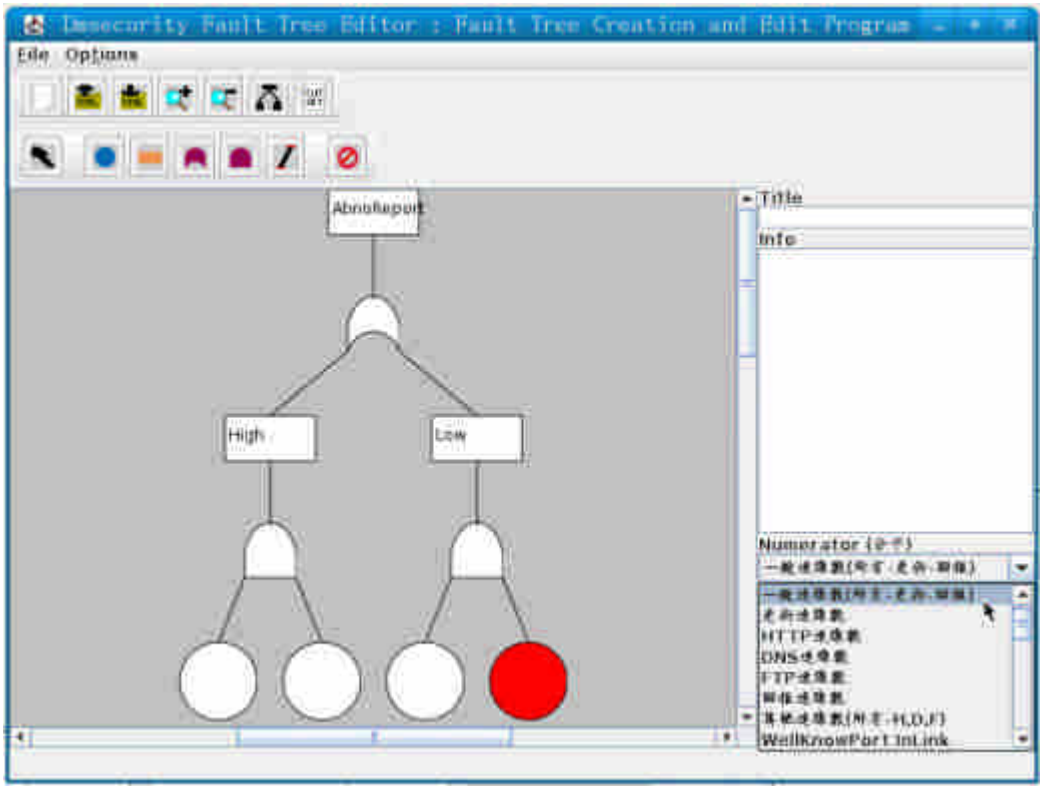


圖 80、回報數異常之編輯畫面

2. 失誤樹風險分析

讀取 XML 格式之失誤樹，並由資料倉儲中取得對應之資料進行分析，以下為十一種 events 的資料取得方式。

a. 所有連線數

select [Measures].[Count] on 0, [ips].Children on 1, [hostgroupD].[3] on 2 from [log]

b. 更新連線數

select [Measures].[Count] on 0, [ips].Children on 1, [hostgroupD].[7] on 2 from [log]

c. HTTP 連線數

select [Measures].[Count] on 0, [ips].Children on 1, [portD].[80] on 2 from [log]

d. DNS 連線數

select [Measures].[Count] on 0, [ips].Children on 1, [portD].[53] on 2 from [log]

e. FTP 連線數

select [Measures].[Count] on 0, [ips].Children on 1, [portD].[21] on 2 from [log]

f. 回報連線數

select [Measures].[Count] on 0, [ips].Children on 1, [hostgroupD].[7] on 2 from [log]

g. Well-known port inlink

with Set [port] As [wellknownportD].FirstChild : [wellknownportD].LastChild
Member [Measures].[AG] As Aggregate([port]) select {[Measures].[AG],
[Measures].[Count]} on 0 , [ipd].Children on 1 from [log]

h. Well-known port outlink

with Set [port] As [wellknownportS].FirstChild: [wellknownportS].LastChild
Member [Measures].[AG] As Aggregate([port]) select {[Measures].[AG],
[Measures].[Count]} on 0 , [ips].Children on 1 from [log]

i. 全域 well-known port inlink

with Set [port] As [wellknownportD].FirstChild : [wellknownportD].LastChild Member [Measures].[AG] As Aggregate([port]) select {[Measures].[AG], [Measures].[Count]} on 0 , [ipd].[All ipds] on 1 from [log]

j. 全域 well-known port outlink

with Set [port] As [wellknownportS].FirstChild : [wellknownportS].LastChild Member [Measures].[AG] As Aggregate([port]) select [Measures].[AG] on 0 , [ips].[All ipss] on 1 from [log]

k. 全域總連線數

select [Measures].[Count] on 0, [ips].Children on 1 from [log]

3. 風險瀏覽器

圖 81 為一風險瀏覽器，其中可以選擇日期、IP 位置、fault tree 種類及機率值進行篩選，瀏覽計算出之失誤樹風險值。

圖 81、風險瀏覽器

5.5 結論

失誤樹分析(Fault tree analysis)可以根據使用者不同的需求制定不同的計算模型，使分析更貼近實際狀況，但在模型的建立上則需要花費較多的工夫，必需仰賴資訊安全領域的專家來分析、拆解出各種交互影響的因素，並且為各種因素之間定義關聯，而模型的建立則大大的影響了分析的準確度。在我們的計畫中，因為資安規則為特殊的客制化規則；因此，我們必需以 fault tree 來實現風險量化分析，才能計算出各種違反規定的風險值，將異常狀況分級，並且利用此技術來對異常行為進行深入的風險量化評估及分析。

第六節、分群法與實作詳解

6.1 分群法簡介

資料分群（data clustering）或是分群演算法（clustering algorithms）是對於靜態數據分析的一門技術，在許多領域受到廣泛應用，包括機器學習（machine learning）、據據挖掘（data mining）、模式識別（pattern recognitition）、圖像分析（image analysis）以及生物訊息（bioinformatics）等。分群法是一種將資料分類成群的方法，其主要的目的乃在於找出資料中較相似的幾個群聚（clusters），在同一群聚的成員們擁有相似的屬性，或是更加相近的距離等。圖 82 即是將一平面座標點集分成四個群聚的結果。

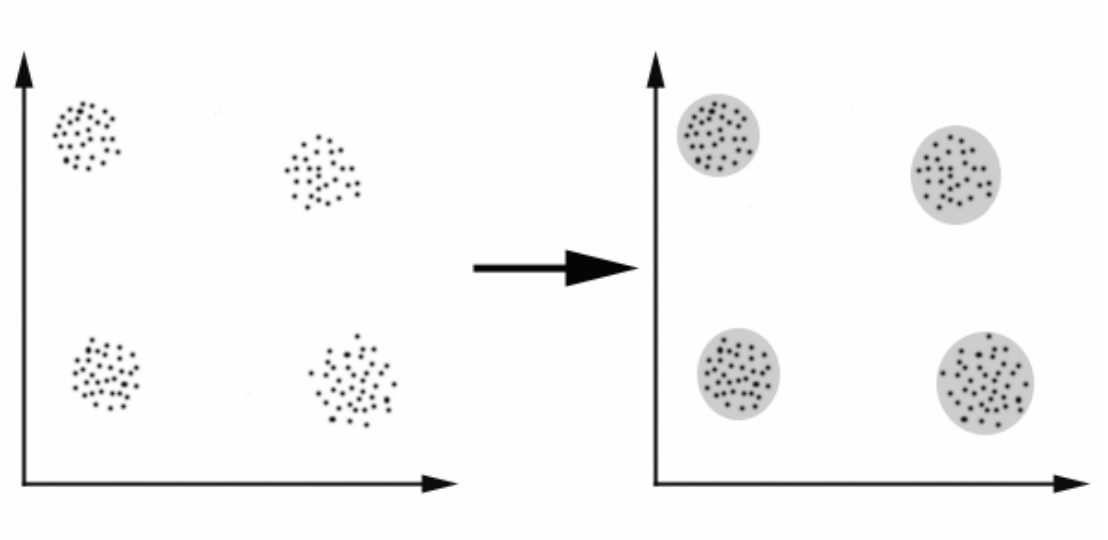


圖 82、Example of clustering

分群類型依照切入的角度不同，有很多不同的面相：

1. 階層式的與分割式的：

此兩種分群是依據群集的集合為巢狀的（nested）或非巢狀的（unnested）來區分，最大的不同在於分群過程中的群數（number of clusters）是否會改變。

階層式分群法（hierarchical clustering）的群數可以由大變小（top-down），或是由小變大（buttom-up），來促進群聚的合併或分裂，最後再選取最佳的群數。

分割式分群法（partitional clustering）則是先指定群數後，再用一套疊代的數學運算法，反覆將數據重新分配到最適當的群集裡以找出最佳的分群方式，通常會根據不同的群集計算其群中心（cluster center），再重新分配。

2. 互斥的（exclusive）與模糊的（fuzzy）：

當分派每個物件至群集時，以傳統集合概念來說，一個物件只屬於唯一一個集合，這種就是互斥分群（exclusive clustering）。非互斥分群（non-exclusive clustering）則用來反應一個物件可以同時屬於多個群集，其中模糊分群（fuzzy clustering）以機率模型探討每個物件屬於不同群集的機率。此兩種分類有時也稱做 hard clustering 和 soft clustering。

3. 完整的（complete）與部份的（partial）：

完整分群（complete clustering）將每個物件指定至一個群集中，而部份分群（partial clustering）卻不是。部份分群的動機為資料集中的一些物件可能不屬於已定義清楚（well-defined）的群集，許多在資料集中的物件可被表示為雜訊（noise）、離群值（outlier）或「不有趣的背景」（uninteresting background）

所有的分群法都有相似的流程，大略可歸納為下列幾點：

a. 收集資料

b. 使用某種方法進行分群

c. 測試分群結果

d. 檢測分群結果，如果未達預期效果，則回到步驟 II，再一次進行分群

6.2 分割式分群法(Partitional Clustering)

分割式分群法其主要目標是要在大量高維的資料點中找出具有代表性的資料點，這些資料點可以稱為是群中心（cluster centers）、原型（prototypes）、代表點（medoids）等，然後再根據這些群中心，進行後續的處理，這些處理可以包含：

資料壓縮：以少數的資料點來代表大量的資料，達到資料壓縮的功能。

資料分類：以少數代表點來代表特定類別的資料，可以降低資料量及計算量，並可以避免雜訊的不良影響。

分割式分群法的目的是希望盡量減小每個群聚中，每一點與群中心的距離平方誤差（square error）。假設我們現在有一組包含 c 個群聚的資料，其中第 k 個群聚可以用集合 Gk 來表示，假設 Gk 包含 nk 筆資料 { x1, x2, …, xnk)，此群

聚中心為 yk。

而這 c 個群聚的總和平方誤差 E 便是每個群聚的平方誤差總和，可稱為分群的「誤差函數」（error function）或「失真度」（distortion）我們分群的方法，就變成是一個最佳化的問題，換句話說，我們要如何選取 c 個群聚以及相關的群中心，使得 E 的值為最小。

6.3 K-Means 分群法(K-Means Clustering)

在所有的分割式分群法之中，最基本的方法，就是所謂的 K-Means 分群法（K-Means Clustering），又稱為 Forgy's algorithm。K-Means 分群法的群中心即是取該群集各資料點的數值平均（mean）而來，又稱為該群集的質量中心。

K-Means 分群法的流程如下：

1.隨機選出 k 個點做為群中心

2.依資料點到群中心的距離分配資料點到最為相近的群集

3.在所有資料點都分配完後，計算該群集質量中心的位置作為群中心

4.重覆步驟二和步驟三，直到群中心不再改變為止

以下是 K-Means 分群法的一個範例：

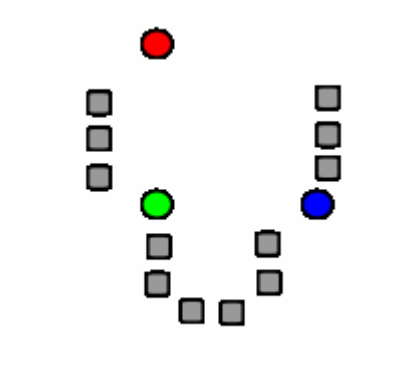


圖 83、首先取出 k(=3)個點作為群中心

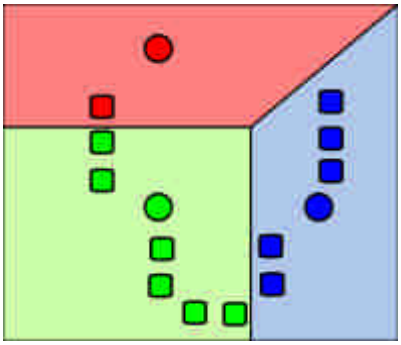


圖 84、依中心點位置將資料分成 k(=3)個群聚

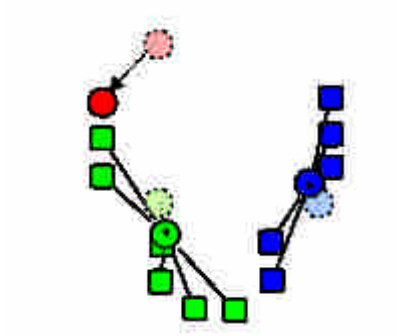


圖 85、根據群聚內成員的屬性決定該群聚的群中心

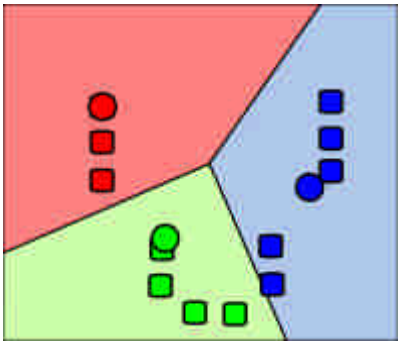


圖 86、反覆進行上述步驟直到群中心不再改變為止

6.4 K-Medoids 分群法(K-Medoids Clustering)

K-Medoids 分群法（K-Medoids Clustering）是 K-Means 分群法的一個變形。K-Medoids 分群法與 K-Means 分群法的不同，在於其群中心是由該群集中最具代表性的點（medoid）來決定，而在 K-Means 分群法中，群中心是由各群集資料點的數值取其平均（mean）而得到，也稱為該群集的質量中心，但這質量中心位置並不一定真實有資料點存在，此時便可以改用 K-Medoids 分群法。

K-Medoids 分群法的流程如下：

1. 隨機選出 k 個點做為群中心

2. 依資料點到群中心的距離分配資料點到最為相近的群集
3. 在所有資料點都分配完後，取各群集中最接近該群集質量中心的點作為新的群中心
4. 重覆步驟二和步驟三，直到群中心不再改變為止

以下是 K-Medoids 分群法的一個範例：

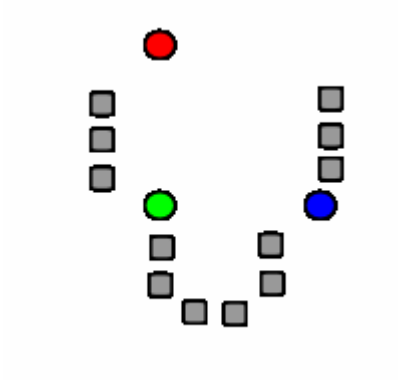


圖 87、首先取出 k(=3)個點作為群中心

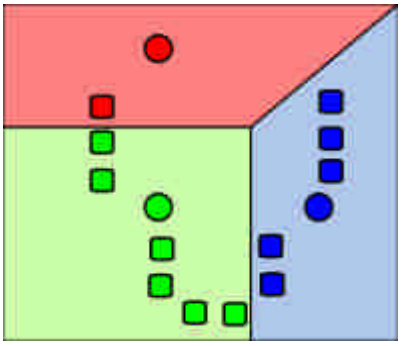


圖 88、依中心點位置將資料分成 k(=3)個群聚

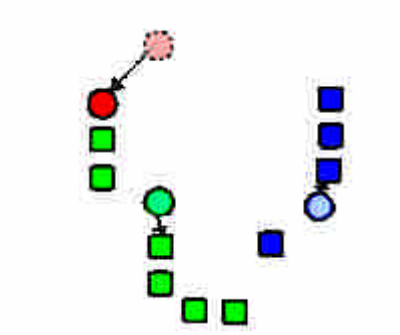


圖 89、取出群集中最接近質量中心的點作為群中心

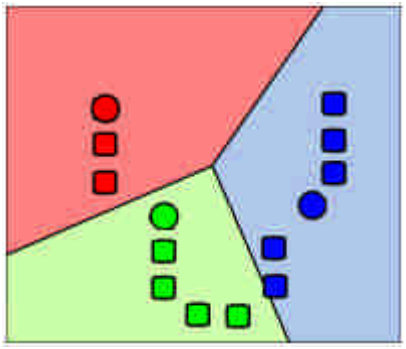


圖 90、反覆進行上述步驟直到群中心不再改變為止

6.5 應用防火牆日誌找出群聚及應用

透過防火牆日誌（firewall log）我們可以得到主機的連線履歷（profile），每一履歷包含有該主機的連線紀錄，針對此連線紀錄，我們可以將擁有相似連線行為的主機分配到同一個群聚裡，並且利用群組內發生異常的主機來進行預警。

1. 分群

假設有 A、B、C 三主機，其中 A 連線到 update server 的次數為 150，B 連線到 update server 的次數為 180，而 C 完全沒有連線到 update server，則我們會認為 A、B 是屬於同一個群聚，C 是另一個群聚，因為 C 和 A、B 的連線行為較不相似。

實際上計算連線履歷的相似度時，除了連線次數，連線對象外，連線的先後順序也會影響兩主機之間的相似程度。透過這樣子的分群，我們可以確保各主機的連線行為與該群聚的其他主機相似程度很高。

相似度計算的部份，我們相每台主機的履歷(profile)依不同時間切成每小時的快照(snapshot)，再根據使用的服務(如 http、ftp 等)不同進行計算兩兩快照間的距離，以下以兩主機 A、B 在同一小時內進行 ftp 連線的快照紀錄做一個簡單的例子。

A	B
192.168.1.250	192.168.1.250
192.168.1.250	192.168.1.252
192.168.1.250	192.168.1.252
192.168.1.252	192.168.2.3
192.168.1.252	

A 主機對 192.168.1.250 進行了 3 次的連線行為，然後對 192.168.1.252 進行了 2 次的連線行為，而 B 主機先對 192.168.1.250 進行 1 次的連線行為，再對 192.168.1.252 進行 2 次的連線行為，最後對 192.168.2.3 有一次的連線行為。將其轉換成序列時如下表：

A	<192.168.1.250, 192.168.1.250, 192.168.1.250, 192.168.1.252, 192.168.1.252>
B	<192.168.1.250, 192.168.1.252, 192.168.1.252, 192.168.2.3>

其中著色部分是 A、B 兩主機相同的部份，而黑色部份是兩主機不同的地方，在此一範例裡，A、B 兩主機之編輯距離為 3，意指，B 主機和 A 主機的差別是，對於 192.168.1.250 的連線少了 2 次，另外多了 1 筆對 192.168.2.3 的連線。在實際計算上，兩兩主機間的編輯距離並未遵守三角不等式等一般度量的性質，故我們採用一修正過的編輯距離(normalized edit-distance)來進行進算，以此評估兩兩主機不相似的程度。這裡附上相關的數學式子：

原先的編輯距離是：

$$GLD(X,Y) = \min\{W(P_{X,Y})\}$$

修正的編輯距離是：

$$d_{N-GLD}(X,Y) = \frac{2 \cdot GLD(X,Y)}{\alpha \cdot (|X| + |Y|) + GLD(X,Y)}$$

$$\alpha = \max\{\gamma(a \to \lambda), \gamma(\lambda \to b), a, b \in \Sigma\}$$

α 在這裡是編輯(增加或刪去)的 cost，一般定義為 1。

另外，由於一主機之履歷(profile)含有多筆快照(snapshot)，所以各主機間的距離為其下對應的快照之距離取平均。

2. 預警

其應用可以用在預警（Early Warning）上，由自定的規則集裡找出違反規則的主機 A，然後針對 A 所屬的群聚裡其他的主機，依照其相似程度我們可以列出一個異常指數（Risk Score），依據異常程度高低給予預警。

在原先的設計架構上，並不依據各自定規則計算 risk score，而是考量該群主機中，已觸發的資安事件主機數。考慮有三個主機群 A、B、C 與二條資安規則 α、β 的情形如下表：

	違反 α 的主機數	違反 β 的主機數	違反規則的主機數總和
A 主機群(50)	1	0	1

B 主機群(20)	0	2	2
C 主機群(40)	2	1	3

在 A 主機群中有 50 台主機，B、C 二主機群內各包含 20、40 台主機，則我們可以清楚了解，在 A 主機群中觸發資安事件的比率為 1/50，在 B 主機群中觸發資安事件的比率為 1/10，而在 C 主機群中是 3/40。相較之下，雖然在 B 主機群中觸發的資安事件總數較 C 主機群中為少，但其嚴重性卻比 C 主機群來得重，故在計算 risk score 時，會將此一比率做為比重加於其上。原始 risk score 的計算則是以各群內主機與觸發資安事情的主機之相似度做為依據，與觸發資安事件的主機越相依，其 risk score 分數越高。

6.6 結論

分群法可以讓使用者將有興趣的資料分成不同的群聚，藉以更加了解各群聚之間的特徵與資料的特性。在以主機連線履歷作為資料分析的情況下，我們可以找出連線行為相近的主機，若該群聚裡有一主機被攻擊，我們可以預期同一群聚裡的主機有較高的機率被攻擊，藉此可以提醒系統管理員對這些主機提高警覺。

6.7 應用防火牆日誌找出群聚之實作詳解

在應用防火牆日誌來找出不同主機間的群聚時，首先要為各主機建立一份履歷（profile），而一份的主機履歷中又以小時為單位將連線資訊整理成數份快照（snapshots），每一份快照中則依據連線的方向進行分類。對於連入主機的連線（in-link connections），依照其服務（services）的不同進行合併，統計使用該服務（如：http，ftp 等）的主機有那些；同時也對連出的連線（out-link connections），依其取用在其他主機的不同服務項目進行合併，統計該主機對外連線的類型。

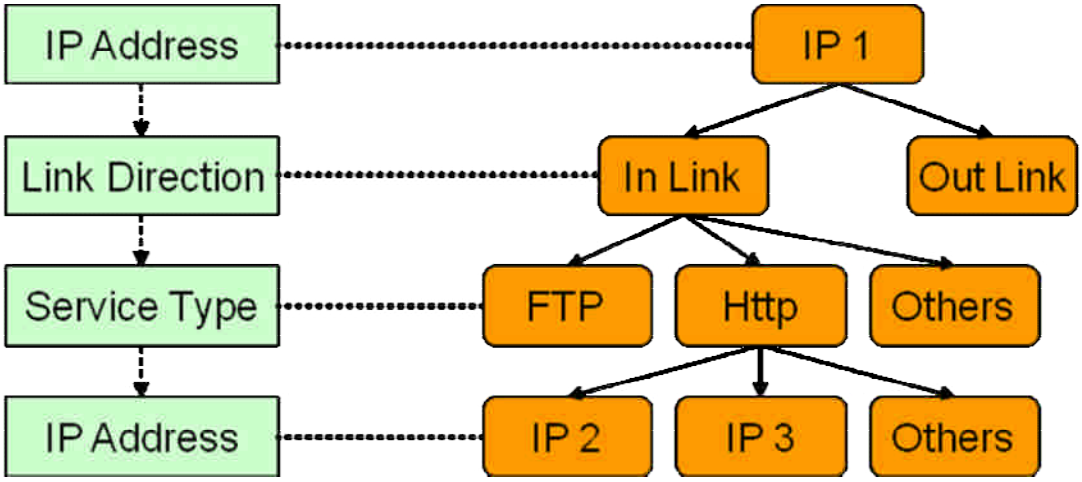


圖 91、連線快照階層圖

對於每一部主機可以用上面階層圖的概念來分析其快照，在每一層中都紀錄了下一層的連線量統計，以 140.113.9.15 這部主機來說，我可以在最上層（IP Address）看到一共有 3248 筆紀錄連入 140.113.9.15 這部主機，而該主機連往外面的記錄只有 53 筆。在第二層（Link Direction）中依服務項目不同統計該特定連線方向的使用量，例如連入 140.113.9.15 這部主機的流量中，取用 ftp 服務的一共有 458 筆，而 http 服務的一共有 1622 筆等。最下一層（Service Type）則紀錄了取用 140.113.9.15 這部主機中某特定服務的主機有那些，其連線次數為多少，如連入 140.113.9.15 這部主機並使用 ftp 服務的主機有 140.113.9.3，140.113.9.5 等，其連線量分別是 51 次和 69 次。

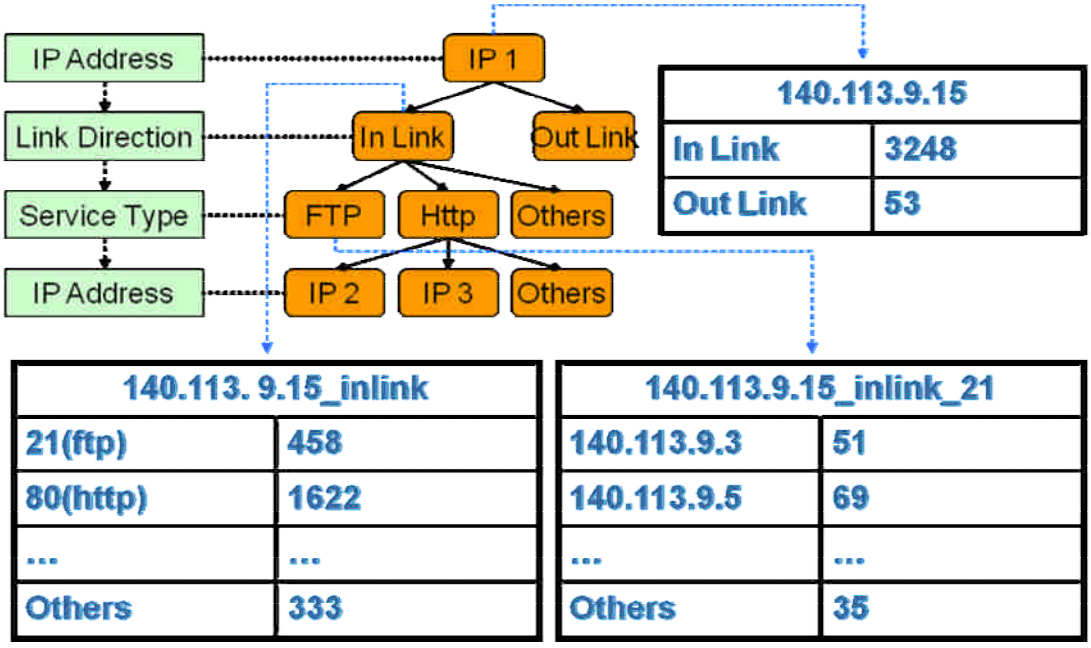


圖 92、連線快照各階層資料紀錄

在比較兩主機 A、B 的履歷相似度之時，我們依序比較其相同時間下的快照相似度並取其平均，式子如下：

$$D_p(p_A, p_B) = \sum_i D_s(s_A^i, s_B^i)$$

而兩兩快照之間的相似度則以再依其連線方向的不同各別進行計算，並加上不同的比重（weight），其式子如下：

$$D_s(s_A^1, s_B^1) = \alpha \cdot D_s^{in}(s_A^1, s_B^1) + (1-\alpha) \cdot D_s^{out}(s_A^1, s_B^1)$$

各快照間同方向的連線紀錄相似度，則將其以連線時間順序轉換成序列（sequence）後，計算其編輯距離（edit distance）作為相似度的依據。

第六章、系統整合

隨著網際網路日益普級，如何保留網路所帶來的資訊傳遞便利性，同時防範有心人士利用網路作為攻擊管道成為網路時代重要的安全議題。雖然目前既有的防火牆與若干入侵偵測系統可以偵測與阻止大部分不法的攻擊行為，但對於新型、無法由已知的規則來判別之入侵行為，將可能因為無法隨時調整或更新其過濾法則與攻擊特徵，造成許多變形攻擊手段可以輕易突破防線，達到侵入的目的。因此將誘捕網系統、威脅分析與預警系統以及封包標記系統進一步整合，並把所收集的資訊分析以列表的方式呈現，為了讓使用者方便操作，也設計人性化的人機介面。針對兩大研究議題，誘捕網系統、威脅分析與預警系統以及封包標記系統核心術研究所收集之現有資料，依上述各系統整合和人機之介面設計，以下分別說明如下：

系統整合主要是在整理各系統回傳的資訊。而系統整合所負責的範圍包括資料的呈現、資料的保存、各系統間互相的通訊及各系統的控制。系統整合還需要確定傳輸的資料是完整及安全的。而此研究主要針對各種異質之網路資料來源做整合，分析出對構建經驗法則有幫助的「資料分析維度」，如：網路服務名稱、網路連線狀態等系統性資料。亦可根據現有的網路歷史資料做初步的統計數據來幫助經驗法則的制定。這些資料都可以成為用以幫助現在的系統上面很少對各個網路攻擊偵測及網路安全防護做系統上的整合，因此整合系統的概念大多建立在此上面。

第一節、中央監控系統

本計畫設置監視控制中心(Monitor and Control Center , MCC)，以利統一監視整個系統的狀況，並且迅速有效地採取應變反應措施，MCC 採用 client-server computing 的組織，client 最主要的功能是把所需要的資訊安全的傳送到 server 端上。而 server 最主要的功能是負責 server 端的呼叫程式執行緒，並且執行它。server 即 MCC，而 client 指誘捕網系統、威脅分析與預警系統以及封包標記系統。Server 端上面分別有幾個重要的部分主要是負責通訊、通訊安全、資訊存取、介面。

1. 通訊

因為各個子系統所用的編寫語言不同所以必須要建立一個 API 提供給使用

者使用，API 是一個介面，它最主要是提供 client 及 server 上面的互動，及 API 的實作將配合各個子系統的作業系統來做修改。Server 端負責的通訊主要是跟 Client 端上面的互動溝通,以及如何把封包完整的傳遞到另一方。Client 與 server 之間的通訊有必要保持機密，我們在傳輸時將進行 SSL 加密，也就是在建立一個溝通管道的時候，雙方首先必須要有金鑰協定，然後利用加密解密的演算法建立一個安全的溝通管道以防資料遭竄改或竊取。

2. 資料保存

各系統在完成資料的傳輸(遞)之後，所有進入 MCC 與由 MCC 發出的訊息均將由 MCC 作成記錄，以備日後分析與追查之用。由於進度與預算的限制，資料保存將採用免費公用軟體 mysql 系統。

3. 介面

分成使用者操控及資料的文字或圖片的呈現,介面主要是從各個子系統的要求來做建立，使用者操控主要是看每個子系統所提供的服務來建立一個介面，使用者可以透過這個介面來使用子系統所提供的服務。

第二節、系統整合規劃

資訊安全的維護大多是從防禦的角度來著手的，例如防火牆、入侵偵測系統、資料加密等。這種思維，是先找出系統可能出現的問題，然後針對問題進行分析以謀求對策。隨著 Internet 的高度發展，各種以網路為基礎的應用系統越來越多，各領域對網路的依賴也越來越強，然而網路的安全議題卻也不斷地面臨著巨大的挑戰。雖然與資訊安全維護相關的研究成果，不斷的推陳出新，但入侵者的技術水平也在不斷地提高，資訊安全技術的演進，似乎永遠跟不上新問題的出現。經常看到的是，某項入侵事件造成了重大損失後，補救措施才因應而生，這讓我們充分理解到，被動的安全維護機制，並不足以確保網路安全。整合雛形系統裡面主要有蜜網、追蹤與預警三個子系統，各子系統的關聯圖如下圖所示。

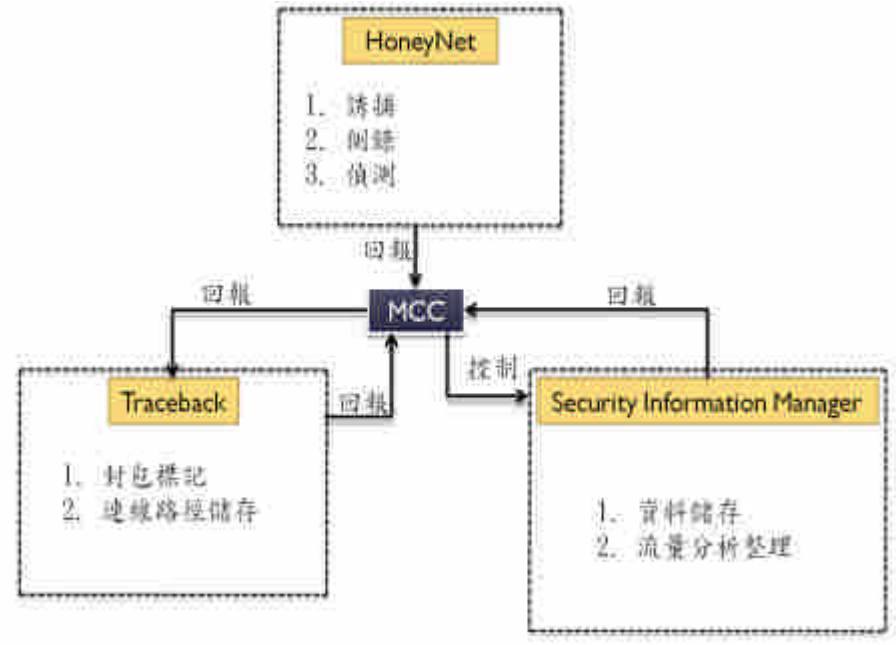


圖 93、系統關連圖

(1) 誘捕網系統 (HoneyNet)

誘補技術的核心是設置一套具備高度隱匿性的監視機制，並記錄入侵者的活動資訊，以針對其行為模式進行分析。誘捕網是一種系統誘捕技術，主要是一種對攻擊者進行欺騙的技術，通過佈置一些作為誘餌的主機、網路服務以及信息，誘使攻擊者對他們進行攻擊，減少對實際系統所造成的安全威脅，並藉以發掘入侵者的入侵過程。

但誘捕網有其自身特點，首先，誘捕網是由多個蜜罐以及防火牆、入侵防禦

系統、系統行為記錄、自動報警、輔助分析等一系列系統和工具所組成的一整套體系結構，這種體系結構創建了一個高度可控的網路，使得安全研究人員可以控制和監視其中的所有攻擊活動，從而去了解攻擊者的攻擊工具、方法和動機。

建置流程首先第一部份便是遇上防火牆，防火牆會對進入的駭客資料封包及來源位址、目標位址、連接的網路服務協定種類以及使用者的身份等，來進行判斷，若是可信任的來源位址就可連到被允許的 internet 上，但是當進入區網內的是駭客，此時的防火牆也會依據資料及區網內部的資安規格來訂定 rule，若是防火牆根據 rules 判斷是惡意的攻擊者時，會直接丟棄掉封包，但是當防火牆無法從 rules 裡判斷時，又對此資料來源可疑時，則會送往蜜罐，經由虛擬蜜罐可以提供多種工具，以方便對攻擊的駭客進行資訊採集、分析。之後蜜罐將分析結果送往 MCC。再由 MCC 來對蜜罐索取有用資訊及下指令。

(2) 網路封包標記追蹤系統(Traceback)

封包標記技術是利用 IP 標頭一些很少用到的欄位，以機率來選擇填入封包經過的部份路徑資訊，縱使攻擊者偽造來源位址，也可以從多個封包的記號找出攻擊路徑資訊，同時從來源路徑資訊若發現不符合 routing 位置的來源也能協助判讀是否為攻擊封包。

建置流程便是網路誘捕系統吸引入侵者，引導入侵者進入受控到的環境之中，此時 Honeynet 則對 MCC 發出警訊，MCC 對 router 下指令索取駭客的封包路徑資訊及偽造來源位址等，進而透過網路追蹤技術，定位到原始的入侵者，並且設置數台主機甚至是一個複雜的網路系統，透過這套誘捕系統可以將駭客侵入系統的一切資訊，包括攻擊過程、使用的工具等全都記錄下來，以供分析之用。分析之後 router 收集的資料便回傳送給 MCC。

(3) 網路威脅風險分析與威脅預警系統(Security Information Manager)

此計畫的主要項目是網路威脅風險分析與威脅預警之核心技術，則是在研發一套針對網路威脅的風險分析模式，以應用在資訊攻防競爭下之預警機制的建立。目前市面上有眾多的入侵偵測系統，但這些偵測系統僅能針對部份可能的網路入侵方式進行監控與防堵。為了有效地分析我們所面臨的所有網路威脅，資安單位必須同時佈署多個不同的入侵偵測系統，並請資安專家對這些入侵偵測系統及相關伺服器進行監控。

根據以上的編輯與驗證工具建構法則後，如何透過法則推論引擎進行網路威

脅分析及風險量化、威脅預警等模式，亦為本計畫之重要研究項目之一。我們將分析現有的網路威脅分析、網路預警等之研究成果，研發法則推論基礎之網路威脅分析與預警模式。

建置流程首先第一部份便是遇上防火牆，防火牆裡訂定了區網內部的一些資安規格，這些規格主要是由區網內部自行定義的，傳入 MCC 之後，MCC 會把 log 拿來分析統計一些數值，由 SIM(指網路威脅風險分析與威脅預警之核心技術)來進行歸納、篩選、設定規則參數，之後在 MCC 上展示。

結合上述 Windows 誘捕網情蒐之核心技術與軟體單元研發、網路威脅風險分析與威脅預警之核心技術與軟體單元研發如下圖說明：

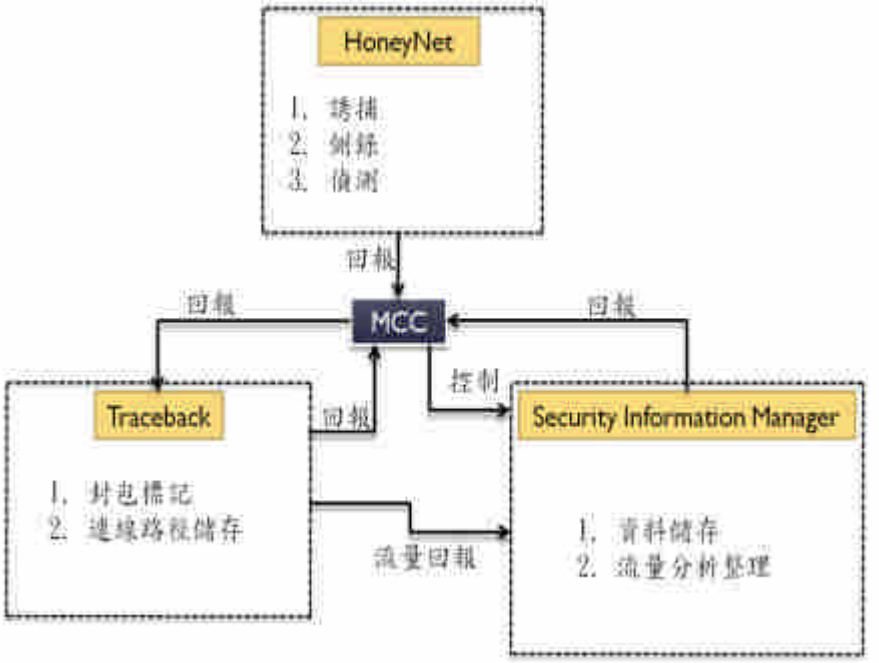


圖 94、系統整合關連圖

整合關聯圖系統流程說明：

誘捕網系統(HoneyNet)誘使攻擊者對其進行攻擊。在攻擊的過程中，系統將秘密地記錄下整個攻擊的過程，包含攻擊者的網路行為(如流量攻擊、封包內容、封包的來源、cpu、memory 系統負載)、登入主機後輸入的指令，甚至於在主機上安裝的木馬後門程式，都將被記錄下來，因此 Honeynet 查出被攻擊的攻擊者之後回報警訊給 MCC，MCC 再聯絡 router，因為駭客進行攻擊時，對防禦方來說同時也提供了機會追蹤攻擊的來源，這對攻擊者將達成有效的嚇阻效果，網路攻擊追蹤技術主要分為三個層面。第一，受害者端方面必須能及早偵測出攻擊封

包，如何從大量接收的封包中判斷何者是合法正常封包（legitimate packets），何者又是被偽造來源位址之惡意封包，這是重建攻擊路徑的基石，一個正確且快速反應的偵測機制將可確保受害者在被癱瘓前即阻斷攻擊者之後續攻擊。接著，判斷出攻擊封包後，如何追蹤回溯到攻擊者端，這部份由於攻擊者可偽造來源端位址，因此必須引進一套封包標記（packet marking）機制來重建攻擊路徑（可搭配封包資料摘要以增加路徑收斂速度），始能徹底找出攻擊者。Router 再把上述分析，有關攻擊者的 router link 及標記等資訊，收集好資訊之後，router 把資訊傳回給 MCC，此時 MCC 再把收集的資訊放入 log 之後通過 data mining，透過分析這些擷取到的結果，可以對新型的網路威脅進行預警，此後再把 rule 傳回 MCC，在 MCC 的 screen 上 show log 及 attack path。

第三節、系統間傳輸加密

本計劃為求系統間傳輸時的安全性，我們在整合系統時，於資料傳輸時都加入 SSL 加密機制，不但可以大幅降低資料被竄改的危險性，還可達到傳輸資料的減量。以下我們將詳細介紹系統傳輸的加密機制。在本系統裡我們採用的則是 Java 裡的 Keytool 建立 SSL 機制。

1. SSL 介紹

SSL 加密是網頁伺服器和瀏覽器之間以加解密方式溝通的安全技術標準，這個溝通過程確保了所有在伺服器與瀏覽器之間通過資料的私密性與完整性，SSL 是一個企業級標準，它被數百萬個網站用來保護他們與客戶的線上交易資訊，而為了使用 SSL 安全連結，一個網頁伺服器需要一張憑證。 讓使用者確認 WEB 伺服器的身份。具有 SSL 功能的軟體（如 WEB 網頁瀏覽器程式）會依照列在這類程式中受信任的認證中心根憑證列表，來自動檢驗伺服器的數位憑證是否有效，且是通過適當的認證中心（如迅通誠信）所發行。鑒別伺服器身份在安全電子交易環境中，對於使用者而言是非常重要的一環。例如，當使用者欲傳送自己的信用卡資訊給網站伺服器之前一定會想先確認該伺服器的身份如何。要求所有在用戶端及伺服器之間所傳遞的資訊由傳送端軟體加密，以及由接受端軟體解密。如此可以保護資訊不會在網路上截獲。另外，所有在 SSL 聯機中所傳遞的資訊亦需要對資訊完整性進行驗證，SSL 會自動檢測資訊在傳遞途中是否有遭

到修改的危險。如此以來，使用者可以安心地將個人資料（如信用卡資訊）傳遞到網站上，並信任 SSL 機制會保護這些資訊的隱私及安全性。

2. SSL 加密流程

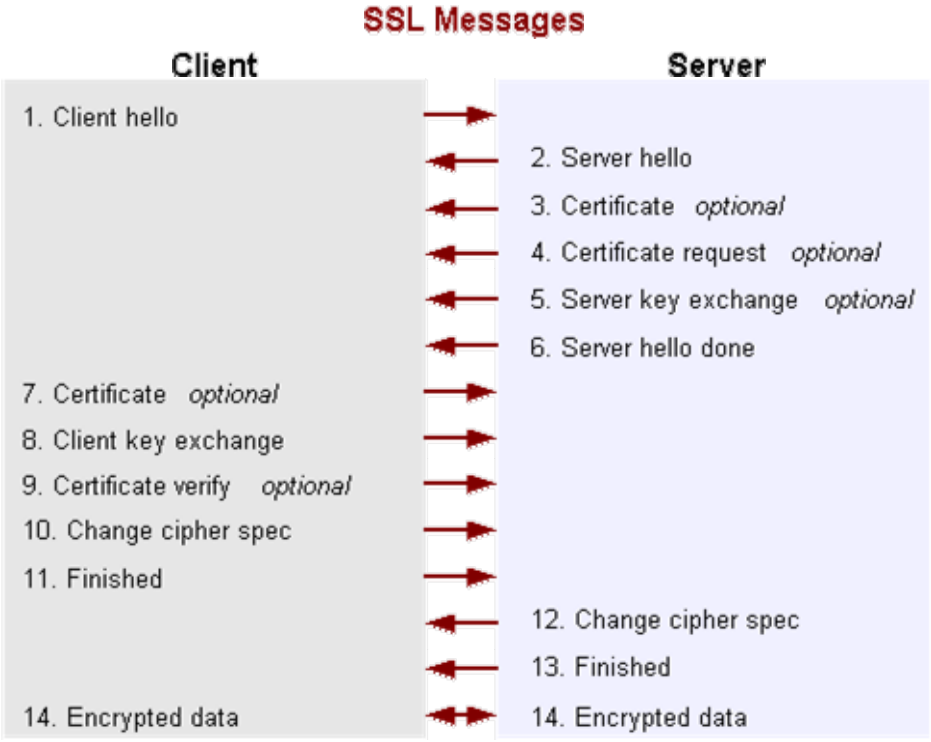


圖 95、SSL 加密流程

1. Client 傳訊息到 Server 確認身分。
2. Server 傳送 Server Public Key 給 client。
3. Client 產生後續傳輸資料需要用來加密/解密用的 Session Key，並以 Server 傳過來的 Public Key 加密後回傳給 Server。

4. Server 用 Private Key 解開 Client 以 Public Key 加密回傳的資料，以取得 Session Key。

5. 後續 Client 與 Server 間傳送資料就以此 Session Key 來做資料加密與解密的處理，處理的時間會比使用 Server 的 Public Key 與 Private Key 要快速許多。

3. Java 使用函式

Client

```
SocketFactory socketFactory = SSLSocketFactory.getDefault();
Socket socket = socketFactory.createSocket(hostname, port);
```

Server

```
ServerSocketFactory ssocketFactory = SSLServerSocketFactory.getDefault();
```

```
ServerSocket ssocket = socketFactory.createServerSocket(port);
Socket socket = ssocket.accept();
```

4. 建立憑證與使用方式

使用 Java 內建 keytool 建立憑證

```
keytool –genkey –keystore mySrvKeystore
```

Client 端與 Sever 端建立連線

- Client

  - java - Djavax.net.ssl.trustStore=mySrvKeystore.dat -Djavax.net.ssl.trustStorePassword=XXXXXX Client

- Server

  - java -Djavax.net.ssl.keyStore=mySrvKeystore.dat -Djavax.net.ssl.keyStorePassword=XXXXXX Server

5. 系統運用

我們主要將 SSL 加密機制用於 MCC 與 SIM 及 Honeynet 之間的傳輸。如下圖所示

圖 96、SSL 加密運用

第四節、系統介面說明

以下我們將詳細解說系統整合完之介面，並在後面章節以實際網路攻擊驗證其功能。

1. 使用者登入介面

進入系統以下為我們第一個看見的介面

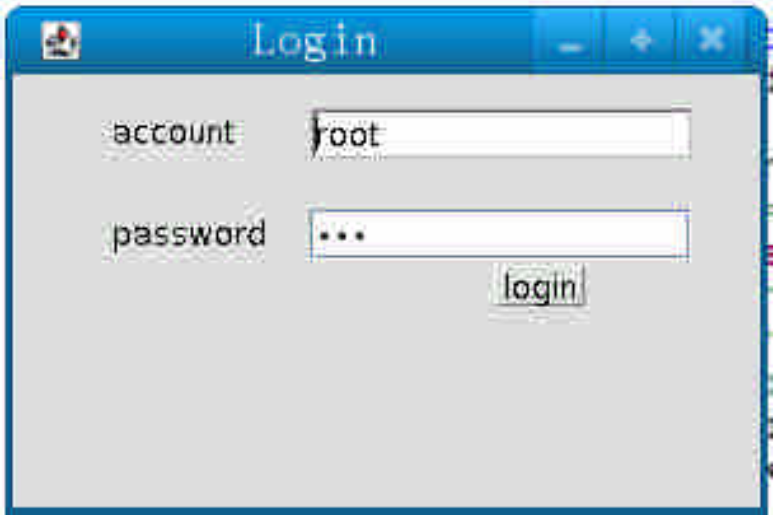


圖 97、使用者登入介面

- 輸入錯誤的帳號密碼，程式立即關閉 。
- 輸入一般使用者帳號密碼，僅開啟 MCC 操作介面。
- 輸入正確 root 帳號密碼，可同時開啟帳號管理介面以及 MCC 操作介面。

2. 帳號權限管理介面

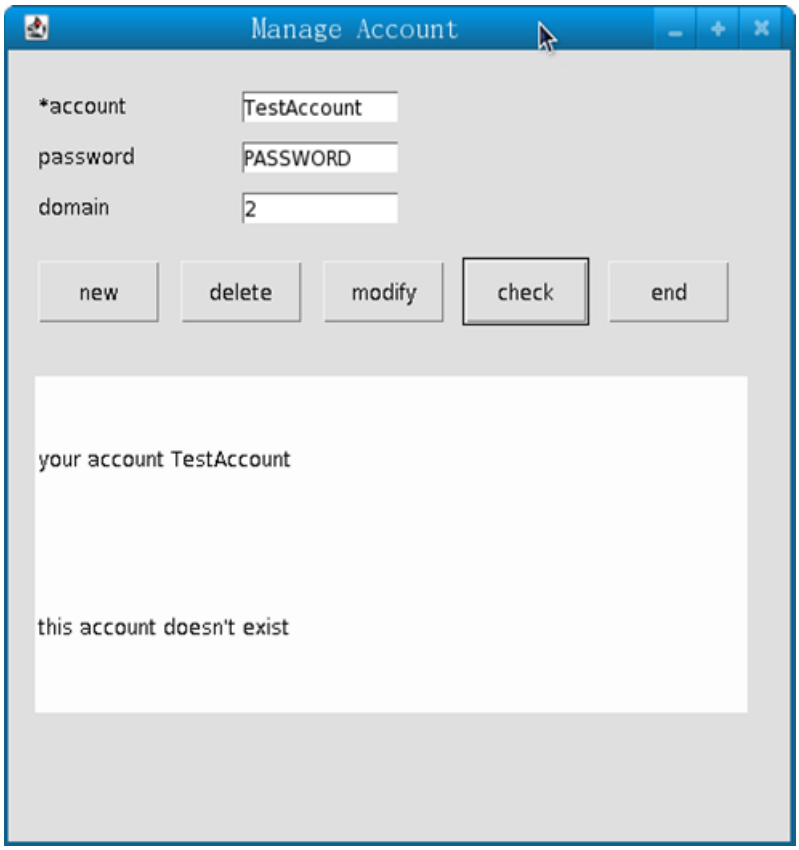


圖 98、帳號權限管理介面

- New:新增帳號。必須輸入帳號、密碼。網域若不符合規定將不予新增。
- Delete:刪除帳號。刪除帳號只需輸入帳號資訊。
- Modify:修改帳號資訊。 必須輸入帳號、密碼、網域。
- Check:查詢帳號。查詢不存在帳號，沒有資訊。反之則顯示密碼、網域。

3. 中央監控系統

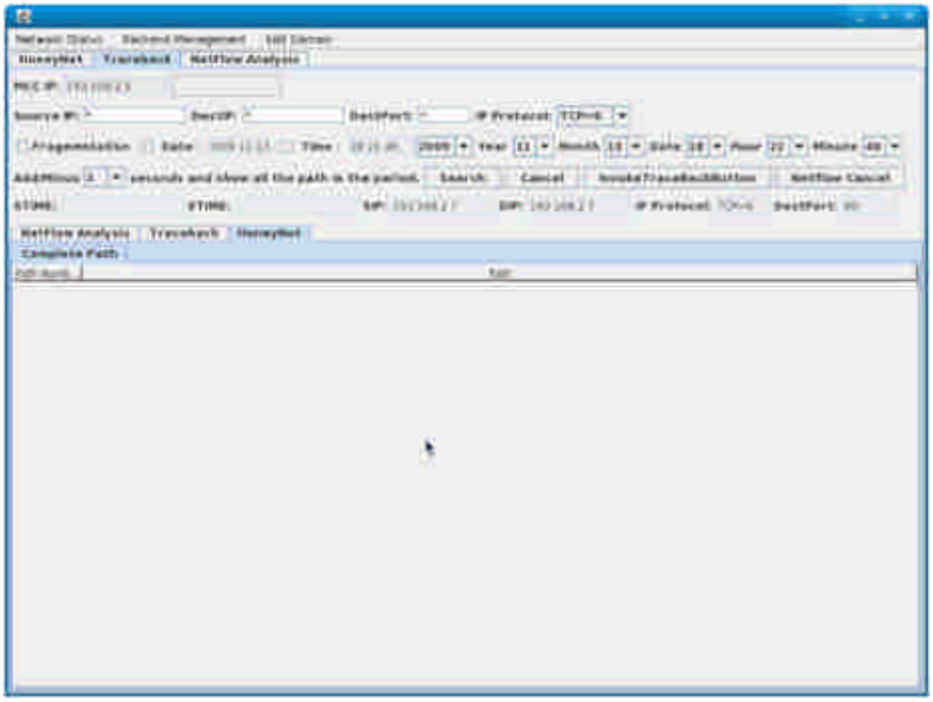


圖 99、中央監控系統

上圖為 MCC 的主要管理介面，藉由此介面我們可以分別對誘捕網系統、網路風險分析及預警系統和封包標記追蹤系統進行監控，使得管理人員可以很輕易的操控各界面。以下我們將講解各界面的功能。

a. 誘捕網管理介面

這裡我們講解的為誘捕網系統的管理介面，藉由此介面，當有攻擊者遭到我們的引誘且攻擊時，我們可以在這裡接收到即時的訊號，並且執行我們的防禦策略。

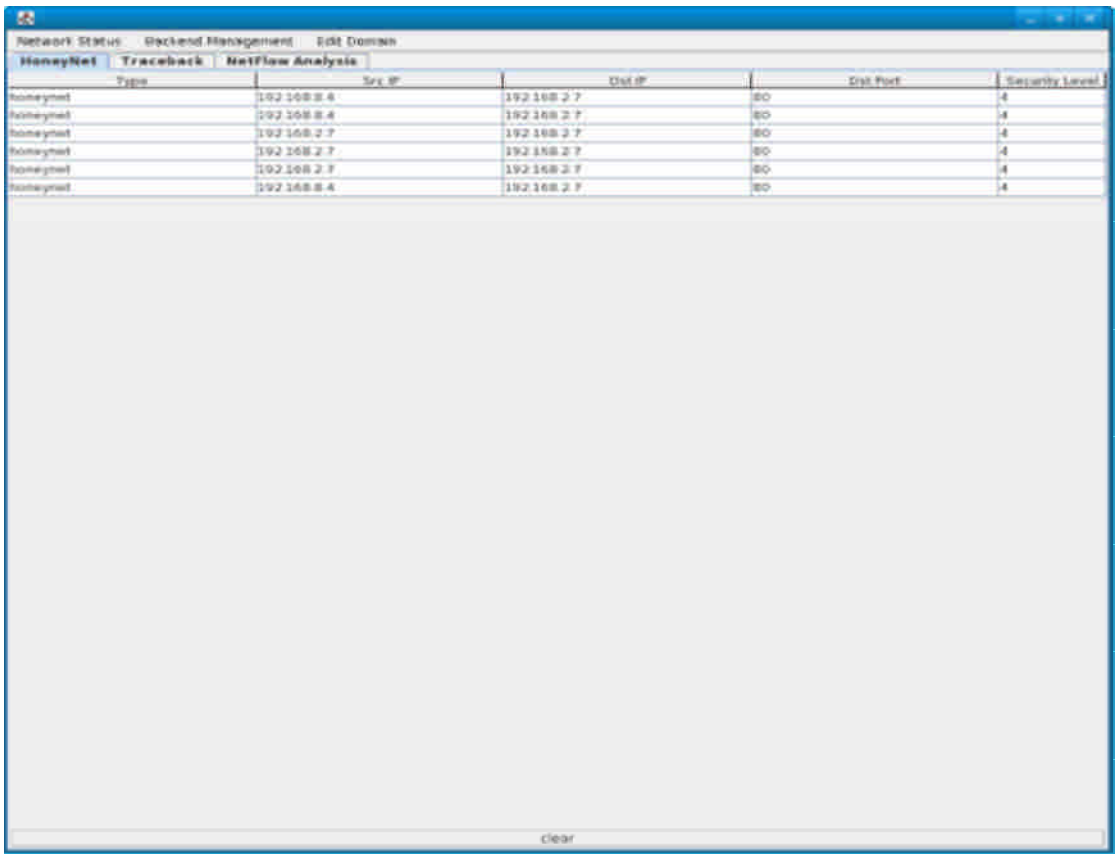


圖 100、誘捕網管理介面

針對此介面我們可以看到有許多標示的地方，說明如下

- Type : 目前受到攻擊的 Web Server。
- Src IP: 攻擊者 IP 位置。
- Dst IP: 受到攻擊的網頁位置。
- Dst Port: 受到攻擊的 Port。
- Security Level: 受到攻擊的防護層級。
- Clear: 清除目前的記錄。

b. 封包標記與追蹤管理介面

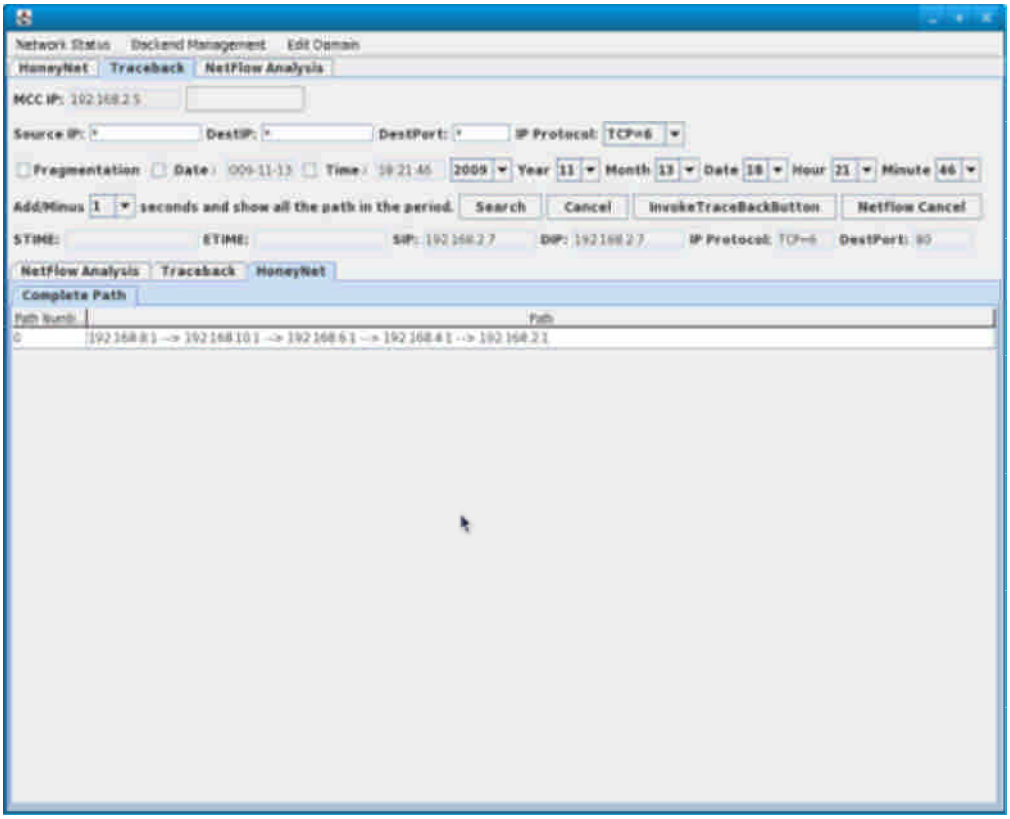


圖 101、封包標記與追蹤管理介面

MCC IP: 在此標示中央監控系統的 IP 位置。

Source IP: 在此設定搜詢路徑的起始 IP 位置。

Dest IP: 在此設定搜詢路徑的最終 IP 位置。

IP Protocal: 設定 IP 協定

Year: 顯示年份

Month: 顯示月份

Data: 顯示日期

Hour: 顯示小時

Minute: 顯示分鐘

Search : 搜詢功能

Invoke TraceBackButton: Traceback 搜尋

Netflow: 取消 Traceback

c. 網路風險分析及預警系統

(1) 主介面

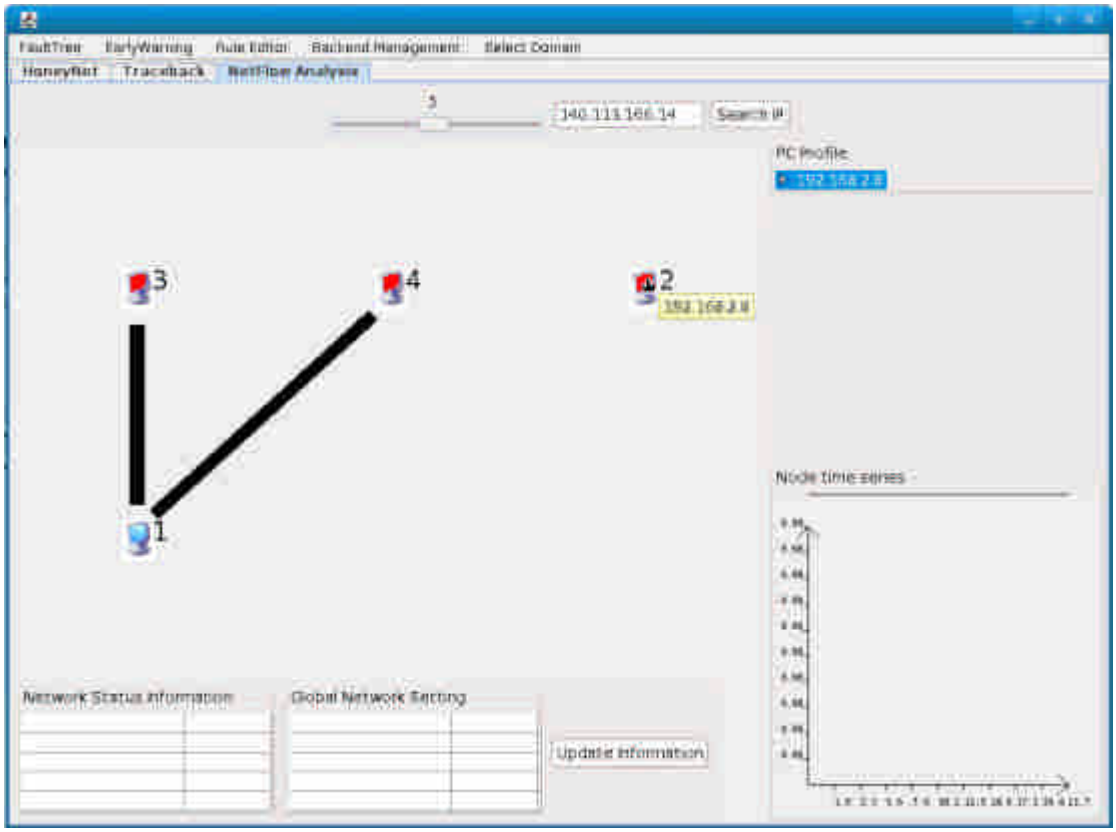


圖 102、可疑主機分析介面

此為風險網路分析及預警系統的主要顯示介面，以下我們將詳細介紹：

- 紅色主機：可疑主機，在圖中我們可以發現標示 2、3、4 的主機為紅色主機，它們被系統辨識為可疑主機。
- 藍色主機：正常主機，圖中標示 1 的主機為藍色主機，它被系統辨識為正常主機。
- 黑線：我們可以看見主機與主機之間有黑線連接，表示彼此之間的傳輸，當線越粗的時候表示傳輸量越大，反之，當線越細時，傳輸量就越小。
- PC Profile:當我們點選後會顯示該點選主機的連線資訊，表示該主機有傳出資料致哪些主機，或式從哪些主機接收訊息。
- Node Time Series: 當我們點選了某台主機後，會這裡列出該主機的連線狀況，讓我們知道在某時間點的傳輸量。

  藉由此介面我們可以掌控區域網路內所有主機的狀況。

(2) 法則編輯器介面

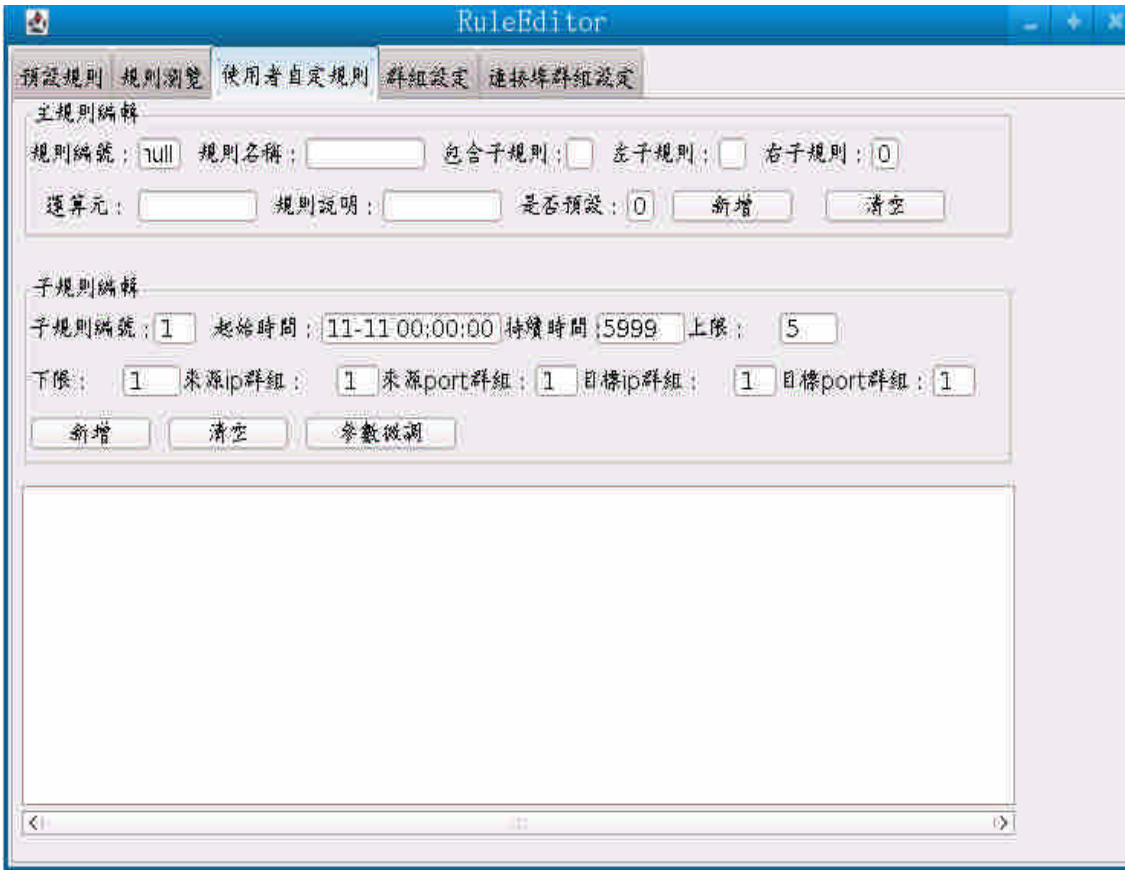


圖 103、法則編輯器

此介面我們可以點選工具列中的 Rule Editor，皆呼叫出此介面。此界面解說如下：

- 預設規則：這儲存原本系統內建的資安規則，共有五項，為公認最可判定可疑主機的資安規則。

- 規則瀏覽：在此我們可以瀏覽所有資安法則，系統管理者可以藉由此進行區域網路內的管理。
- 使用者自訂規則：管理者可以針對各區域網路特性，藉由此功能來新增資安法則。

- 群組設定：管理者可藉由此進行群組的設定
- 連接阜群組設定：在此我們可以針對連接阜進行群組的設定。

(3) 失誤樹編輯器介面

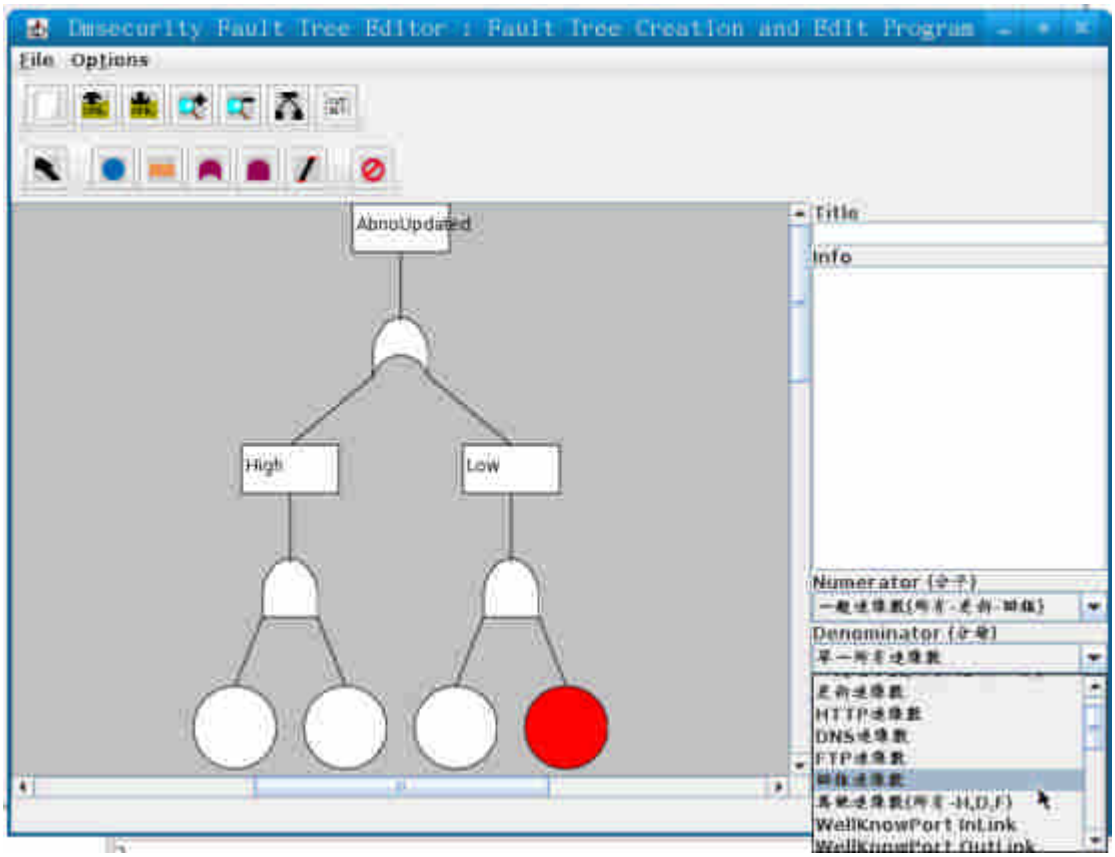


圖 104、失誤樹編輯器介面

此介面我們可以點選工具列中的 Fault Tree，皆呼叫出此介面。藉由此我們可以便改裡面的數值以及結構。

(4) 風險分析介面

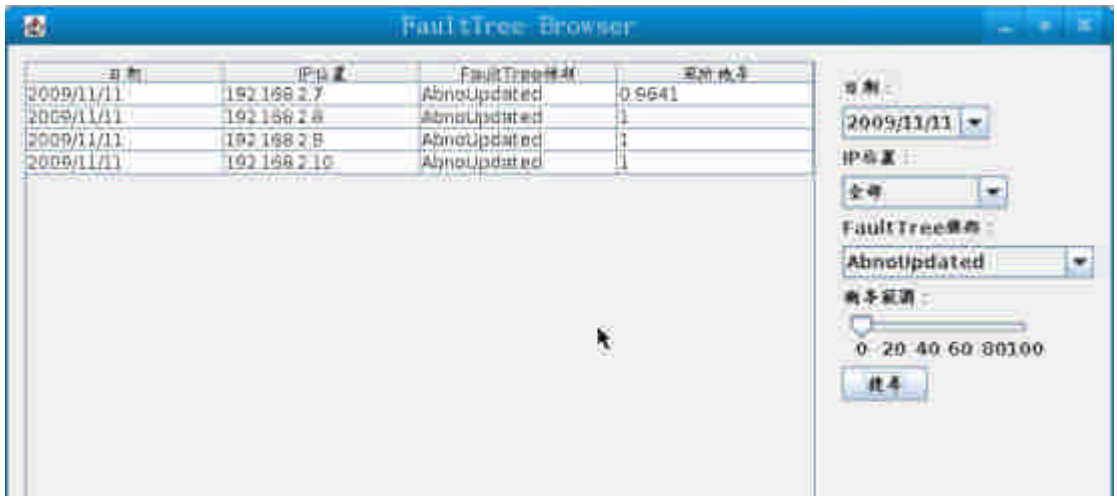


圖 105、風險分析介面

在此介面裡我們可以看見各主機為可疑主機的機率，圖中顯示的為違反資安法則的機率以及主機位置，右方也提供相關得搜尋功能。

第七章、議題一－Windows 誘捕網情蒐之核心技術與回追技

術軟體發展

第一節、議題研究及說明

隨著網路快速發展，訊息快速的流動，人們的生活因此而便利，但隨之而起的試各式各樣的電腦病毒，電腦病毒是一種具有自我複製能力的電腦程式，往往可以嚴重影響受感染的電腦，其主要透過網路瀏覽以及下載、電子郵件以及可移動磁碟等等進行傳播。根據中國國架電腦病毒應急處理中心發表的報告統計，病毒中接近 45%的病毒為木馬程式，木馬病毒一般可稱為遠端監控軟體，如果木馬能連通的話，那麼可以說已經得到了遠端電腦的全部操作許可權，操作遠端電腦與操作自己電腦沒什麼大的區別，這類程式可以監視被控使用者的網路攝影機與截取密碼。而 Windows NT 以後的版本自帶的「遠端桌面連線」，如果被不良使用者利用的話，那麼也與木馬沒什麼區別。使用者一旦中毒，就會成為「喪屍」或被稱為「肉雞」，成為駭客手中的「機器人」，通常駭客或指令碼小孩（script kids）可以利用數以萬計的「喪屍」發送大量偽造包或者是垃圾資料包對預定標的進行拒絕服務攻擊，造成被攻擊標的癱瘓。此外網路蠕蟲佔了總病毒數量的 25%，蠕蟲病毒漏洞利用類，也是我們最熟知的病毒，通常在全世界範圍內大規模爆發的就是它了。如針對舊版本未打更新的 Windows XP 的衝擊波病毒和震蕩波病毒。有時與殭屍網路配合，主要使用緩衝區溢位技術其餘剩下的病毒大致式檔案型、破壞型和巨集病毒。

然而，針對各式各樣的病毒，早有許許多多的防毒軟體誕生如 Kaspersky Anti-Virus、NOD32、Avira AntiVir 等等，用於消除電腦病毒、特洛伊木馬和惡意程式的軟體，可以算是電腦的防禦系統。防毒軟體的實時監控方式因軟體而異。有的防毒軟體，是透過在內部記憶體裡劃分一部分空間，將電腦裡流過內部記憶體的資料與防毒軟體自身所帶的病毒函式庫（包含病毒定義）的特徵碼相比較，以判斷是否為病毒。然而依靠特徵碼進行防禦的機制確有以下弱點：

1. 無法變識未知病毒(現在已經有部分殺毒軟體有主動防護和啟發式殺毒功能)。
2. 查到病毒後，不能夠徹底清除病毒。
3. 保護自身。目前有些病毒，能夠終止防毒軟體的行程，再繼續破壞。
4. 軟體所耗用的資源相當的龐大。
5. 更新速度問題。如果網速過慢，則防毒軟體不能夠進行各種更新。

而其最大的問題則是由於越來越多的病毒誕生，使用特徵碼判定的方法將可能導致過大的資料庫，以及更長的判定時間，於是在本計劃裏面我們將採取不同的防禦策路，利用網路誘捕技術協助傳統 IDS 進行網路的防禦。

網路誘捕技術是一種欺騙駭客攻擊的技術，它被用來吸引入侵者，使他們進入受控的環境之中(Data Control)，並使用各種監控技術來捕獲入侵者的行為(Data Capture)。網路誘捕技術的核心是設置一套具備高度隱匿性的監視機制，並記錄入侵者的活動資訊(Data Collection)，以針對其行為模式進行分析(Data Analysis)，進而透過網路追蹤技術，定位到原始的入侵者(Trace Back)。

網路誘捕系統則是應用網路誘捕技術，設置數台主機甚至是一個複雜的網路系統，透過這套誘捕系統可以將駭客侵入系統的一切資訊，包括攻擊過程、使用的工具等全都記錄下來，以供分析之用。事實上，網路誘捕系統往往也具有混淆駭客選定攻擊目標的作用，以對真正運作中的伺服器，提供絕佳的掩護功能。

現行針對網路誘捕系統的研究有兩大類，其一是蜜罐(Honeypot)，另一則是所謂的蜜網(Honeynet)。一般而言，蜜罐是一種具備誘捕能力的電腦系統，它藉著置放某些具備吸引力的資源，來吸引駭客的攻擊。通常蜜罐系統不修補系統的安全漏洞，以使入侵者有極大的發揮空間，以期擷取他們更多的訊息。但是由於傳統的蜜罐系統，並未與自身的網路系統有所區隔，一旦蜜罐被攻破，入侵者反而會利用這個蜜罐系統作為跳板，攻擊其他的系統，因此近年此一領域的發展，逐漸導向了蜜網的

概念。

雖然蜜網也是由蜜罐所組成，但它不是單一的系統，而是一個網路。一個典型的蜜網系統通常由防火牆、路由器、入侵偵測系統(IDS)及數個蜜罐系統所組成，其中防火牆及入侵偵測系統可以對進出蜜網的資訊，執行資料控制(Control)與捕獲(Capture)的任務，並據以獲取入侵者的基本資料。蜜網內部可以設置多種類型的作業主機(如 Linux、Solaris、Windows、FreeBSD 等)來充當蜜罐系統，以提供入侵者一個更加真實的網路環境的感受。藉著各個主機系統所提供不同伺服器的作業環境，也將易於探索駭客所使用的工具及其手法。

在駭客進行攻擊時，對防禦方來說同時也提供了機會來追蹤攻擊的來源，這對攻擊者來說將會達到有效的嚇阻效果。網路攻擊追蹤技術主要分為三個層面。首先，受害者端方面必須能及早偵測出攻擊封包，如何從大量接收的封包中判斷何者是合法正常封包（legitimate packets），何者又是被偽造來源位址之惡意封包，這是開始攻擊路徑重建的基石，一個正確且快速反應的偵測機制將可確保受害者在被癱瘓前即阻斷攻擊者之後續攻擊。接著，判斷出攻擊封包後，如何追蹤回溯到攻擊者端，這部份由於攻擊者可偽造來源端位址，因此必須引進一套封包標記（packet marking）機制來重建攻擊路徑（可搭配封包資料摘要以增加路徑收斂速度），始能徹底找出攻擊者。本計畫裡還結合封包標記與追蹤系統，在我們吸引到攻擊者後，馬上進行攻擊路徑的回追，可避免攻擊者進行 IP 的假冒，我們在接下來的章節裡將進行進一步的解說。

第二節、議題實作

針對此議題我們防禦策略的示意圖如下：

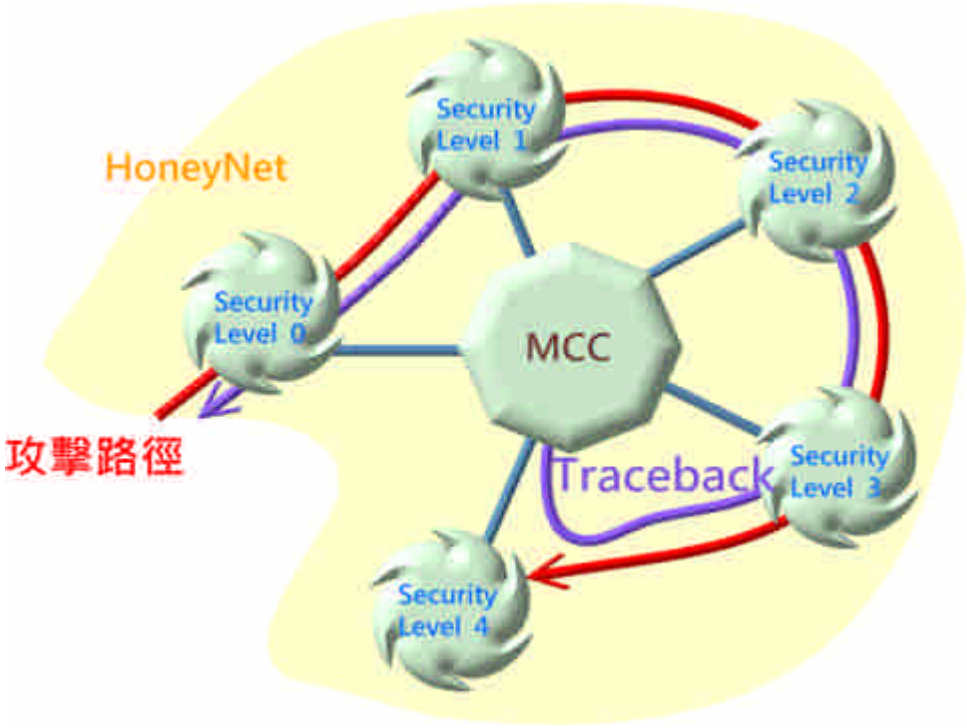


圖 106、防禦策略示意圖

其中紅色路徑代表攻擊者入侵的入進，攻擊者在攻擊網頁時，我們隨著他的攻擊方式逐漸改變防護等級，讓攻擊者誤以為此網頁有逐漸在修補漏洞，是重要網頁，再施以不同的網頁攻擊方法，但實際中，在每個防護層級遭受攻擊後都回報中央監控系統(MCC)，讓管理人員知道有攻擊者入侵，並且由封包標記與追蹤系統進行攻擊者位置的追蹤。我們詳細的記錄攻擊者行為，作為防網路防禦的參考，我們可藉此發現新型的攻擊，使其他電腦主機得以預防，而不是緊緊依靠傳統的 IDS 去防禦。以下我們解釋我們如何實現此防禦策略。

下圖為我們在本議題實作之系統架構圖:

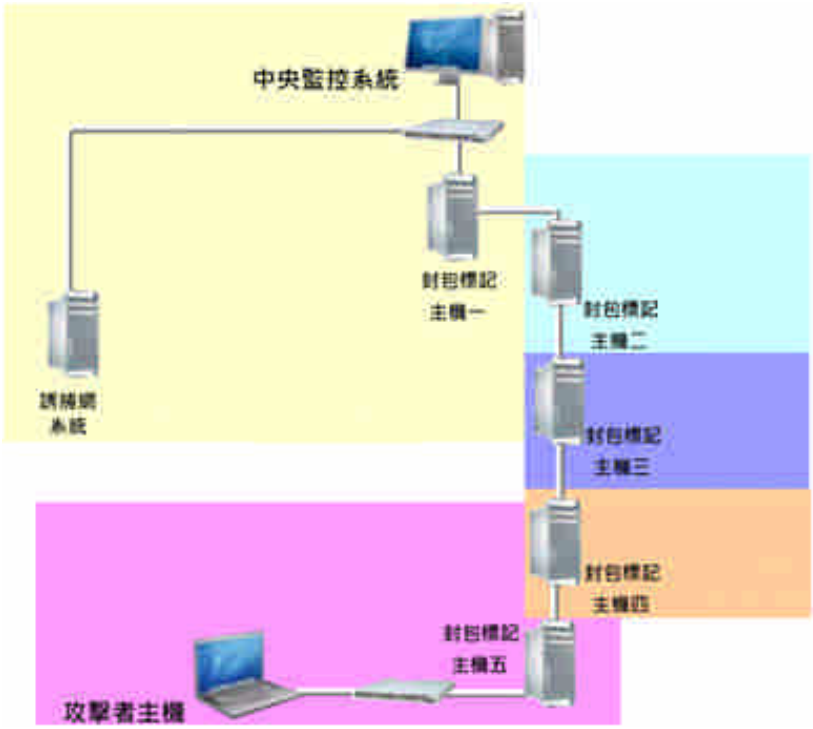


圖 107、系統架構圖

本系統主要由七台主機構成，各顏色區塊代表者不同的網域，其中誘捕網系統主機主要負責誘捕攻擊者，當攻擊者攻擊了此系統，會將攻擊行為訊息傳回中央監控系統，接著各台封包標記主機可針對傳輸的封包進行標記，管理者可藉由中央監控系統進行攻擊封包的路徑查詢，發現攻擊者位於哪個網域，避免使用者有 IP 假冒行為。

第三節、實境模擬與成果展示

在本節我們將以實際的網路攻擊進行系統功能的測試，以下為我們假定的攻擊者入侵情境。

第五營區陳上士在網路上習得 SQL Injection 相關知識，於是企圖運用此手法入侵架設在第一營區的國軍入口網站，竊取他人帳號密碼，進一步取得機密資料。另一方面，王上校登入中央監控系統監控，攻擊方法及路徑如下圖所示：

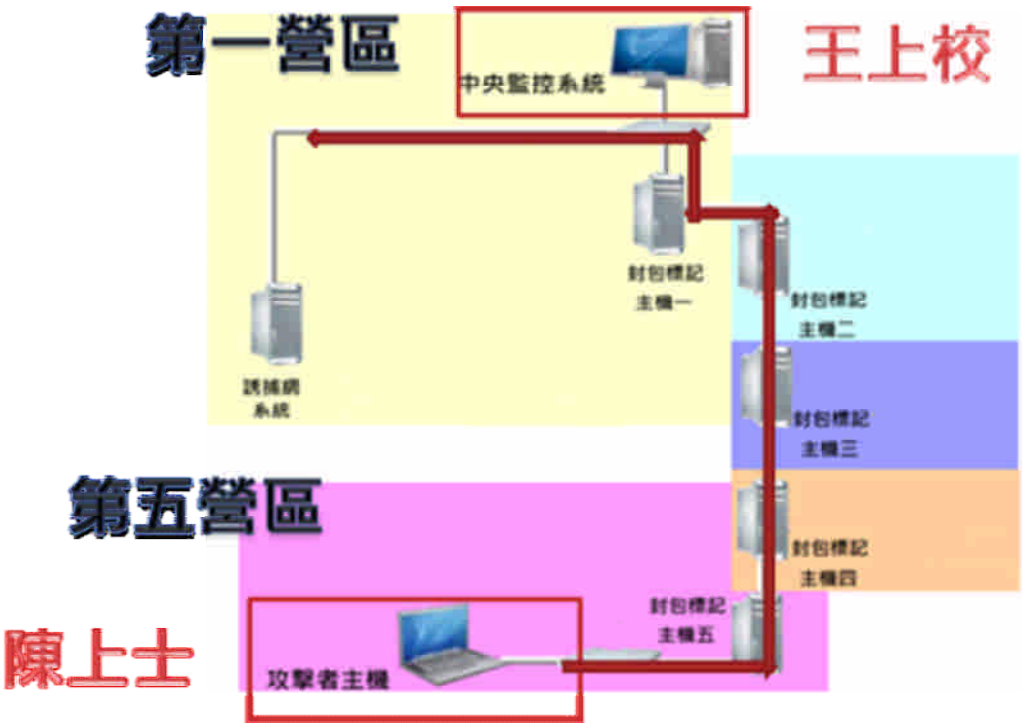


圖 108、實境模擬

於情境模擬前我們線列出網路位置配置表：

攻擊者主機	192.168.8.4	中央監控系統	192.168.2.5
誘捕網系統	192.168.2.7	封包標記主機一	192.168.2.1
封包標記主機一	192.168.4.1	封包標記主機一	192.168.6.1
封包標記主機一	192.168.10.1	封包標記主機一	192.168.8.1

表格 35、網路位置配置表

(1) 陳上士第一次攻擊

陳上士在第一次攻擊時，採用較為簡易的入侵方式輸入特殊字元：

- 攻擊原理及目的

- 透過特殊字元，使攻擊 SQL 成立，藉此規避檢查並且取得登入權限，在本範例哩，王尚是在帳號部份輸入攻擊字串，密碼則不受限制。

◦ 攻擊字串範例

'or 1=1 #

◦ 攻擊畫面

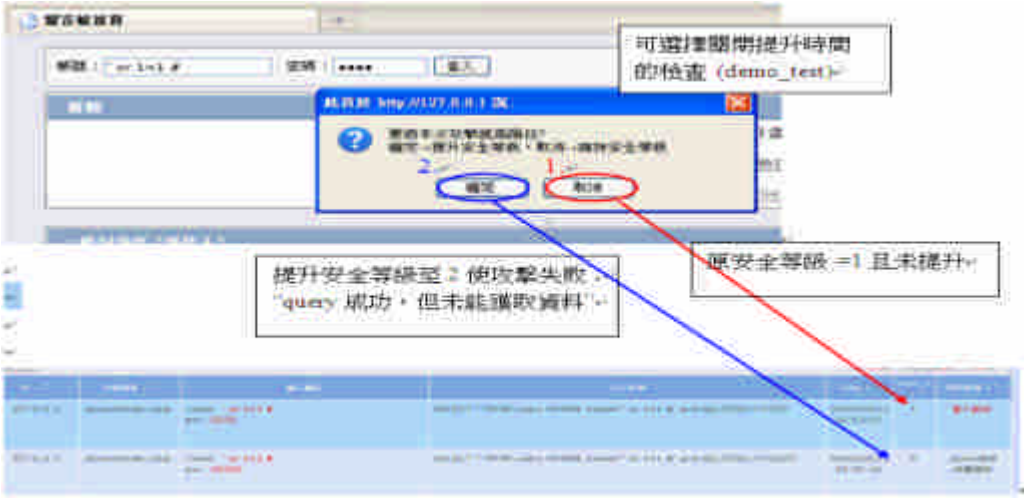


圖 109、攻擊畫面

◦ 攻擊成功：

攻擊成功後我們發現王上士並無輸入正確密碼便可登入系統，特殊字元攻擊成功，以下為攻擊成功後的畫面。

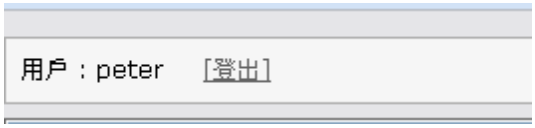


圖 110、成功畫面

(2) 系統防禦

在陳上士攻擊成功後，中央監控系統收到預警訊息，如下

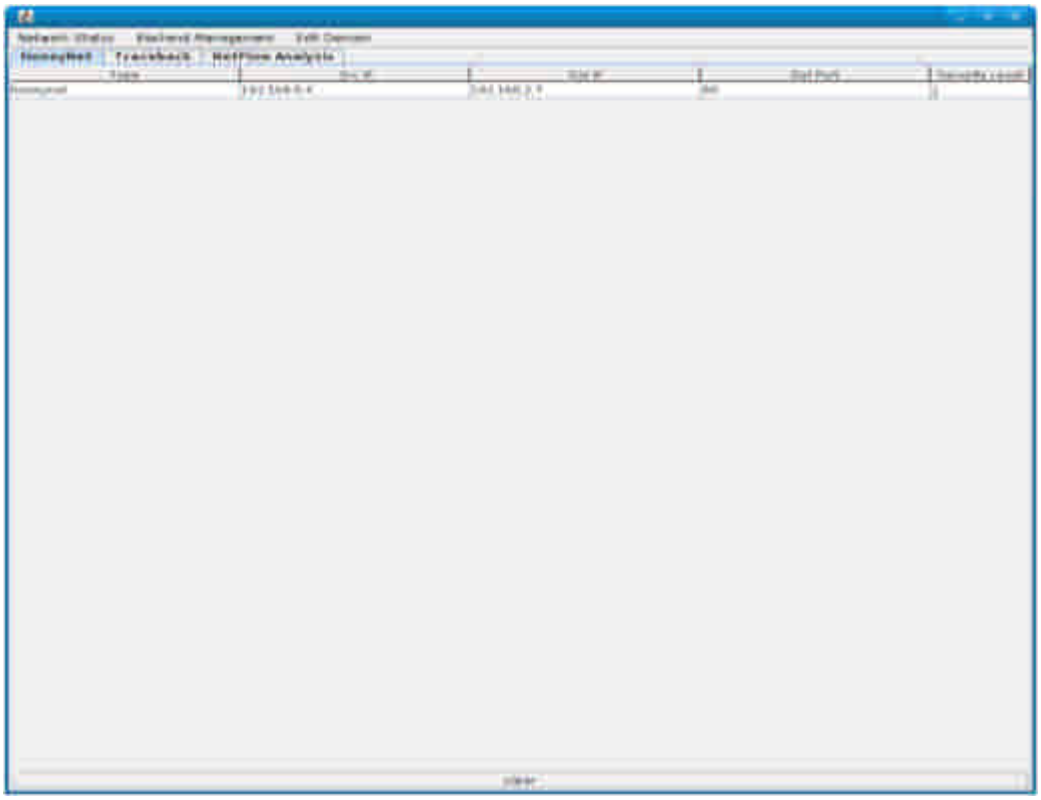


圖 111、系統防禦

管理者可發現誘捕網捕獲攻擊者，並且是從 IP: 192.168.8.4 位置對我們誘捕網主機位置 IP: 192.168.2.7 進行 Security Level 1 進行攻擊。此時我們提升安全層級，讓攻擊者誤以為我們修補完漏洞，並且網頁是有持續更新的。同時我們切換到封包標機及追蹤系統管理介面，系統已經自動追蹤路徑成功，圖如下：

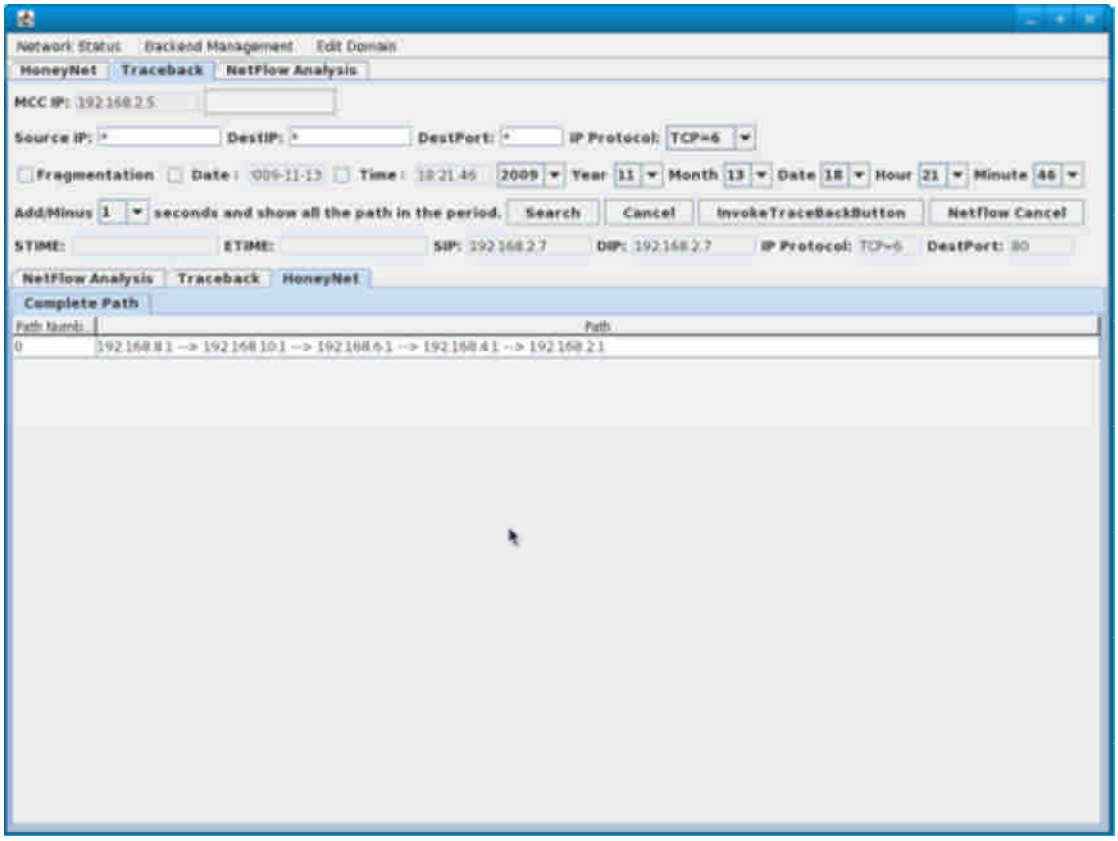


圖 112、路徑追蹤

我們可發現攻擊者的攻擊路徑已經被我們追蹤出來，在本次攻擊裡，陳上士經過的網域是：

192.168.8.1->192.168.10.1->192.168.6.1->192.168.4.1->192.168.2.1

證實本次攻擊裡，攻擊者未假冒 IP 位置，系統已掌握了攻擊者。

(3) 陳上士第二次攻擊：

陳上士在防護層級提升後仍然進行特殊字元攻擊，但由於系統防護等級已經提升了，故本次他失敗了，如下圖所示：

圖 113、攻擊失敗

(4) 陳上士第三次攻擊：

陳上士在第二次攻擊失敗後，認為網頁修補了漏洞，於是他找尋新的攻擊方式，使用新的攻擊方式，本次嘗試錯誤嘗試型攻擊。

◦ 攻擊原理及目的：

- 網站所提供的錯誤訊息，原本是讓開發者除錯用，但攻擊者同樣可以利用錯誤訊息來反覆詢問主機，順藤摸瓜地竊取出重要資訊，進而擁有登入權限。

◦ 攻擊字串範例

- 原本網址

http://140.115.53.35/board/board.php?id=3

- 更改如下下字串

http://140.115.53.35/board/board.php?id=9 **UNION ALL SELECT NULL**

◦ 攻擊畫面

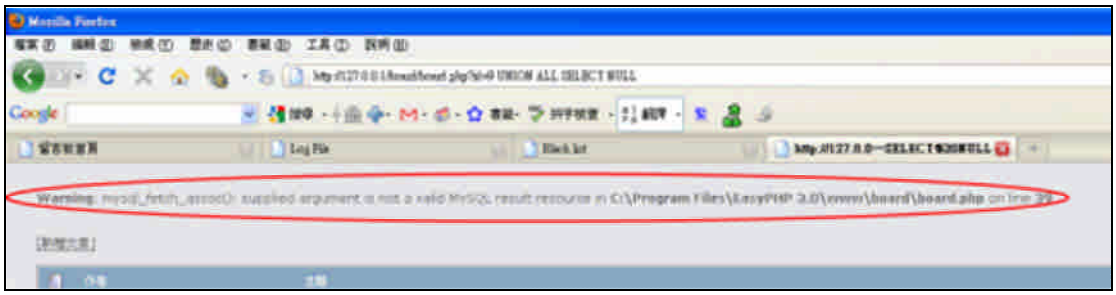


圖 114、攻擊畫面

此時陳上士發現出現了錯誤訊息，於是開始進行錯誤嘗試，在試了幾次後發現錯誤訊息消失，代表攻擊成功。

◦ 嘗試成功之字串

http://140.115.53.35/board/board.php?id=9 UNION ALL SELECT NULL, NULL, NULL

◦ 嘗試成功畫面

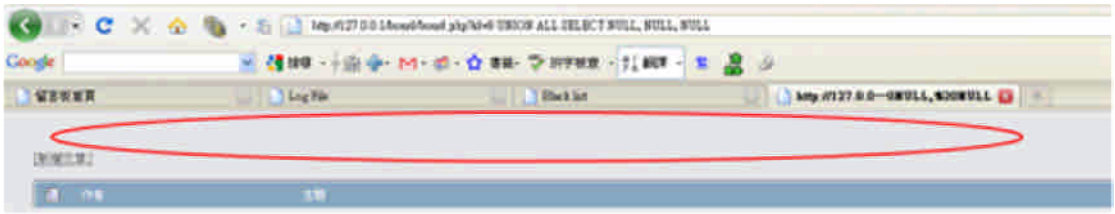


圖 115、嘗試成功畫面

在嘗試成功後，陳上士知道資料庫裡的欄位僅有 3 欄，陳上士開始猜測資料表的欄位名稱，首先觀看網頁原始碼。

```
<form action="index.php" method="post" class="middletext">
        帳號 : <input name="name" size="20" type="text" >
          密碼 : <input name="pw" size="10" type="password">
        <input value="登入" type="submit">
</form>
```

圖 116、網頁原始碼

欲取得資料欄位名稱，首先陳上士必須先取得資料庫名稱，於是他在網誌列述入以下指令：

```
http://140.115.53.35/board/board.php?id=9 UNION ALL SELECT
NULL,TABLE_SCHEMA,NULL  FROM INFORMATION_SCHEMA.COLUMNS
limit 352,900
```

並成功取得名稱如下圖：

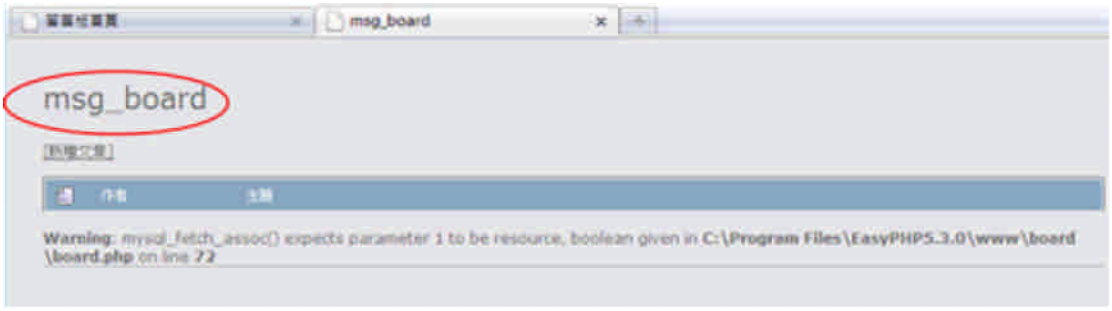


圖 117、取得資料庫名稱

陳上士此時知道資料庫名稱為 msg_board，並企圖藉由此得取資料表名稱，輸入以下指令:

```
http://140.115.53.35/board/board.php?id=9 UNION ALL SELECT
NULL,TABLE_NAME,NULL  FROM INFORMATION_SCHEMA.COLUMNS
limit 352,900
```

輸入完之後得到資料表名稱如下：

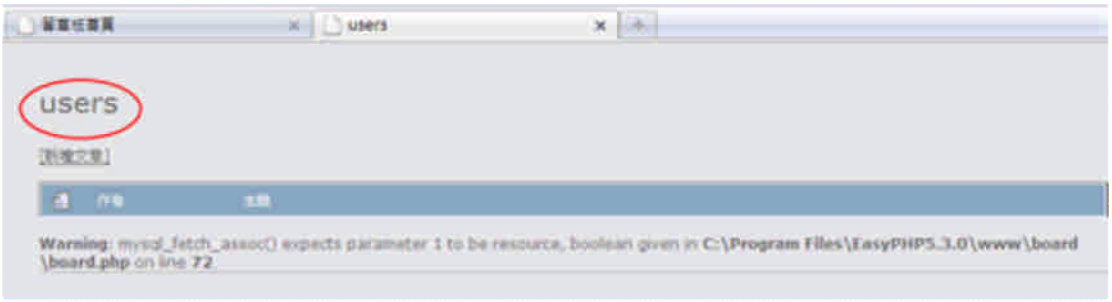


圖 118、取得資料表名稱

陳上士取得資料名稱後便可藉由以下指令取得欄位名稱，指令如下

http:// 140.115.53.35/board/board.php?id=9 UNION ALL SELECT NULL,name,NULL FROM users

畫面如下：

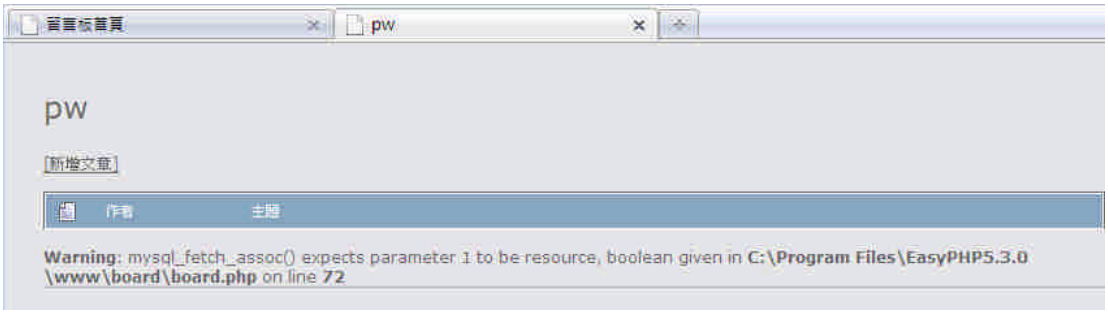


圖 119、取得欄位名稱

最後根據欄位名稱陳上士便可以取得密碼以及帳號，取得帳號的指令如下：

http:// 140.115.53.35/board/board.php?id=9 UNION ALL SELECT NULL,name,NULL FROM users

取得帳號畫面：

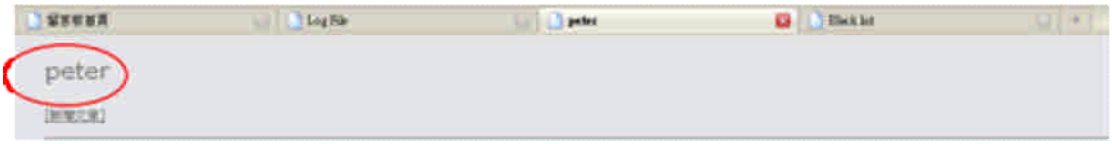


圖 120、取得帳號

同樣我們可以藉由以下指令取得密碼

http:// 140.115.53.35/board/board.php?id=9 UNION ALL SELECT NULL,pw,NULL FROM users

到此陳上士的攻擊行為結束，並且認為他已經竊取到了帳號密碼。

(5) 系統防禦

然而從陳上士第一次攻擊開始到最後，中央監控系統這邊都完整的記錄其過程，並且可以知道陳上士在誘捕網裡的行動，記錄如下：

輸入欄位	SQL字串	Date	user_lv	用戶狀態	Input_Lv / 攻擊狀態	Comments
tid : 9 UNION ALL SELECT NULL,pw,NULL FROM users	SELECT * FROM thread WHERE tid=9 UNION ALL SELECT NULL,pw,NULL FROM users	2009/10/12 13:57:31	2	獲取資料成功	L1 : 獲取資料成功 L2 : 獲取資料成功 L3 : query失敗 L4 : query失敗	安全等級維持 L2
tid : 9 UNION ALL SELECT NULL,name,NULL FROM users	SELECT * FROM thread WHERE tid=9 UNION ALL SELECT NULL,name,NULL FROM users	2009/10/12 13:57:25	2	獲取資料成功	L1 : 獲取資料成功 L2 : 獲取資料成功 L3 : query失敗 L4 : query失敗	安全等級維持 L2
tid : 9 UNION ALL SELECT 1111,2222,3333 FROM users	SELECT * FROM thread WHERE tid=9 UNION ALL SELECT 1111,2222,3333 FROM users	2009/10/12 13:57:16	2	獲取資料成功	L1 : 獲取資料成功 L2 : 獲取資料成功 L3 : query失敗 L4 : query失敗	安全等級維持 L2
tid : 9 UNION ALL SELECT 1111, 2222, 3333	SELECT * FROM thread WHERE tid=9 UNION ALL SELECT 1111, 2222, 3333	2009/10/12 13:56:46	2	獲取資料成功	L1 : 獲取資料成功 L2 : 獲取資料成功 L3 : query失敗 L4 : query失敗	安全等級維持 L2
tid : 9 UNION ALL SELECT NULL, NULL, NULL	SELECT * FROM thread WHERE tid=9 UNION ALL SELECT NULL, NULL, NULL	2009/10/12 13:56:38	2	獲取資料成功	L1 : 獲取資料成功 L2 : 獲取資料成功 L3 : query失敗 L4 : query失敗	安全等級維持 L2
tid : 9 UNION ALL SELECT NULL	SELECT * FROM thread WHERE tid=9 UNION ALL SELECT NULL	2009/10/12 13:56:25	2	query 失敗	L1 : query失敗 L2 : query失敗 L3 : query失敗 L4 : query失敗	安全等級維持 L2

圖 121、攻擊記錄

一切的行動都詳細的被系統記錄下來，包含陳上士輸入的字串、攻擊的時間、是否攻擊成功以及公及的層級，都一一詳細的被記錄下來，藉由此我們可以預防未知的攻擊，而在最後陳上士取得帳號密碼，事實上是我們在一開始就安排好的假的帳號密碼，只要有人用此帳號密碼登入立刻判斷為攻擊行為，並不會妨礙正常的登入行為。

同我們中央監控系統亦運作著，如下

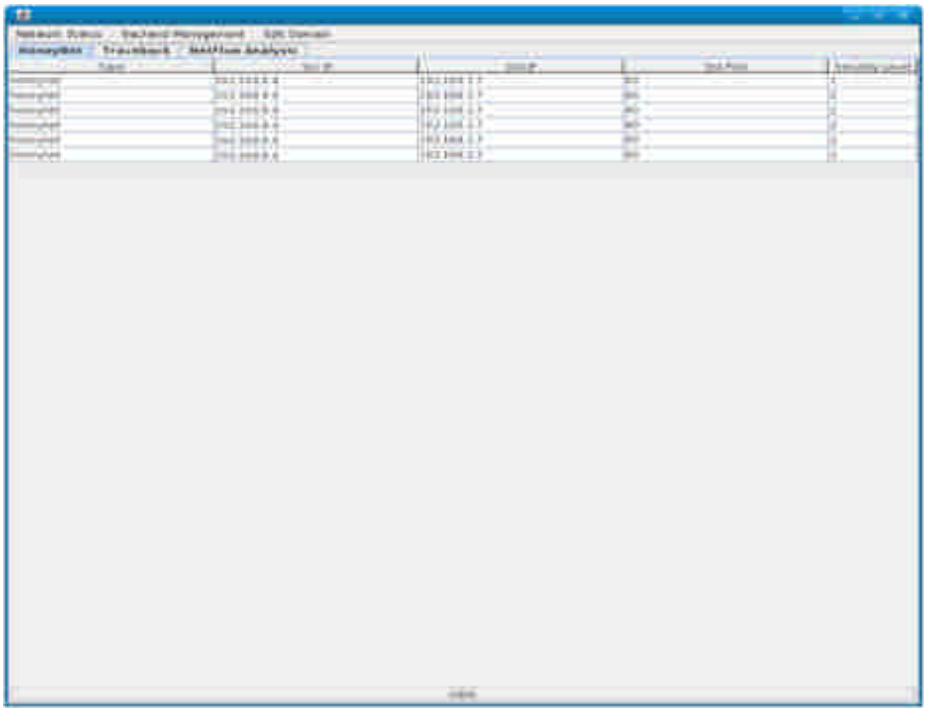


圖 122、中央監控系統

除了顯示第一次針對第一防護層級的攻擊，也顯示陳上士在第三次攻擊裡的記錄，此時系統管理員王上校提升了系統防護層級，讓攻擊者再度以為網站漏被修補，此網站持續更新。同時封包標記與追蹤系統系統在每次攻擊時也自動追蹤，但由於攻擊者 IP 位置沒有改變，攻擊路徑相同，如下：

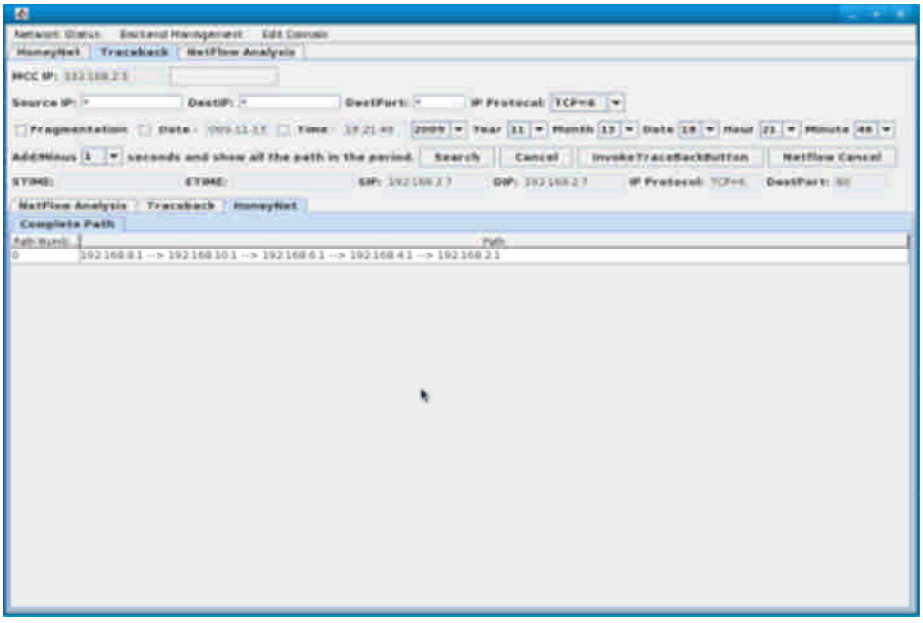


圖 123、攻擊路徑

(6) 陳上士第四次攻擊

陳上士在上次攻擊成功後，企圖再利用相同方法取得其他帳號資訊，但由於防護層級的提升，使得攻擊行為失敗，如下：

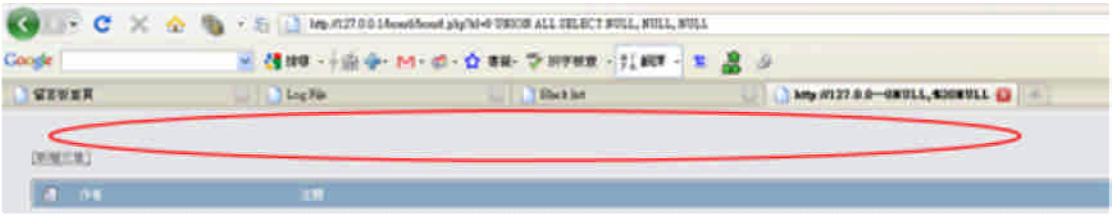


圖 124、嘗試失敗

無論陳上士如何錯誤嘗試，都不會出現錯誤訊息，使的他無法判斷是否攻擊成功，便是攻擊失敗。

(7) 陳上士第五次攻擊

陳上士進一步使用較為困難的攻擊方法，運用時差型攻擊，使瀏覽器的產生時間延遲判斷使否攻擊成功。

- 攻擊原理及目的：
  - 原理同上，也是藉由反覆詢問來竊取資料，但是在錯誤訊息被屏蔽的情況下，攻擊者可以讓網站傳輸產生延遲，來達到如同顯示錯誤訊息般的效果。
- 攻擊字串範例：

  UNION ALL SELECT BENCHMARK(800000,sha1(111111)),NULL,NULL

- 攻擊畫面

圖 125、攻擊畫面

藉由此陳上士又可判斷他下的攻擊指令使否有成功。

(8) 系統防禦

然而此次攻擊行為仍然被系統所掌握，被中央監控系統發現再度提升安全層級，封包標記系同樣在運作。

(9) 陳上士第六次攻擊

陳上士繼續使用時差型攻擊，但自從防護層級提升後，他便無法再以此攻擊方法進行判斷攻擊行為。攻擊失敗

(10) 陳上士第七次攻擊

自此系統遭受的攻擊都是常見的 SQL Injection，陳上士自己針對系統漏洞研發出一項新型的網路攻擊，本次攻擊為模擬新型網路攻擊，我們在網址列後面輸入 demo_test=1 代表新攻擊。

URL：http://140.115.53.35/board/board.php?id=9 UNION ALL SELECT BENCHMARK(100000,sha1(111111)),NULL,NULL FROM users& demo_test=1

(11) 系統防禦

本系統面臨到未知攻擊的考驗，我們的策略如下：

1 所有進出資料庫的關鍵資料，皆會被攔截並且紀錄。

2 檢查 輸入的帳號資料與所取得的帳號權限，是否相符

3 製造誘餌資料(例如：帳號密碼)，若以此資料登入，視為攻擊。

故本次攻擊行為並未會對我們的系統造成危險，而即使王尚是在未來使用更都的新型攻擊，事實上也只是替我們收集更多的攻擊訊息，使得我們系統防禦更為堅強，輔助傳統 IDS，補足其不足之處。

(12) 相關查詢

除了及時反映網路攻擊事件外，我們可以在事後管相關的網路黑名單，如下圖：

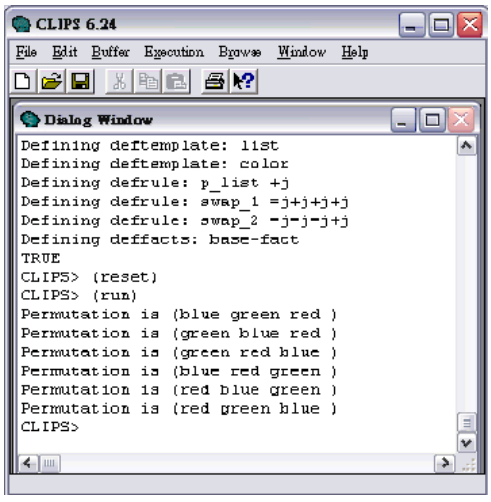
圖 126、管理黑名單

第八章、議題二－風險分析與威脅預警之核心技術與軟體單元研發

第一節、議題研究及說明

雖然目前市面上有眾多的入侵偵測系統，但這些偵測系統僅能針對部份可能的網路入侵方式進行監控與防堵。為了有效地分析所面臨地網路威脅，資安單位必須同時佈署多個不同的入侵偵測系統，並請資安專家對這些入侵偵測系統及相關伺服器進行監控。然而，由於各伺服器與入侵偵測系統產生的日誌為數眾多，加上資安專家可能會進行任務調動，因此，一個自動化網路威脅分析與預警系統，將能幫助資安專家更有效率地分析網路威脅，並舒緩資安單位在因資安專家任務調動所產生的空窗期所面臨的網路威脅。

為了預防網路各式攻擊行為以及保護區域網路內之安全，我們在本計劃裡將採用資安法則來實現區域網路內的預警，一般而言，資安法則的來源有二。一是由資安專家自行設立，由資安專長將其用來分析網路威脅的知識轉化成資安法則。目前的法則推論系統(rule-based inference system)大多缺乏便利的操作介面，如：CLIPS 必須以純文字定義法則的結構及推論流程，針對特定的問題進行客製化後才有辦法使用，CLIPS 必須以純文字畫面進行操作如下圖所示：



且在開發上必須撰寫大量描述語法來定義法則如下圖所示：

```
(deftemplate list (multislot item))
(deftemplate color (slot name) (slot seq))

(defrule p_list (list (item $?f ?s $?t))=>(printout t "Permutation is (" $?f " " ?s " " $?t ")" crlf))

(defrule swap_1 ?list<-(list (item $?f ?s $?t))(color (name ?f) (seq ?n1))(color (name ?s) (seq ?n2))
(color (name $?t) (seq ?n3))(test (> ?n2 ?n1))=>(assert (list (item ?s $?f $?t))))

(defrule swap_2 ?list<-(list (item $?f ?s $?t))(color (name ?f) (seq ?n1))(color (name ?s) (seq ?n2))
(color (name $?t) (seq ?n3))(test (> ?n3 ?n2))=>(assert (list (item $?f ?t ?s))))

(deffacts base-fact   (list (item red green blue))  (color (name red)(seq 1))  (color (name green)(seq 2))
 (color (name blue)(seq 3)))
```

圖 128、描述語法

在操作面及維護上均不便利。此外，由於資安專家對人工智慧不一定熟悉，因此我們將設計一個「資安法則編輯器」，藉由高擴充性的法則定義與編輯功能及操作便利的介面打破此項困境以協助資安專家將抽象地資安知識轉化成資安法則。根據以上的法則編輯與驗證工具建構法則後，如何透過法則推論引擎進行網路威脅分析及風險量化、威脅預警等模式，亦為本計畫之重要研究項目之一。我們將分析現有網路威脅分析、網路預警等之研究成果，研發法則推論基礎之網路威脅分析與預警模式。

此外，法則的獲取方式可透過系統自動學習而得。在近年來的研究結果中，學者們利用不同的人工智慧技術(如資料探勘、模糊理論、基因演算法等)，從既有的網路威脅資料中學習出資安法則。我們將根據以上的研究成果，進一步評估現有商業法則推論引擎是否能有效的達成網路威脅分析以及威脅預警模式等，進而採用一可行商用法則推論引擎進行學理與實務之驗證。

除了應用資安法則以外，我們針對如何增進判斷的準確率也做了相關的研究，我們在判斷時運用分群法以及失誤樹分析，分群法又稱資料分群（data clustering）或是分群演算法（clustering algorithms）是一種將資料分類成群的方法，其主要的目的乃在於找出資料中較相似的幾個群聚（clusters），在同一群聚的成員們擁有相似的屬性，或是更加相近的距離等。圖 138 即是將一平面座標點集分成四個群聚的結果。

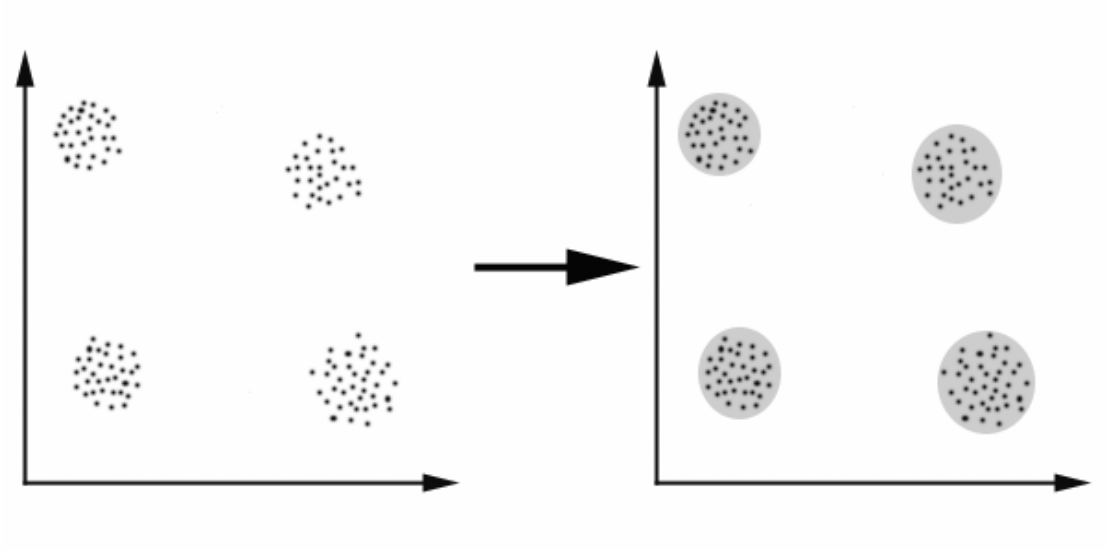


圖 129、Example of clustering

一般而言，分群法可以大致歸為兩大類：

1. 階層式分群法（hierarchical clustering）：

群數（number of clusters）可以由大變小，或是由小變大，來進群聚的合併或分裂，最後再選取最佳的群數。

2. 分割式分群法（partitional clustering）：

先指定群數後，再用一套疊代的數學運算法，找出最佳的分群方式以及相關的群中心。

所有的分群法都有相似的流程，大略可歸納為下列幾點：

e. 收集資料

f. 使用某種方法進行分群

g. 測試分群結果

檢測分群結果，如果未達預期效果，則回到步驟二，再一次進行分群，

失誤樹(fault tree)是 1962 年由貝爾實驗室(Bell Telephone Laboratories)發表，從系統的失效現象做出發，根據這些失效現象，分析失效發生的原因及造成系統失效的可能部位。而為了使失誤樹能順利運作，我們需要輸入精確的錯誤資料和適當的模型(model)，才能計算出失誤發生的風險。**錯誤! 找不到參照來源。**為一失誤樹的範例，用來表示 Top event 發生的風險，其中每一個圓形的節點都代表一個事件，而失誤樹則透過一些邏輯的組合來描述這些事件對 Top event 的影響。例如**錯誤! 找不到參照來源。**的失誤樹告訴我們：「當 X_2 和 X_3 同時發生或

X_1 發生時，top event 變會發生」。

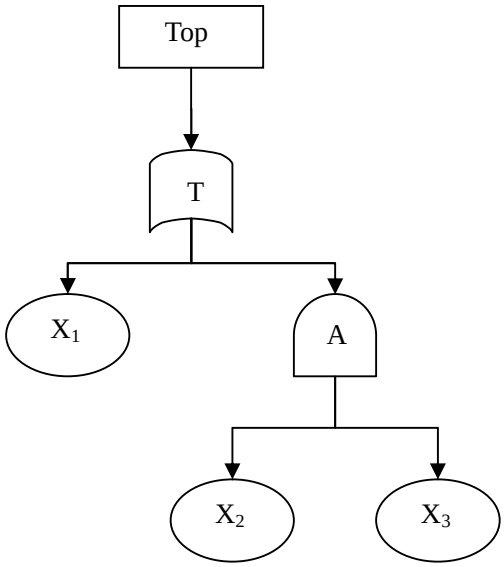


圖 130、Example of fault tree

我們也可以透過下面的式子來計算 top event 的風險值：

$T = X_1 \cup A$

$A = X_2 \cap X_3$

因此，我們可以得到 $T = X_1 \cup (X_2 \cap X_3)$，若以 P_{X_i} 來代表 X_i 發生的機率，則 $P_T = 1 - ((1 - P_{X_1})(1 - P_{X_2} P_{X_3})$，此即為 top event 可能發生的風險。

綜合以上研究，我們可以針對區域網路作更準確的預警。

第二節、議題實作

針對此議題我們防禦策略的示意圖如下：

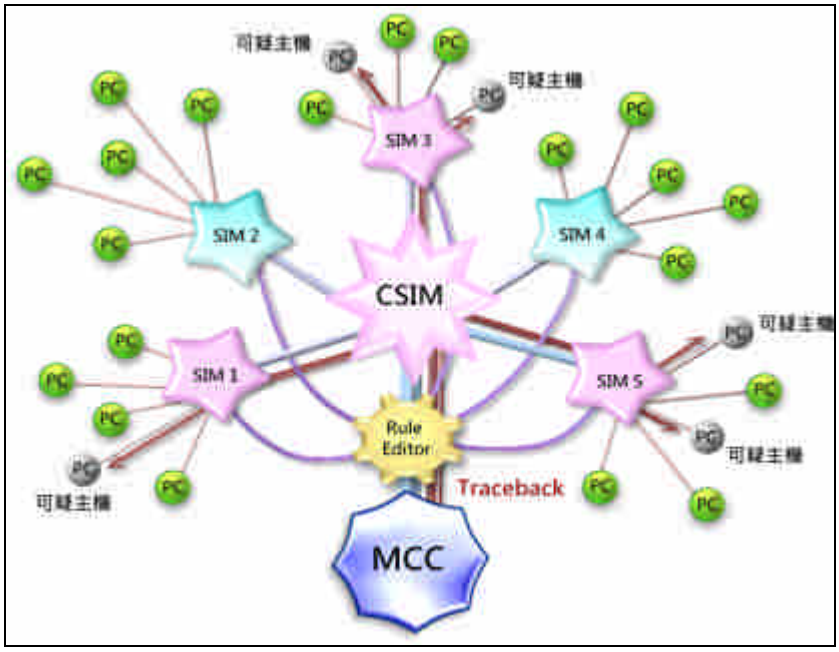


圖 131、防禦策略示意圖

議題二的防禦策略可分成 Offline 以及 Online 兩類。

- Offline：藉由平常收集的資料，建立出可疑的主機名單，提早防範攻擊。
- Online：利用平常建立的資料庫即時監控主機異常行為。在本次展示中，攻擊者以 NMAP 進行 IP 掃描，但被系統偵測找出攻擊者的攻擊路徑。
- Rule Editor: 法則編輯器，管理者可藉由此進行區域網路內資安法則的調整。
- CSIM: 中央網路風險分析預警主機，藉由此可對各個網路分險分析預警主機查詢。
- SIM: 網路風險分析預警主機，藉由此可對各網域進行網路風險分析預警資料的收集。
- Traceback: 當 MCC 下達指令時，進行攻擊者的位置回追。

根據此防禦策略。我們於計劃中實做了一套系統，如下：

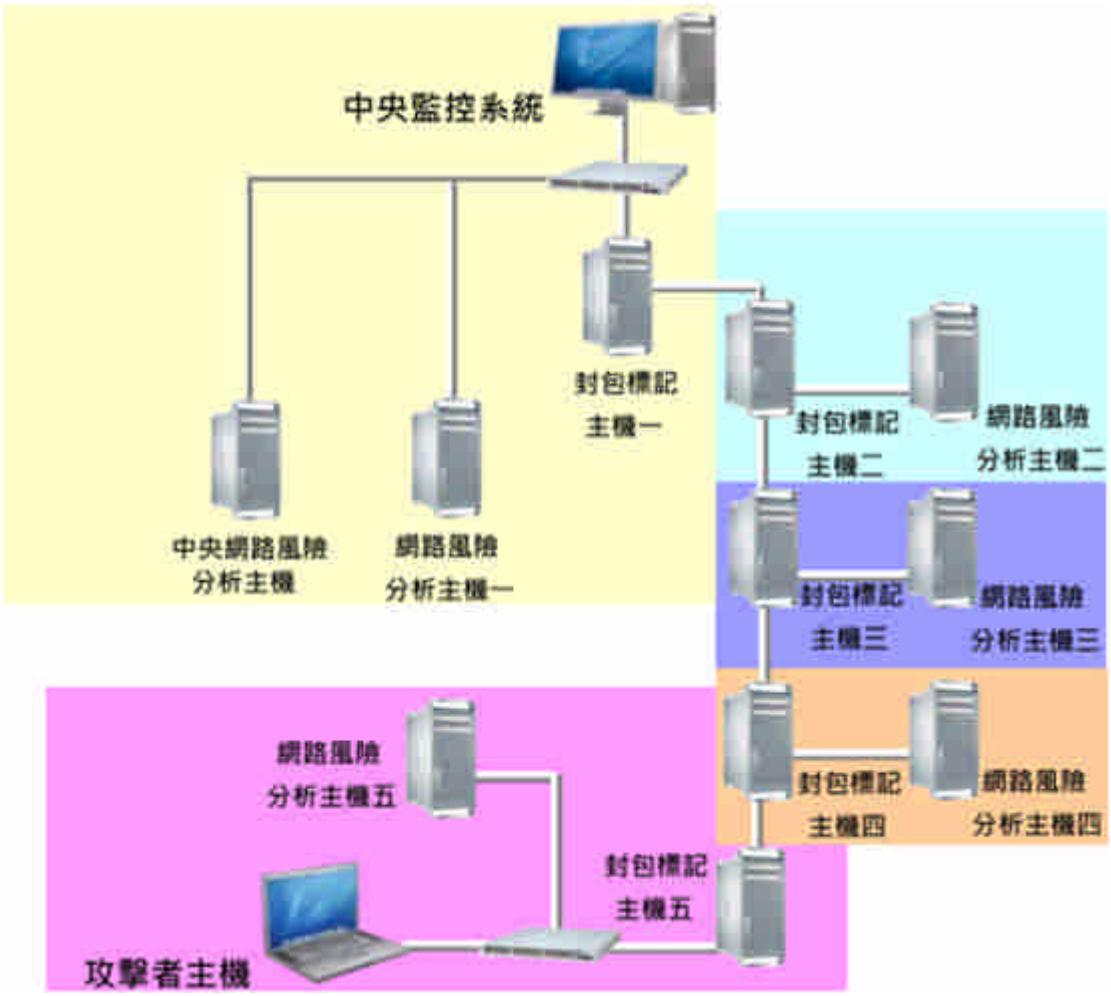


圖 132、系統架構圖

主要為整合網路風險分析及與警系統、封包標記與追蹤系統、中央監控系統而成，藉由此系統我們可以分析各網域內的可疑主機，假若網域內的主機有產生攻擊行為，攻擊其他網域內的主機，借由中央監控系統我們可以輕易的查覺，接著再藉由封包標記我們可以追蹤正確的攻擊路徑防止 IP 假冒行為。

第三節、實境模擬與成果展示

在本節我們將以實際的網路攻防驗證系統功能性，以下我們將詳細逐一介紹，首先介紹實境模擬一：

1. 實境模擬一：

李上兵(ip: 192.168.2.8)未更新主機裡的軟體，造成主機存在許多安全漏洞，我們欲藉由中央監控系統發現。

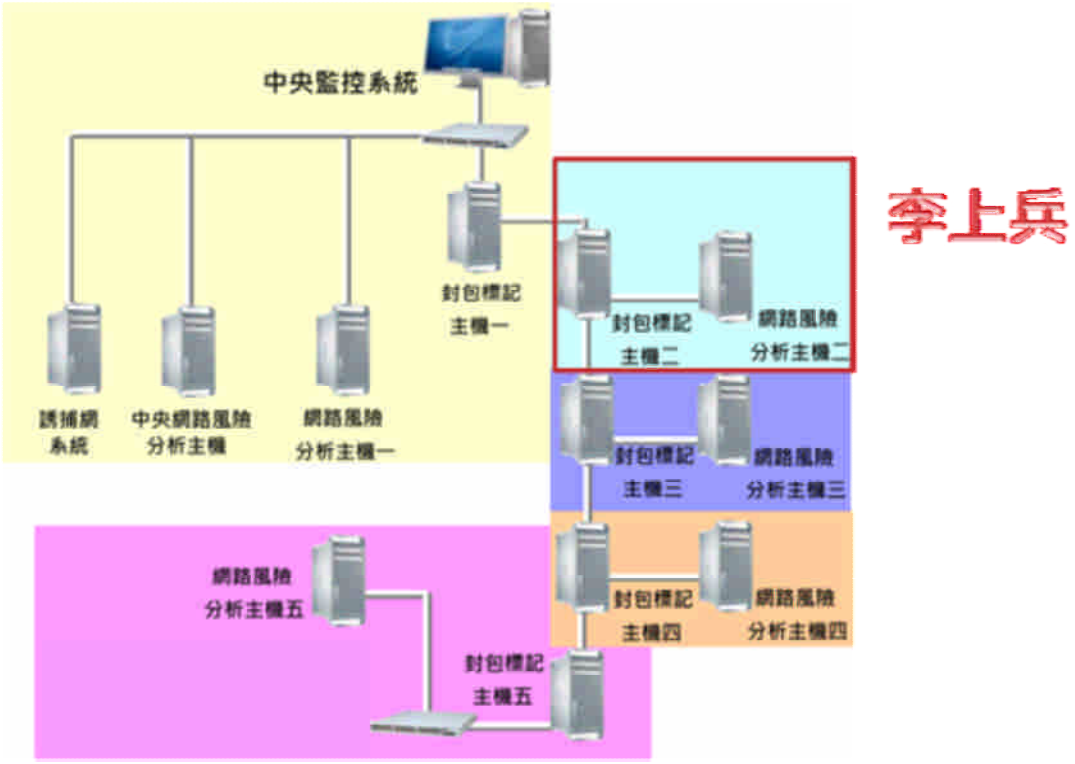


圖 133、實境模擬

(1) 中央監控系統

a. 中央監控系統管理員例行性的檢查是否有人違反資安規則造成網路危害，首先進入規則瀏覽以續點選查詢。

圖 134、Rule Editor

b. 中央系統管理員在檢查道更新頻率異常的那一項發現有可疑主機。

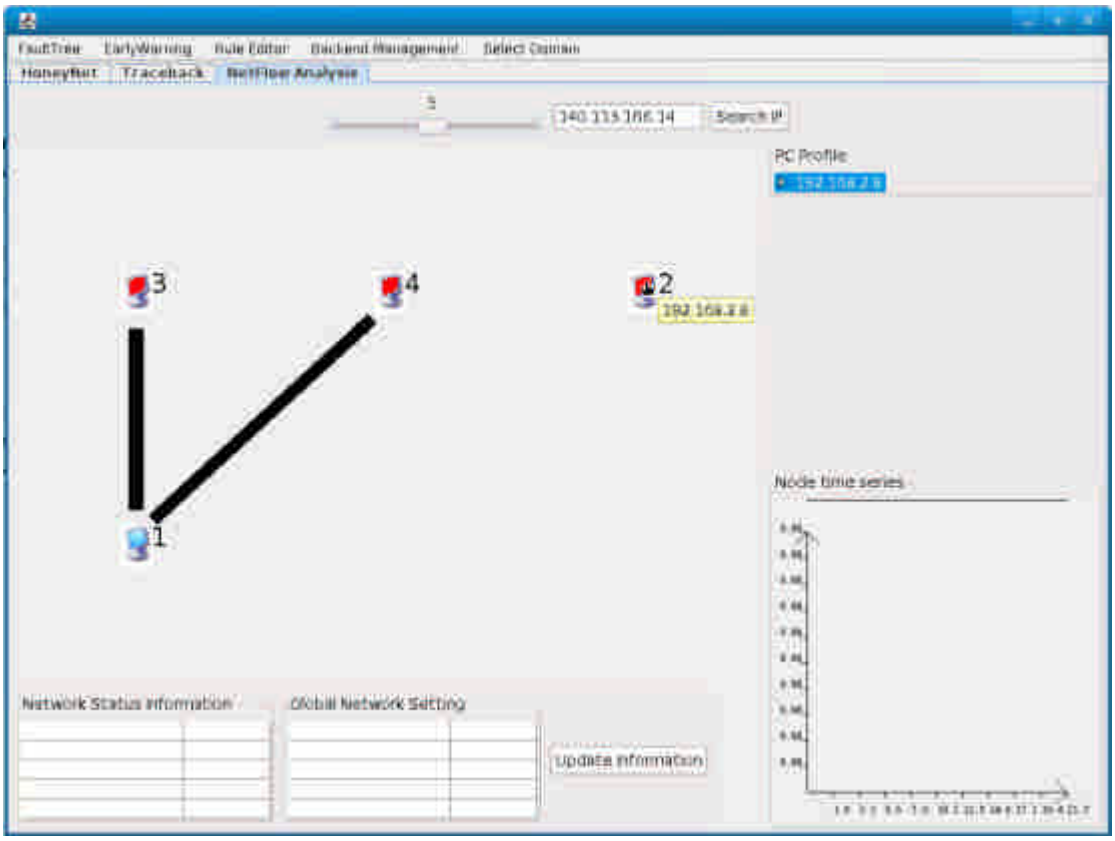


圖 135、可疑主機分析

除了 3，4 號主機有大量的異常更新行為，還發現 2 號主機完全沒有對外連線更新，將滑鼠指標移到主機上發現主機 IP 為 192.168.2.8 即為李上兵之主機，右

下角連線狀況亦為零。接著管理員查詢可疑主機的危害機率。

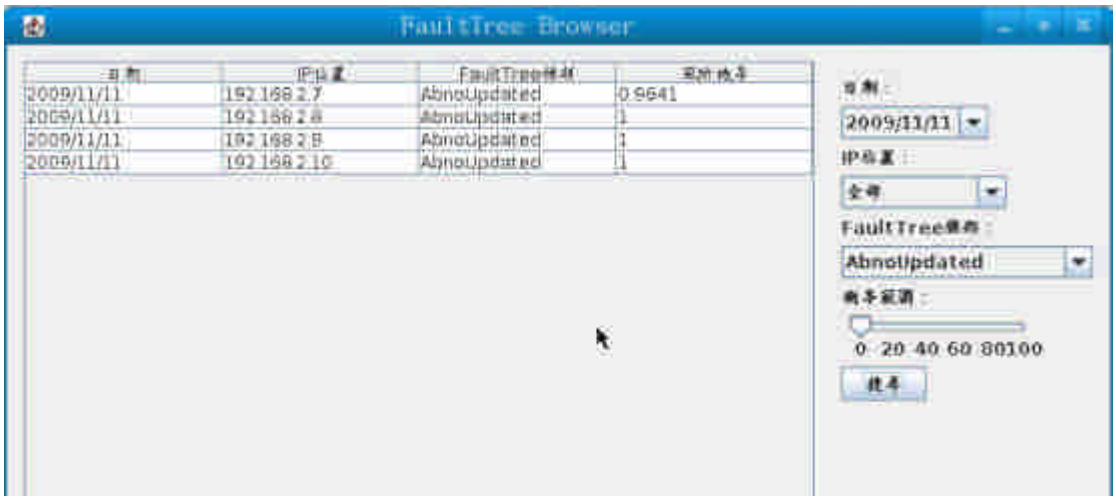


圖 136、攻擊記錄

發現該台主機異常機率高達 100%，至此，系統管理員可立即通知使用者進行系統的更新。

2. 實境模擬二：

陳上士使用 NMAP 工具掃描王上校的電腦主機，企圖入侵王上校的主機

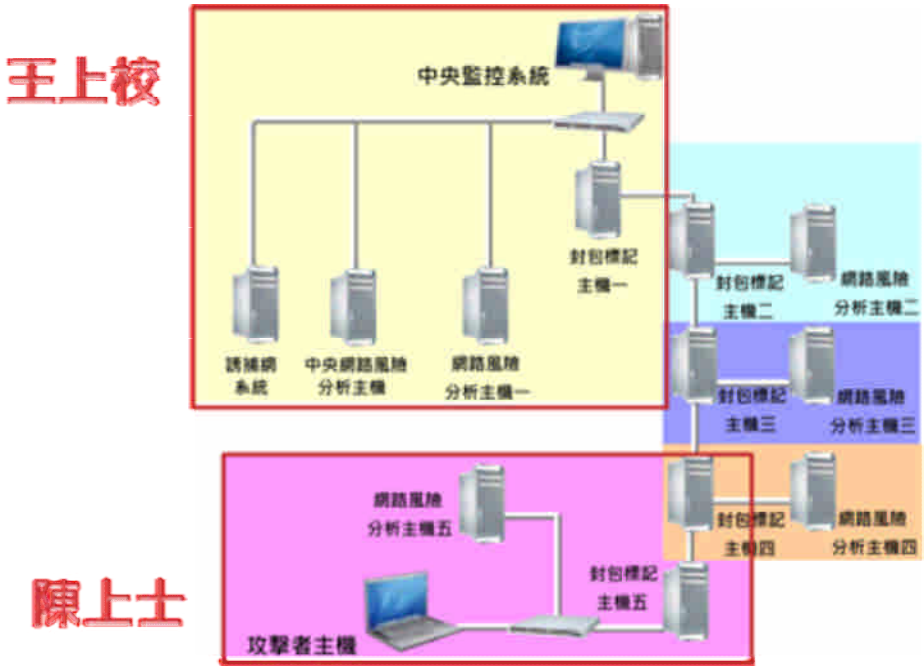


圖 137、實境模擬

(1) 陳上士攻擊：

陳上士(IP:192.168.2.3) 以偽造 IP 192.168.10.5 對王上校 (IP 192.168.8.3) 進行 Scan Port 攻擊.攻擊畫面如下：

圖 138、攻擊畫面

(2) 中央監控系統

中央系統管理員例行性檢查區域網路安全性

圖 139、檢查網路安全

發現在 Scan Port 的資安法則裡於 2009-12-02 有主機遭受攻擊，於是管理者趕緊查看可疑主機。

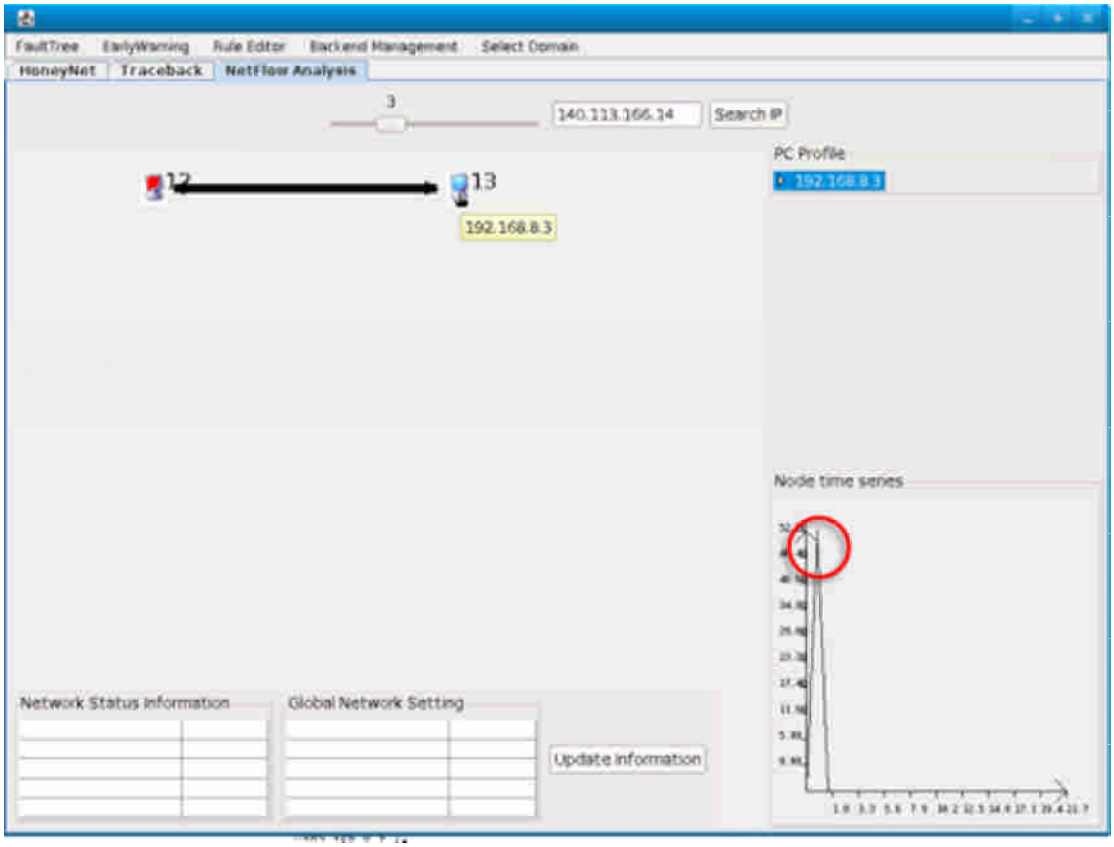


圖 140、可疑主機分析

發現有可疑主機 192.168.10.5(假冒 IP)對 192.168.8.3 進行攻擊，並且我們於右下角的連線狀狀圖可發現，於某一時段有極高之流量，故可判斷為惡意的攻擊行為。我們點選可疑主機圖示，切換到封包標記與追蹤系統管理介面進行攻擊路徑追蹤，有大量的路徑追蹤資料如下。

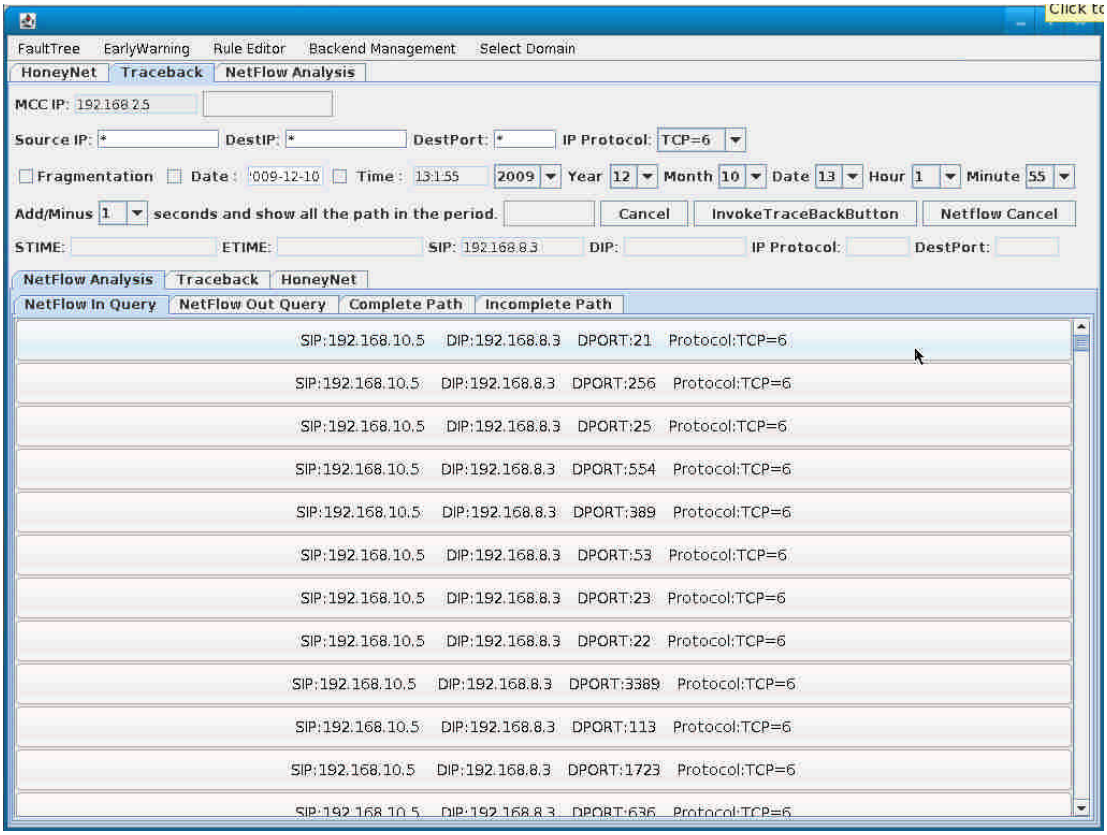


圖 141、可疑攻擊路徑

我們找尋其中一項點選並追蹤，發現雖然顯示 192.168.10.5 但攻擊路徑如下圖為：

192.168.2.1->192.168.4.1->192.168.6.1->192.168.10.1->192.168.8.1。

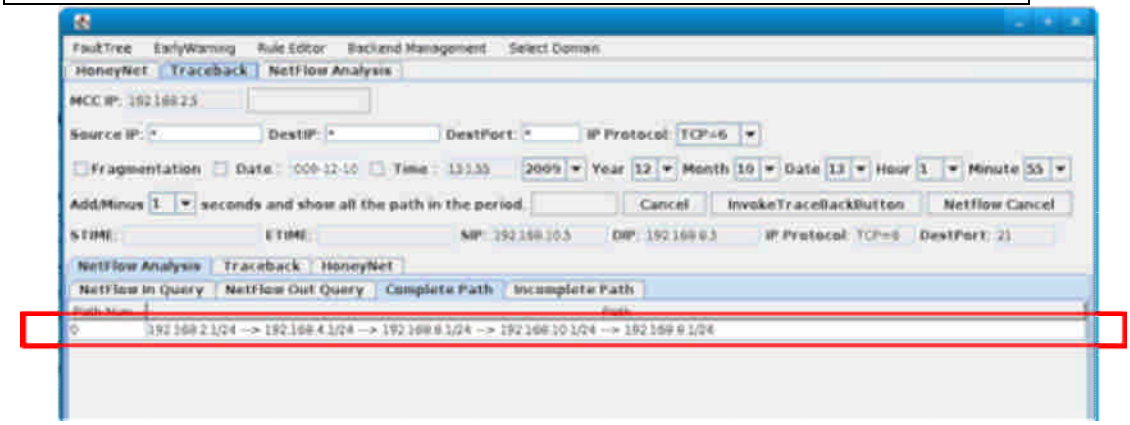


圖 142、路徑追蹤

我們可判斷此為一個 IP 假冒攻擊，若為 192.168.10.5 發起的攻擊，攻擊路徑應該是

192.168.10.1->192.168.8.1。

故我們可立即聯絡該使用者並且做出警告，使得區域網路內的安全得以確保。

第九章、執行成果

本計畫於今年度實現兩大議題目標：(一)在第一年度設計之誘捕網雛形雛形進行發展，加入防護等級提升機制、增強誘捕能力與封包標記與追蹤系統進行整合，以防止使用者偽造 IP 位置。(二)在第一年度設計之網路風險分析及預警系統進行發展，除了提升系統效能之外，並滿足管理者客製化的需求。此外，整合封包標記與追蹤系統進行整合，防止偽造 IP 攻擊。並整合至中央監控系統，以統整各系統資訊，及時因應網路攻擊。我們將於以下章節分別介紹今年度計劃執行成果列表、計劃執行成果比較、98 年國防論文研討會以及計劃的成果發表會。

第一節、成果統計及列表

項目		件數
創新轉移技術		14 件
系統軟體		19 件
技術文件		22 件
Publication	研討會	15 件
	期刊	4 件
	投稿中	1 件

表格 36、成果統計

1. 創新轉移技術(14 件)

	技術名稱
WINDOWS 系統誘捕技術與隱形記錄、WINDOWS 網頁伺服器之誘捕風險與安全控管技術	(1) SQL injection 誘捕主機設計：首頁畫面設置成「中科院」網站，連結均以「內嵌框架」形式開啟，以隱藏此處仿製之 URL。可誘捕 SQL injection attack 和 Script injsection attack。 (2) Buffer Overflow 攻擊偵測技術：當入侵者利用 Buffer Overflow 攻擊導致系統當機時，由於 system log files 內會有明顯的跡象出現，亦即會有與 Buffer Overflow 攻擊有關，且屬於 Buffer Overflow 攻擊獨有的 log records 產生。

	(3) Port Scan 攻擊偵測技術：port scan 攻擊在 system log files 的 argus 資料表中就有這種攻擊的特有 data records 產生，從 argus 資料表就可以看出有大量的連線數對同一 ip、連續 ports 發送封包，因此可以藉由比對此一特有 data records，來直接判斷是否發生了 port scan 攻擊。
網路封包標記與追蹤技術	(4) 核心免重編技術：本技術需修改核心，主要是為了新增自訂的變數，並且讓使用者自由修改變數內容，因此需要在核心新增變數與讀取的函式。要加入的變數總共有五個，第一個是 MYIID，每一台 MPC 所具有的編號，第二個是 MYRIP，儲存網域或網址的內容，具有多個，第三個是 MYSUB，與 MYRIP 互相配對的 netmask，也是具有多個，第四個是 MYCNT，用來表示目前所具有的 MYRIP 的數量，第五個是 SETPN，用來設定是否進行標記。增加這五個變數，可以使標記設定能夠以網域進行判斷是否進行標記，以及修改 IID 時，不需要重新編譯核心，達到有彈性的效果。 (5) Bridge module 技術： 在 MPC 當作 bridge 傳送的時候，是經由模組內的 bridge module 所運作的，因此我們在修改傳送封包內容時，藉由所經過的模組，改其傳送封包的流程，讓它在經過 MPC 後，能夠增加封包標記內容。在 bridge 模組中，封包進行傳遞時，會經過 br_forward.c 檔案內的__br_forward 這個函式，利用它所經過的途徑，將所要標記的程式碼寫入這個函式內，就可以達到封包修改的目標。 使用 system call 語言實做呼叫修改核心變數技術：為了讓使用者在 user space 能夠進行資料的讀取與修改，因此就需要編寫 system call 的函式，讓它能夠由這些函式修改上面五個變數內容。 (6) 封包監聽與記錄之 Buffer 技術：在監聽的部份加入 Buffer，使標記的資料能夠在短時間保留在 Buffer 內，直到 Buffer

	滿載或者一段時間後，再寫入資料庫，以減少短時間內相同資料的產生。 (7) 資料庫資料自動刪除程式技術：標記資料的記錄，空間容量一段時間後，就有可能造成空間不足，因此我們設定為每天都會定時將七天以前的資料，進行刪除的動作，以防止空間不足。 (8) 網頁設定資料庫技術： MySQL 資料庫設定主要是在於權限以及遠端存取的控制，我們使用 phpMyadmin 去進行網頁式設定。
網路威脅風險分析與威脅預警之核心技術	(9) 履歷建立技術：所建立的履歷為一階層式的結構，初始的網路連接履歷為空，接著從防火牆日誌(firewall log)中取得每個主機的 IP 位置和網際網路服務後**錯誤! 找不到參照來源。**，並在 others 紀錄此 IP 所有連線數目和連線型態(如 Http、Ftp)。 (10) 異常行為法則分析之技術：未更新之主機(no update PCs)、未回報之主機(no report PCs)、爬網站(crawl web site)之主機、含有不安全服務(unsafe services)組合之主機、未列管伺服器以及使用未申請 Port 的主機，觀察這些異常行為發生時，反應在防火牆日誌(firewall log)上的連結情形與平時有何差異，再利用攻擊樹(attack tree)來計算威脅發生的機率，進一步將威脅分險量化。當主機遭到惡意攻擊時，它的連結情形會與其他主機有很大的差異，藉由視覺化連結展示器，這些異常的主機將很輕易的被辨認出來。再利用比對目前連結情形與履歷的差異，可用來驗証我們的觀察是否正確，便可以找出異常的主機及惡意攻擊 (11) 資安法則編輯技術：系統偵測出的異常狀況可能是惡意攻擊的行為亦可能是誤判。為了提升系統的準確度及實用性，建立一個容易使用(easy to use)的規則編輯器，讓資安專家可以自行定義異常規則、資安規定及建立網址的黑名單、白名

	單，輔助系統辨識異常狀況。
視覺化系統整合人機介面技術	(12) WINDOWS 系統誘捕技術與隱形記錄、WINDOWS 網頁伺服器之誘捕風險與安全控管技術人機介面 (13) 網路封包標記與追蹤技術人機介面 (14) 網路威脅風險分析與威脅預警之核心技術人機介面

表格 37、創新轉移技術

2. 系統軟體

WINDOWS 系統誘捕技術與隱形記錄、WINDOWS 網頁伺服器之誘捕風險與安全控管系統	(1) Honeywall 軟體：資料的搜集與儲存。 (2) Honeypot 軟體：資料的誘捕與捕獲。 (3) 資料分析軟體：分析 Honeywall 和 Honeypot 之資料。 (4) 每單元資料交換介面軟體：以 TCP 之方式傳輸設計
網路封包標記與追蹤系統	(5) 封包標記軟體：將傳送進來的封包進行封包標記。 (6) 封包轉送軟體：將標記過後的封包進行轉送。 (7) 記錄封包標記軟體：將傳送進來的封包標記內容讀取出來，記錄到資料庫。 (8) 資料庫存取軟體：資料庫存取只能由本地端進行存取，外部查詢需透過 Socket。 (9) Java 程式執行軟體：路徑追蹤 Java GUI 介面的執行。
網路威脅風險分析與威脅預警之核心系統	(10) 資料庫存取軟體：提供資料庫記錄(1)原始防火牆日誌(firewall log)、(2)資安規則、(3)各主機之履歷結構(Profile)、(4)MCC 使用記錄及(5)HoneyNet 使用記錄。 (11) 履歷(profile)系統軟體：將資料庫中之原始 firewall log 依不同的主機及連線建立各主機之 profile，並存入資料庫中，以供查詢。 (12) 異常行為分析軟體：以 MCC 設定之資安規則對資料庫中之各主機 profile 進行篩選，分別針對六項規定找出異常之主

	機。 (13) MCC 控制介面軟體：利用 Java 通訊平台，接受 MCC 下達之指令。
視覺化系統整合人機介面系統	(14) 中控路徑追蹤軟體：路徑追蹤 Java GUI 介面的執行。 (15) 查詢路徑追蹤軟體：中控中心透過 Java GUI 介面來查詢可疑 IP。 (16) HoneyNet 回報軟體：將有異常的行為進行誘捕並傳送至中控中心。 (17) Traceback 回報軟體：將路徑封包傳回中控中心。 (18) NetFlow Analysis 回報軟體：將六項資安規則裡面的資料傳送至中控中心顯示。 (19) 資料庫存取軟體：資料庫存取只能由本地端進行存取，外部查詢需透過 Socket。

表格 38、系統軟體

3. 技術文件(22 件)

WINDOWS 系統誘捕技術與隱形記錄、WINDOWS 網頁伺服器之誘捕風險與安全控管技術	(1) 《HoneyNet》在 VM 環境中建立 Honeynet (2) 《HoneyNet》安裝 Honeynet 於 VM 環境 (3) 期中報告 (4) 期末報告
網路封包標記與追蹤技術	(5) linux 系統核心編程 (6) deb 套件安裝 (7) 資料庫的建立 (8) 期中報告 (9) 期末報告 (10) 系統的使用手冊 (11) 系統的安裝手冊
網路威脅風險	(12) 應用資料探勘於網路異常攻擊行為之研究報告

分析與威脅預警之核心技術	(13) 商用推論引擎評估報告 (14) 威脅預警技術 (15) 網路攻擊分析及偵測經驗法則設計 (16) 系統介面規格 (17) 系統架構規格文件 (18) 軟體設計規格書 (19) 風險分析 (20) 資料庫說明文件 (21) 程式清單 (22) 測試報告

表格 39、技術文件

4. Publication(研討會、期刊、投稿中)

研討會：

[1] Yu-Lung Huang, Chih-Ya Shen, Shiuhpyng Shieh, Hung-jui Wang, Cheng-Chun Lin, “Provable Secure AKA Scheme with Reliable Key Delegation in UMTS,” IEEE Conference on Secure Software Integration and Reliability Improvement, 2009

[2] S.-C. Chiu, M.-K. Shan and J.-L. Huang, ``Automatic System for the Arrangement of Piano Reductions," Proceedings of International Workshop on Advances in Music Information Research (AdMIRe-09), December 14-16, 2009.

[3] S.-Y. Chiu, S.-C. Chiu and J.-L. Huang, ``On Mining Repeating Pattern with Gap Constraint," Proceedings of International Symposium on Advances of High Performance Computing and Networking (AHPCN-09), June 25-27, 2009.

[4] S.-C. Chiu, M.-K. Shan, J.-L. Huang and H.-F. Li, ``Mining Polyphonic Repeating Patterns from Music Data Using Bit-String Based Approaches," Proceedings of IEEE International Conference on Multimedia & Expo (ICME-2009), June 28-July 3, 2009.

[5] C.-C. Hung, C.-W. Chang and W.-C. Peng, ``Mining Trajectory Profiles for Discovering User Communities," Proceeding of the 1st Workshop on Location-Based Social Networks (In conjunction with ACM GIS), Seattle, USA, Nov. 3-6, 2009.

[6] C.-C. Hung and W.-C. Peng, ``Clustering Object Moving Patterns for Prediction-based Object Tracking Sensor Networks," Proceedings of the ACM 18th International Conference on Information and Knowledge Management (CIKM), Hong Kong, Nov. 2-6, 2009.

[7] L.-Y. Wei, W.-C. Peng, C.-S. Lin and C.-H. Jung, ``Exploring Spatio-Temporal Features for Traffic Estimation on Road Networks," Proceedings of the11th International Symposium on Spatial and Temporal Databases (SSTD 2009), Aalborg, Denmark, July 8-10, 2009.

[8] Paul Y.-S. Chiang and W.-C. Peng, ``Slab Routing: Adapting Two-Dimensional Geographic Routing to Three-Dimensions," Proceedings of the 6th IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2009), Rome, Italy June 22-26, 2009. (Acceptance rate ~18.8% 81 out of 431 submissions)

[9] Y.-M. Chang, L.-Y. Wei, C.-S. Lin, C.-H. Jung, W.-C. Peng, I.-H. Chen, ``Exploring GPS Data for Traffic Status Estimation," Proceedings of the 10th International Conference on Mobile Data Management (Demo paper), Taipei, Taiwan, May 18-20, 2009.

[10] L.-Y. Wei and W.-C. Peng, ``Clustering Data Streams in Optimization and Geography Domains," Proceedingds of the 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2009), Bangkok, Thailand, April 27-30, 2009.

[11] Yi-Chi Wu, Huei-Ru Tseng, Wuu Yang, Rong-Hong Jan, DDoS Detection and Traceback with Decision Tree and Grey Relational Analysis , The 3rd International Conference on Multimedia and Ubiquitous Engineering (MUE 2009, June 4-6, 2009, Qingdao, China), 2009.

[12] Huei-Ru Tseng, Rong-Hong Jan; Wuu Yang, A Chaotic Maps-based Key Agreement Protocol that Preserves User Anonymity, IEEE ICC'2009 International Conference on Communications (Ad hoc and Sensor Networking Symposium, June 14 - 18, Dresden, Germany), 2009.

[13] Tzu-Han Hung, Jiunn-Yeu Chen, and Wuu Yang, Wei Chung Hsu, Program Type Recognition for Compiler Optimization, 7th Workshop on Optimizations for DSP and Embedded Systems (ODES-7, March 22nd, 2009, Seattle, Washington), 2009.

[14] Hung Meng Fen, Li Xuan Lang, Jin Cheng Lin, Chen Jian Hua, Hsi-Lu Chao,

Ming-Hann Huang, and Ho Ian, “The Research and Development of Honeynet with IP Traceback,” submitted to *The Conference on National Defense Science and Technology*.

[15] Weng Xing Guo, Li Qiao, Wen-Chih Peng, Jiun-Long Huang, Zhung-Xun Liao, Kuby Huang, Tracy Lin, and TsungWei Wang, “A Network Monitoring System for Abnormal Network Behaviors,” submitted to *The Conference on National Defense Science and Technology.*

期刊：

[1] Shih-I Huang, Shiuhpyng Shieh, “Secure Encrypted-Data Aggregation for Wireless Sensor Networks,” accepted for publication, Wireless Networks.

[2] Shih-I Huang, Shiuhpyng Shieh, “Secret Search Mechanism for Wireless Sensor Networks with Passive RFIDs,” accepted for publication, International Journal of Security and Networks

[3] Hsi-Lu Chao and Chen-Lung Chang, “A Fault-tolerant Routing Protocol in Wireless Sensor Networks,” accepted by International Journal of Sensor Networks.

[4] J.-L. Huang, S.-C. Chiu and X.-M. Huang, ``GPE: A Grid-based Population Estimation Algorithm for Resource Inventory Applications over Sensor Networks," Journal of Information Science and Engineering, Vol. 25, No. 1, January 2009.

投稿中：

[1] Chiachen Wu, Shiuhpyng Shieh, John Kubiatowicz, Ellen Huang, “Anonymous Fault-Tolerant Routing for Overlay Networks,” submitted to ASIACCS 2010.

第二節、成果比較

1. 系統整合成果比較

項目	第一年	第二年
中央監控系統	1. 無帳號密碼管理 2. 雛形介面	1. 有帳號密碼管理及資料庫建立 2. 完整介面
中央監控系統與 路徑回追系統	無整合及控制	可利用中央監控系統查詢攻擊路徑
中央監控系統與 網路風險分析預警系統	1. 無整合及控制 2. 無加密傳輸	1. 可利用中央監控系統控制及查詢。 2. SSL 加密傳輸
中央監控系統與 誘捕網系統	1. 無整合及控制 2. 無加密傳輸	1. 可利用中央監控系統控制及查詢。 2. SSL 加密傳輸

表格 40、系統整合成果比較

2. 誘捕網系統成果比較

項目	第一年	第二年
建立方式	傳統 Web Honeypot 架設	架設於 VM 增加可攜性
防護層級	無設定防護層級	利用防護層級設定增加攻擊難度
系統設計	可抓捕基本 SQL injection	新增數種 SQL injection 捕抓
與路徑回追系統	無整合	資料傳輸連線建立

表格 41、誘捕網系統成果比較

3. 封包標記與追蹤系統成果比較

項目	第一年	第二年
封包標記	使用 Option 欄位	使用 Identification 欄位避免封包在傳輸過程中被拋棄
資料庫	利用 buffer 減低寫入資料量	加上三向交握再進一步減低寫入資料量
路徑回追	人工處理	自動重建
與誘捕網系統	無整合	資料傳輸連線建立
與網路風險分析預警系統	無整合	資料傳輸連線建立

表格 42、封包標記與追蹤系統成果比較

4. 網路風險分析及預警系統成果比較

	第一年	第二年
架構	一台 SIM	2-tier 架構： 一台 CSIM、五台 SIM。
資料庫	一般資料庫	運用資料倉儲加快讀取速度。
法則編輯	無法編輯	可利用法則編輯器編輯
失誤樹分析	無失誤樹分析	不僅利用失誤樹分析，且可以由使用者編輯失誤樹架構。
異常攻擊偵測	可偵測但無法預警	可預警及偵測
與路經回追系統	無整合	資料傳輸連線建立

表格 43、網路風險分析及預警系統成果比較

第三節、研討會及成果展示

1. 國防科技學術合作計畫成果發表會

時間：中華民國 98 年 11 月 26 日(星期四)

地點：龍潭宏碁渴望園區「渴望學習中心」

圖 143、國防科技學術合作計畫成果發表會

圖 144、國防科技學術合作計畫成果發表會

2. 98 中山科學研究院學合案期末成果展示

時間：中華民國 98 年 12 月 16 日(星期三)

地點：龍園研究園區

圖 145、98 中山科學研究院學合案期末成果展示

圖 146、98 中山科學研究院學合案期末成果展示

第十章、結論與建議

此研究計畫為期兩年，本報告為第二年的成果報告，本年度計劃執行成果豐富，投入研究人員二十三人，開發系統軟體共計十九件，產出創新技術共十四件，技術文件二十二件，可供轉移技術共計十四件，論文共計二十件（其中一件尚在投稿中）。除上述成果產出外，我們進行實務設計以及實現，以實際網路攻防進行功能測試，確保系統能偵測與防禦網路攻擊行為。

本計劃的研究項目主要為針對「Windows 誘捕網情蒐之核心技術與回追技術軟體發展」與「風險分析與威脅預警之核心技術與軟體單元研發」兩大議題。在「Windows 誘捕網情蒐之核心技術與回追技術軟體發展」議題中，我們發現防禦機制雖然對已知的攻擊有良好的防禦表現。但面對新型攻擊行為時，防禦能力卻是十分有限。為補強傳統防禦機制不足之處，我們針對第一年開發之誘捕網雛形系統進行發展，加入防護等級提升機制，利用不同的防護等級給予攻擊者不同阻力，使攻擊者錯以為網頁漏洞遭到修補，使用進階方式攻擊網站。同時，我們將詳細記錄攻擊者在誘捕網裡的活動，藉此發現是否有新型網路攻擊方式。本系統將與傳統防禦機制相輔，進一步防止類似攻擊行為發生。此外，我們結合了封包標記與追蹤系統進行攻擊者路徑的追蹤，每當誘捕網系統偵測到攻擊者活動時，便立即通知封包標記與追蹤系統進行追蹤，藉此可發現攻擊者真實位置，避免攻擊者偽造 IP。在期末成果展示時，我們亦驗證其功能性，確保可應用在真實的網路中，未來將可運用在中科院裡容易遭受攻擊的網站進行網路防禦。

本計劃發展之誘捕網系統主要為針對網路應用程式，偵測其新型攻擊行為。然而，正由於誘捕網可收集各式訊息的特性，使得許多相關機都開始架設誘捕網系統，如學術機構運用誘捕網系統進行網路蠕蟲活動的研究、政府機關則利用誘捕網追查攻擊者目的及位置、防毒軟體公司架設誘捕網搜集惡意軟體及病毒特徵碼、ISP 公司利用誘捕網偵測並防止殭屍網路存在等等相關運用。由於誘捕網系統可運用的範圍十分廣大，我們認為誘捕網技術應當受到重視，尤其在面臨龐大網路資訊時，我們需要利用一套系統來記錄分析未知的訊息，進一步判斷是有網路威脅的存在，誘捕網系統在此過程中將扮演很重要的角色。我們期望在未來可持續加強此技術，並且針對各式用途進行開發，如此，將有助台灣學界以及業界相關領域的發展。

在「風險分析與威脅預警之核心技術與軟體單元研發」議題中，我們針對許多機構普遍會遇到的網路安全問題進行研究，由於網路管理人員需要面臨龐大的網路封包，往往無法逐一做判斷並提前預警，導致入侵事件的產生。因此，我們針對第一年就已開發之雛形系統進行發展，我們將系統架構擴大城 2-tier 架構以方便管理並減少網路頻寬的使用，並利用資料倉儲技術減少 20~40%的資料庫查詢時間。此外，我們亦加入了失誤樹風險量化分析技術，方便管理者判斷可疑主機。為提供客製化的系統，我們在系統中加入資安法則編輯器以及失誤樹編輯器，管理者可針對區域網路內之特性，設定不同的可疑主機判斷標準，增加可疑主機的偵測率。本系統亦與封包標記追蹤系統進行整合，管理者可透過管理介面查詢區域網路內封包的流向。如此可避免 IP 的偽造攻擊，確保可疑主機的網路位址無所遁形。本系統於今年度進行全方面的性能提升，並於期末成果展示逐一驗證其功能性，讓本系統不再只是學術研究的議題，而是可以正式運作於實際環境的監測系統。

本計畫使用網路風險分析與預警系統進行區域網路內的網路防禦工作，幫助管理者掌控區域網路內的資安事件，減少網路威脅的存在。本計劃應用之相關技術早已受到重視，不僅在學界熱烈討論，有許多公司相繼推出相關的資訊安全管理軟體，如 Cisco、Prism Microsystem、TriGeo 等國際公司。因此，我們認為此項技術應持續發展，尤其在面臨越來越複雜的資安問題還有管理上的不易。若持續加強此項技術，開發出更有效之資訊安全管理系統，可以協助國內產業與學界加強資訊安全度。本計劃除了針對兩大議題進行系統的開發，並且設計一中央監控系統使得誘捕網系統、封包標記與追蹤系統以及網路風險分析與預警系統可以完善的溝通，亦可利用監控一區域網域以達到妥善的網路管理。此外，為增強系統整體的安全性，彼此之間傳輸皆以 SSL 加密傳輸，防止區域網域內的攻擊。

本計畫投入大量研究人員，並每月定期與中科院的研究團隊進行一至二次討論會議，逐步達成所有預期目標及工作項目，將各系統設計開發成雛形系統，符合中科院期望，滿足實際需求。並將其系統理論及貢獻撰寫成論文，於國防論文研討會進行論文的發表。為了實際驗證整體系統功能性與實用性，本計畫於 98 年 12 月 16 日在中科院龍園研究園區進行成果展示。最後，我們將相關技術及系統全數轉移至中科院，並於小型區域網路內試運成功，未來可望於中科院正式上線運行。

參考文獻

[1] CVE List, http://cve.mitre.org/cve/index.html

[2] Blue Coat, http://www.bluecoat.com/

[3] Foundry, http://www.foundrynet.com/

[4] Citrix, http://www.citrix.com/lang/English/home.asp

[5] OWASP, http://www.owasp.org/

[6] C. Kruegel and G. Vigna, “Anomaly Detection of Web-based Attacks,”in ACM conference on Computer and Communications Security (CCS), 2003.

[7] X. Wang, J. Zhou, S. Yu and L. Cai, “Data Mining Methods for Anomaly Detection of HTTP Request Exploitations,”in Fussy System and Knowledge discovery (FSKD), 2005.

[8] K.L. Ingham and H. Inoue, “Comparing Anomaly Detection Techniques for HTTP,”in International Symposium on Recent Advances in Intrusion Detection (RAID), 2007.

[9] K. Golnabi, R.K. Min, L. Khan and E. Al-Shaer,“Analysis of Firewall Policy Rules Using Data Mining Techniques,” in IEEE/IFIP Network Operations and Management Symposium, 2006.

[10] Aleksandar Lazarevic, Levent Ertoz, Vipin Kumar, Aysel Ozgur and Jaideep Srivastava, “A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection,” in SDM Conference, 2003.

[11] D.E. Denning, “An Intrusion Detection Model,” in IEEE Transactions on Software Engineering, SE-13:222-232, 1987.

[12] T. Lane, C. E. Brodley, “Sequence Matching and Learning in Anomaly Detection for Computer Security,”in AAAI Workshop: AI Approaches to Fraud Detection and Risk Management, 1997.

[13] K. Sequeira, M. Zaki, “ADMIT: Anomaly-base Data Mining for Intrusions,” in ACM SIGKDD Conference, Edmonton, 2002.

[14] V. Barnett, T. Lewis, “Outliers in Statistical Data,” John Wiley and Sons, NY 1994.

[15] C. C. Aggarwal, P. Yu, “Outlier Detection for High Dimensional Data,” in ACM SIGMOD Conference, 2001.

[16] M.F. Chiang, W.C. Peng and C.H. Lo, "Discovering Popular Co-Cited Communities in Blogspaces," in ICDE workshop on Data Engineering for Blogs, Social Media, and Web 2.0, 2008.

[17] Y. Chi, S. Zhu, X. Song, J. Tatemura, and B. L. Tseng, “Structural and temporal analysis of the blogosphere through community factorization,” in ACM SIGKDD, 2007.

[18] Alberts, Christopher J. and Dorofee, Audrey J, ”OCTAVESM Method Implementation Guide,” v2.0, Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001.

[19] C&A Systems Security Ltd, COBRA: Introduction to Risk Analysis, http://www.ca-systems.zetnet.co.uk/risk.htm

[20] International Information System Security Certification Consortium (ISC)2, Security management, http:://www.os-global.com

[21] L.C. Wuu, Chan S.H., and C.H. Hung, “Implement an IP Traceback on Linux Platform,” 2006 symposium on Open Source Technology and Application, TAIWA.

[22] H. Wang, D. Zhang, and K. G. Shin, “Detecting SYN flooding attacks,” In Proceedings of IEEE INFOCOM, 2002.

[23] S. Singh, C. Estan, G. Varghese, and S. Savage,”Automated Worm Fingerprinting,”In Proceedings of Symposium on Operating Systems Design and Implementation,(OSDI), 2004.

[24] Shigang Chen, and Sanjay Ranka, “Detecting Internet Worms at Early Stage,” IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL.23, No. 10, 2005.

[25] B. E. Brodsky and B. S. Darkhovsky., Nonparametric Methods in Change-point Problems, Kluwer Academic Publishers, 1993.

[26] J. Kang, Z. Zhang, and J. B. Ju, “Protect E-commerce against DDoS Attacks with Improved D-WARD Detection System”, IEEE International Conference on e-Technology, e-Commerce and e-Service, March 2005.

[27] Peter Mell, Karen Scarfone and Sasha Romanosky, “CVSS A Complete Guide to the Common Vulnerability Scoring System Version 2.0,” FIRST, http://www.first.org/cvss.

[28] Microsoft Corporation. Microsoft Security Response Center Security Bulletin Severity Rating System, http://www.microsoft.com/technet/security/bulletin/rating.mspx.

[29] United States Computer Emergency Readiness Team (US-CERT). US-CERT Vulnerability Note Field Descriptions. http://www.kb.cert.org/vuls/html/fieldhelp.

[30] SANS Institute. SANS Critical Vulnerability Analysis Archive. http://www.sans.org/newsletters/cva/.

[31] Yue-Lung Cheng, "Uncertainties in Fault Tree Analysis," Tamkang Journal of Science and Engineering, Vol. 3, No. 1, 2000.

[32] Liang G. S. and Wang M. J., "Fuzzy fault tree analysis using failure possibility, Microelectron Reliability," Vol. 33, No. 4, 1993.

[33] Misra K. B. and Weder G. G., "A new method for fuzzy fault tree analysis," Vol. 29, No. 2, 1989.

[34] Tanaka H., Fan L. T., Lai F. S. and Toguchi K., "Fault tree analysis by fuzzy probability," IEEE Trans. Reliability, Vol. 32, No. 5, 1983.

[35] NG Leveson, SS Cha, TJ Shimeall , "Safety verification of ada programs using software fault trees," IEEE software, 1991.

[36] Guy Helmer, Johnny Wong, Mark Slagell, Vasant Honavar, Les Miller and Robyn Lutz, "A Software Fault Tree Approach to Requirements Analysis of an Intrusion Detection System," Requirements Engineering, Vol. 7, No. 4, 2002.

[37] 林盈達,林柏青,"網路安全產品測試評比-功能與效能面"

[38] P. Berkhin, "Survey of Clustering Data Mining Techniques," Accrue Software, 2002.

[39] Jain, Murty and Flynn: "Data Clustering: A Review," ACM Comp. Surv., 1999.

[40] K. Krishna and M. Narasimha Murty, "Genetic K-Means Algorithm," IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 29, NO. 3, JUNE 1999

[41] Li, Y.J., Bo, L., A Normalized Levenshtein Distance Metric, PAMI(29), No. 6, June 2007, pp. 1091-1095.

[42] Yue-Lung Cheng, “Uncertainties in Fault Tree Analysis,” Tamkang Journal of Science and Engineering, Vol. 3, No. 1, 2000.

[43] Liang G. S. and Wang M. J., “Fuzzy fault tree analysis using failure possibility, Microelectron Reliability,” Vol. 33, No. 4, 1993.

[44] Misra K. B. and Weder G. G., “A new method for fuzzy fault tree analysis,” Vol. 29, No. 2, 1989.

[45] Tanaka H., Fan L. T., Lai F. S. and Toguchi K., “Fault tree analysis by fuzzy probability,” IEEE Trans. Reliability, Vol. 32, No. 5, 1983.

[46] T. Baba and S. Matsuda, “Tracing network attacks to their sources,” IEEE Internet Computing, vol. 6, pp. 20-26, Apr. 2002.

[47] S. Savage, D. Wetheral, A. Karlin, and T. Anderson, “Practical network support for IP traceback,” in Proceedings of ACM SIGCOMM’00, vol. 30, pp. 295-306, Oct. 2000.

[48] A. Belenky and N. Ansari, “IP traceback with deterministic packet marking,” IEEE Communication Letters, vol. 7, pp. 162-164, Apr. 2003.

[49] A. Belenky and N. Ansari, “Tracing multiple attackers with deterministic packet marking (DPM),” in Proceedings of IEEE Pacific Rim Con. Communications, Computers and Signal Processing, vol. 1, pp. 49-52, Aug. 2003.

[50] A. Yaar, A. Perrig, and D. Song, “Pi: A path identification mechanism to defend against DDoS attacks,” in Proceedings of IEEE Symp. Security and Privacy, pp. 93–107, May 2003.

[51] A. Yaar, A Perrig, and D. Song, ”StackPi: New Packet Marking and Filtering Mechanisms for DDoS and IP Spoofing Defense,” IEEE JSAC, Volume 24, Issue 10, pp. 1853 – 1863, Oct. 2006.

[52] R. Chen, J. Park, and R. Marchany, “RIM: Router Interface Marking for IP Traceback,” in Proceedings of IEEE GLOBECOM, pp. 1-5, Nov. 2006.

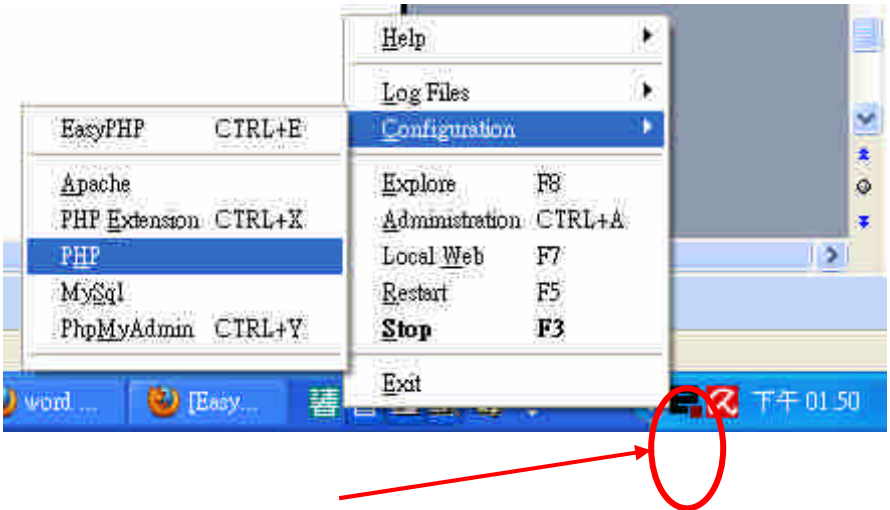
[53] Stefan Savage, David Wetherall, Anna Karlin and Tom Anderson, Practical Network Support for IP Traceback. In Proceedings of ACM SIGCOMM 2000 , pp. 295-306, August 2000.

附錄一、誘捕網系統操作安裝手冊

Windows 環境：

1. Windows XP
2. 安裝 Apache & PHP
   EasyPHP : http://www.easyphp.org/

3. 設定 PHP 環境

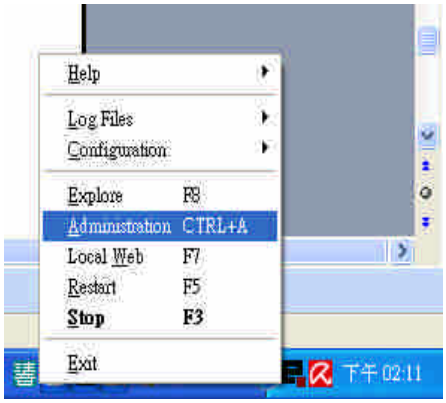


執行後，右下角會出現圖示，按下滑鼠右鍵後會出現選單，設定環境變數是在 Configuration 內，將 magic_quotes_gpc = Off

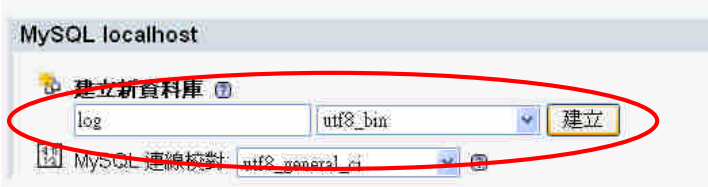
4. 建立資料庫

有兩種方法，4.1~4.3 介紹手動建立，4.4 介紹快速建立，可擇一執行。

4.1 進入 phpMyAdmin



4.2 建立資料庫: log 以及 msg_board



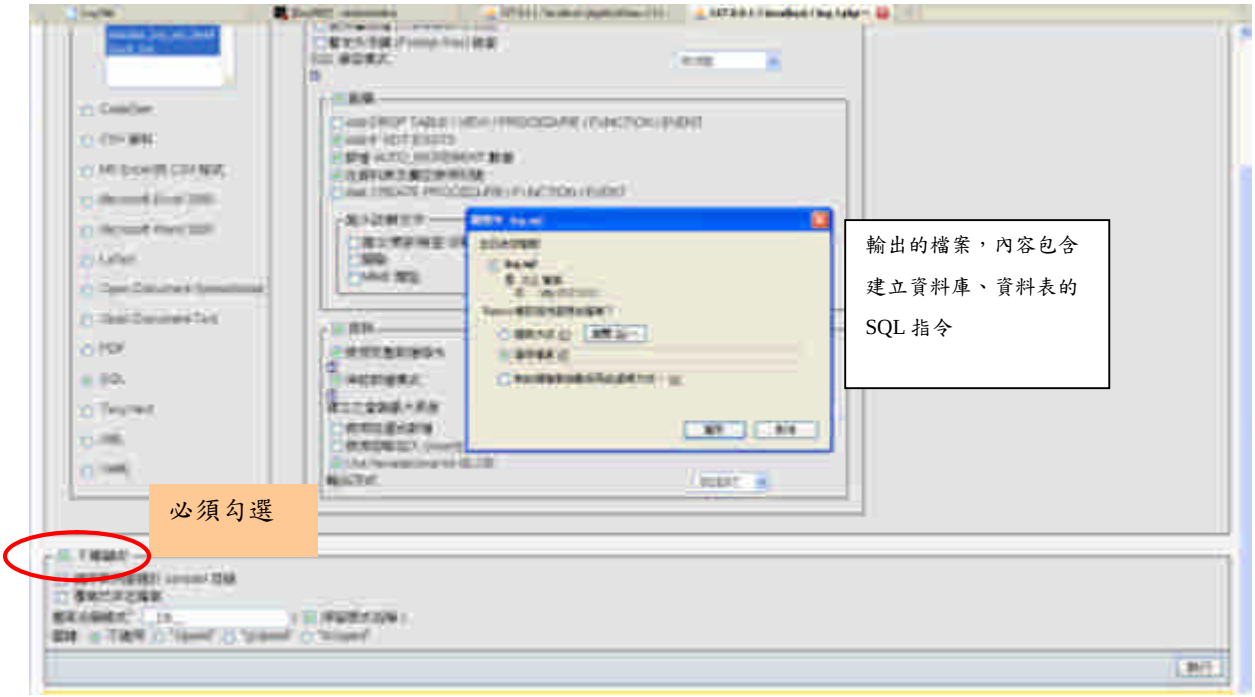
4.3 請參照上述 *資料庫-log*，*資料庫-msg_board*，設定資料表

※4.4 快速建立資料庫：

先複製一份舊資料(輸出)，再將舊資料貼上到新資料庫(載入)

4.4.1 選擇要複製(輸出)的資料庫-log，點選輸出

4.4.2 儲存輸出的檔案



之後再將此檔案進行載入的動作，即可完成。

5. 程式設定

請參考 module.php 的介紹，設定新環境的變數，例如: IP 等。

附錄二、封包標記與追蹤系統操作安裝手冊

系統安裝

第一步驟：

安裝 Linux-Ubuntu 8.04 LTS。

1.將光碟片 Ubuntu 8.04 LTS 放入光碟機。

2.進入開始選單，選擇[中文(繁體)]。

3.將選擇移到[安裝 Ubuntu(I)] 並執行。

4.接續依照畫面指示進行安裝。

FAQ：

當進行安裝時，出現 BusyBox 的問題，在第 3 步將選擇停留在[安裝 Ubuntu(I)]，按 F6 後，螢幕下方會出現修改開機參數的地方(游標在下面那行字的最右邊)，在後面接著輸入底下參數：

[all_generic_ide floppy=off irqpoll]，Enter 進系統，應該就可正常安裝。

第二步驟：

APT-GET 套件安裝。

1.將光碟片 Traceback Source 放入光碟機。

2.複製所有檔案到桌面(ex:/home/[user]/Desktop/底下)。

3.執行 Terminal，進入到複製的[APT-GET PACKAGE]資料夾底下。

4.Terminal 進入 01-G++的資料夾後，執行[# dpkg –i *.deb]，進行套件安裝。

5.如果在安裝過程出現錯誤，請再次執行 4 的指令，錯誤是由於安裝順序的問題，第二次執行就可成功。

6.進入下一個編號的資料夾，重覆 4-5 步驟，將所有套件安裝完成。

7.開啟 Terminal，執行[# sudo vim /etc/bash.bashrc]，在最後加上[export CLASSPATH=$CLASSPATH: /usr/share/java/mysql.jar]。

8.複製 mysql-connector-java-5.0.8-bin 到 [/usr/lib/jvm/java-6-sun-1.6.0.07/jre/lib/ext] 目錄下

9.進行重新開機。

FAQ：

在安裝 mysql-server 套件時，需設定 mysql 中 root 的密碼。在安裝 phpmyadmin 時，所出現的選項請選擇[apache2]。

第三步驟：

Kernel 核心編程

1.將第二步驟所複製的 linux-2.6.24.6.tar 解壓縮到資料夾[/usr/src/]底下。

2.執行 Terminal，輸入[#sudo -s]，將權限換至 root。

3.進入到[/usr/src/linux-2.6.24.6]下，然後執行[# make menuconfig]，進入畫面後，將選擇移到最底下的[LOAD Alternative Configuration File]並進入，按 <Ok>。接續將選擇移到最底下的[SAVE Alternative Configuration File]並進入，按<Ok>，之後執行<Exit>選項。

4.接續在同樣目錄下執行[# make bzImage]，[# make modules]，[# make modules_install]。

5.執行[# mkinitramfs –o /boot/initrd.img -2.6.24.6 2.6.24.6]。

6.執行[# cp /usr/src/Linux-2.6.24.6/arch/ i386/boot/bzImage /boot/vmlinuz-2.6.24.6]。

7.執行[# cp /usr/src/Linux-2.6.24.6/System. map /boot/System.map-2.6.24.6]。

8.執行[# update-grub]，最後再重新開機，開機選單會多出一個核心的選項，選擇剛編譯好的 Kernel 使用。

第四步驟：

System Setup 系統設定

1.開啟 firefox 瀏覽器，網址輸入[http://127.0.0.1/phpmyadmin/]，進入資料庫頁面，使用 root 帳號登入。

2.點選[權限]→[新增使用者]→使用者[wnl]，主機[localhost]，密碼[1234]，[整體權限-全選]→執行，點選[新增使用者]→使用者[wnl]，主機[127.0.0.1]， 密碼[1234]，[整體權限-全選]→執行。

3.回到主頁面，點選[重新讀取權限]。

4.回到主頁面，建立新資料庫[wnl]，點選左邊資料庫的 wnl，再點選右邊選項[輸入]，文字檔案的位置選擇光碟中的[MySQL Database/tam.sql]並且執行，就能夠建立好 table。

5.MCC 裡的 table 也是如上的建法，文字檔案的位置選擇光碟中的[MySQL Database/mcc.sql]並且執行，就能夠建立好 table。

6.router 的設定，開啟 Terminal，執行

(1) #vim /etc/quagga/daemons 編輯/etc/quagga/目錄下的 daemons 將 zebra ，ripd 改為 yes 存檔

[zebra=yes
bgpd=no
ospfd=no
ospf6d=no
ripd=yes
ripngd=no]

(2) 進入光碟 router 裡，複製裡面的設定檔 zebra.conf 和 ripd.conf 到 [/etc/quagga]目錄下，這 2 個檔名不能任意更改。

(3) 修改設定檔 #vim /etc/quagga/ zebra.conf 用來設定主機網卡的 ip 位址和 netmask，如果有要修改 ip，只要修改 interface ethX 下面那一行指令即可，若要刪除或新增某個網卡只要刪除或新增

interface ethX
ip address X.X.X.X/X
ipv6 nd suppress-ra
!

#vim /etc/quagga/ ripd.conf
設定哪些網卡負責哪些網域的動態路由，若要修改只要刪除或新增
network X.X.X.0/24 //網卡 X 負責的網域 network ethX

(4) 進行 GUI 介面的網路設定，可點選右上的網路設定，或由左上點選系統→管理→網路，進行網卡介面設定，出現網路設定介面後，再點選解除鎖定，輸入密碼，點選認證，即可進行每張網卡的設定，再點選要設定的網卡→屬性，設定選固定 IP 地址，輸入要給網卡的 IP 地 址和子網域遮罩，按確定，完成網卡的設定。

FAQ:

有時 GUI 介面雖有設定網卡的 IP，但是網卡實際上沒被分配到 IP，會導致路由設定無法成功，這時可從終端機登入，輸入 ifconfig，檢查在主機上所有網卡介面的相關資訊，若是 eth0 IP 設定沒成功，可輸入指令#ifconfig eth0 X.X.X.X，分配 IP 給 eth0

(5) #/etc/init.d/quagga restart 啟動 quagga 軟體，進行路由封包的交換，主機若是重開機後，quagga 是會自動執行，若是想暫停可輸入#/etc/init.d/quagga stop。

(6)設定完後，可用 ping 指令，檢查設定是否完成，也可輸入 route –n 檢查各主機內的路由表。

(7)若是想要主機當 bridge，則在 GUI 網路介面設定時，每張網卡的網路設定皆點選啟用漫遊，並暫停 quagga 軟體，進行 bridge 指令設定。

7.開啟 Terminal，執行[#sudo vim /etc/rc.local]，在[exit 0]前加入如下：(如果要讓 MPC 也有自動路由功能，請在建 bridge 時，再在終端機手動輸入以下指令，不要寫進 rc.local)

---

ifconfig eth0 0.0.0.0
ifconfig eth1 0.0.0.0
brctl addbr br0
brctl addif br0 eth0
brctl addif br0 eth1
ifconfig br0 xxx.xxx.xxx.xxx netmask ooo.ooo.ooo.ooo up
/home/[user]/mpcset.o load

(xxx.xxx.xxx.xxx 代表 IP Address)

(ooo.ooo.ooo.ooo 代表 Netmask) 加入後將資料儲存。

8.執行[# crontab –e]，在最下方加入[* */12 * * * /home/[user]/myd.o]，並儲存離開，進行重新開機。

第五步驟：

Program Execution 程式執行

1.將光碟中[CMDsource]資料夾內的三個檔案複製到[/home/[user]/]資料夾下。

2.開啟 Terminal，到資料夾[/home/[user]/]，執行[# gcc mpcset.c –o mpcset.o]，[# gcc

mylistener.c –o myl.o –L/usr/lib/mysql -lmysqlclient]，[# gcc myd.c –o myd.o –L/usr/lib/mysql -lmysqlclient]，會產生 mpcset.o、myl.o 和 myd.o 三個檔案。

3.執行[# sudo ./mpcset.o iid 1]，會產生 mpc.config 檔。

4.執行 mpcset.o 進行需要的核心參數設定。

5.開啟額外兩個 Terminal 視窗，分別執行[# sudo ./myl.o eth0]和[# sudo ./myl.o eth1]，進行封包監聽。

6.所有設定及執行完成，功能全部啟動。

7.開啟 Terminal，將光碟裡的[Tracebacke]資料夾複製到[/home/[user]/]，進入到[/home/[user]/Traceback]，執行[# java MPCSide]，即可執行 MPC server 服務，執行[# java Traceback]，即可執行 traceback GUI 介面。

MPC 開機操作

MPC 開機過後，首先會有秒數到數讓你選擇使用哪一個 Kernel，預設是使用 Kernel-2.6.24.6，選擇完畢後，會繼續進入到系統，出現視窗畫面後，輸入帳號密碼進入 Ubuntu 圖型介面。

開機進入後，MySQL Server 與 Apache 已預設為執行，若需確認 Apache 是否有執行可以開啟 Firefox 瀏覽器，在網址輸入 http://127.0.0.1，若有出現”It works”的畫面，表示 Apache 正常執行。

要使 MPC 正常運作，需要執行三個程式，開啟[應用程式]-[附屬應用程式]-[終端機(Terminal)]，總共開啟三個，第一個進入 Traceback 資料夾所放置的地方(預設放置在~/Traceback/，輸入”cd ~/Traceback/”)，之後執行”sudo java MPCSide”，開始 Server 監聽 Port。第二個進入 myl.o 的目錄下(預設為~/，輸入”cd ~”)並執行“sudo ./myl.o eth0”，第三個同樣進入 myl.o 的目錄下並執行“sudo ./myl.o eth1”，後兩個執行是將封包標記進行記錄的執行，若還有其它網卡，如上執行指令，這是 MPC 有加入自動路由的設定。若是 MPC 當 bridge 則執行“sudo ./myl.o mybridge”和任選主機上的一張網卡來監聽，執行“sudo ./myl.o ethX”。

將三個執行完成後，MPC 標記與記錄功能就完全開啟，放置著就能執行它的功能，變成一台可以進行標記的 router 或 bridge，當 router 執行的畫面如圖 A-1，當 bridge 執行的畫面如圖 A-2。

圖 A-1 當 router 執行程式的畫面

圖 A-2 當 bridge 執行程式的畫面

系統核心設定

使用 mpcset.o 這支程式去更改核心的標記設定。首先使用終端機進入到 mpcset.o 的目錄下(預設為~/，輸入”cd ~”)，如果需要將 IID 改變為 2，輸入”sudo ./mpcset iid 2”，觀看目前 iid 值，輸入”sudo ./mpcset iid”。執行參數可以使用”./mpcset.o help”進行查詢。執行結果如圖 A-3 所示。

圖 A-3 執行的結果

在已有安裝 Java 的主機上，放置整個 Traceback 資料夾，然後執行 Traceback.java，輸入"java Traceback"，GUI 介面就能夠執行出來，如圖 A-4。

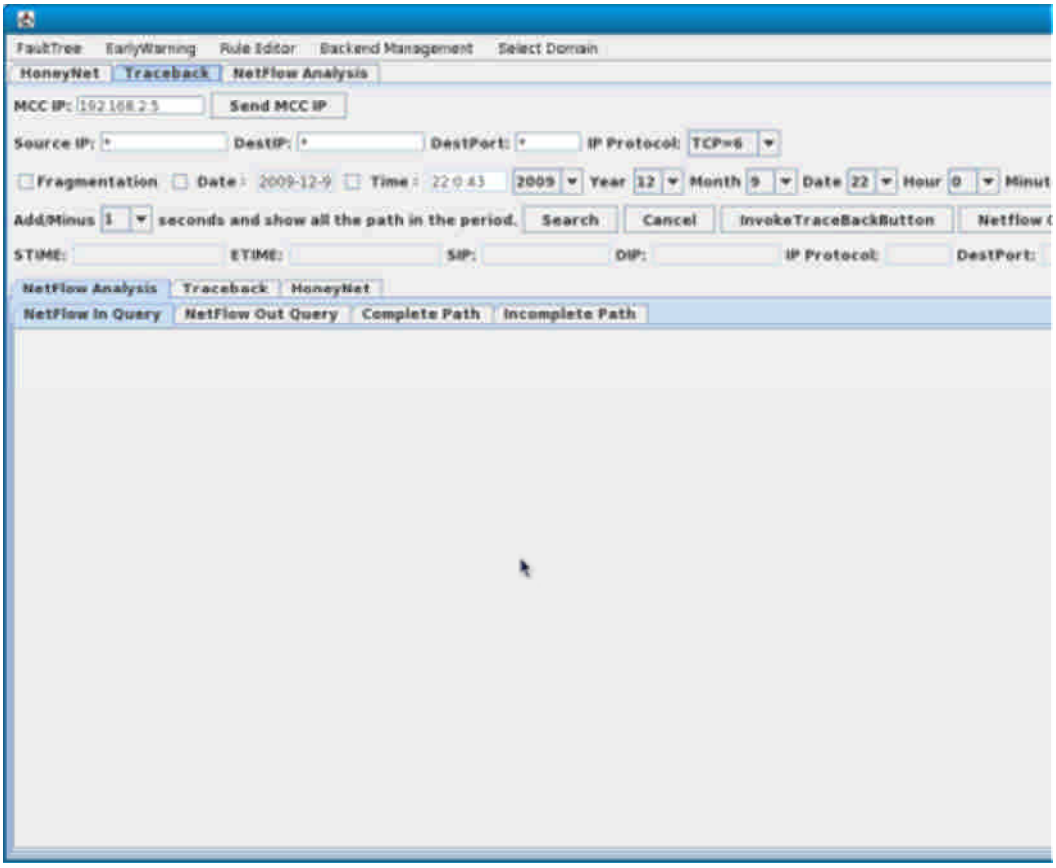


圖 A-4 Traceback 畫面

1.先輸入放置 Traceback.java 的主機 IP，按下按鈕”Send MCCIP”，一定要輸入 IP，否則將無法正常使用此軟體。

2.Traceback -Traceback

在手動輸入查詢條件，進行自動路徑回追的呈現，需要輸入欲查詢封包的相關資訊，其中 Source IP、Destination IP、Destination Port 和 IP Protocol 為必要輸入的查詢條件，Source IP、Destination IP 放置 IP 位址(如 192.168.2.4)，DestPort 則是放數字，其範圍在 0~65535。IP protocol 放置 6 代表 TCP 協定，17 代表 UDP 協定。而 Fragmentation、Date 和 Time 為可勾選項目，勾選 Fragmentation，則路徑只會呈現出入口端，不管封包是否有被切割，皆可使用此選項。按下”Search”，系統即會自動回追，重建封包經過的路徑，呈現畫面如圖 A-5，要進行下一筆查詢，需按”Cancel”。若等一段時間，皆未呈現完整路徑，則此路徑可能為不完整路徑，此時需按”Cancel”，至”Incomplete Path”查看不完整路徑。最下一排五個欄位呈現的是上一筆查詢所輸入的查詢條件。

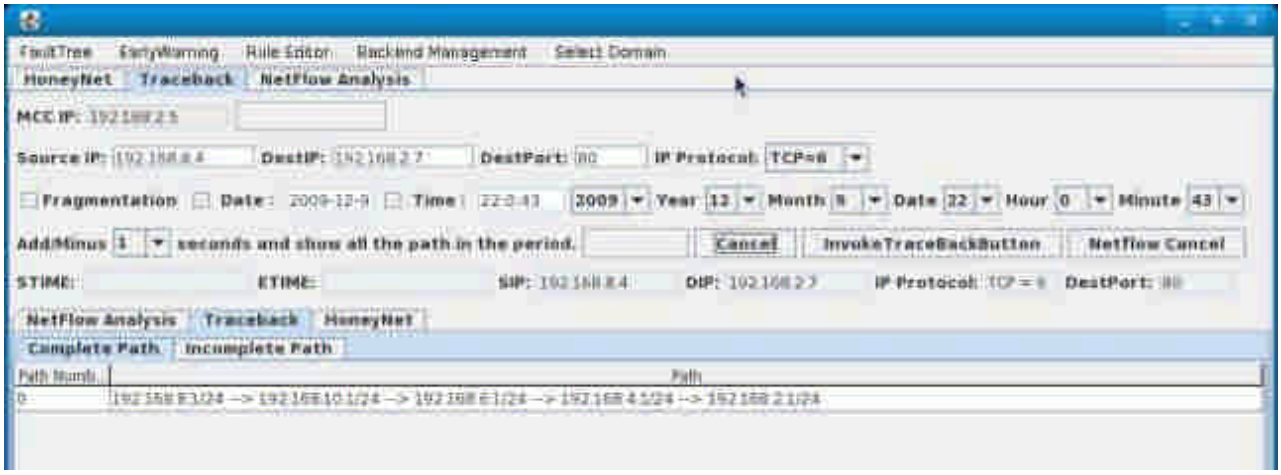
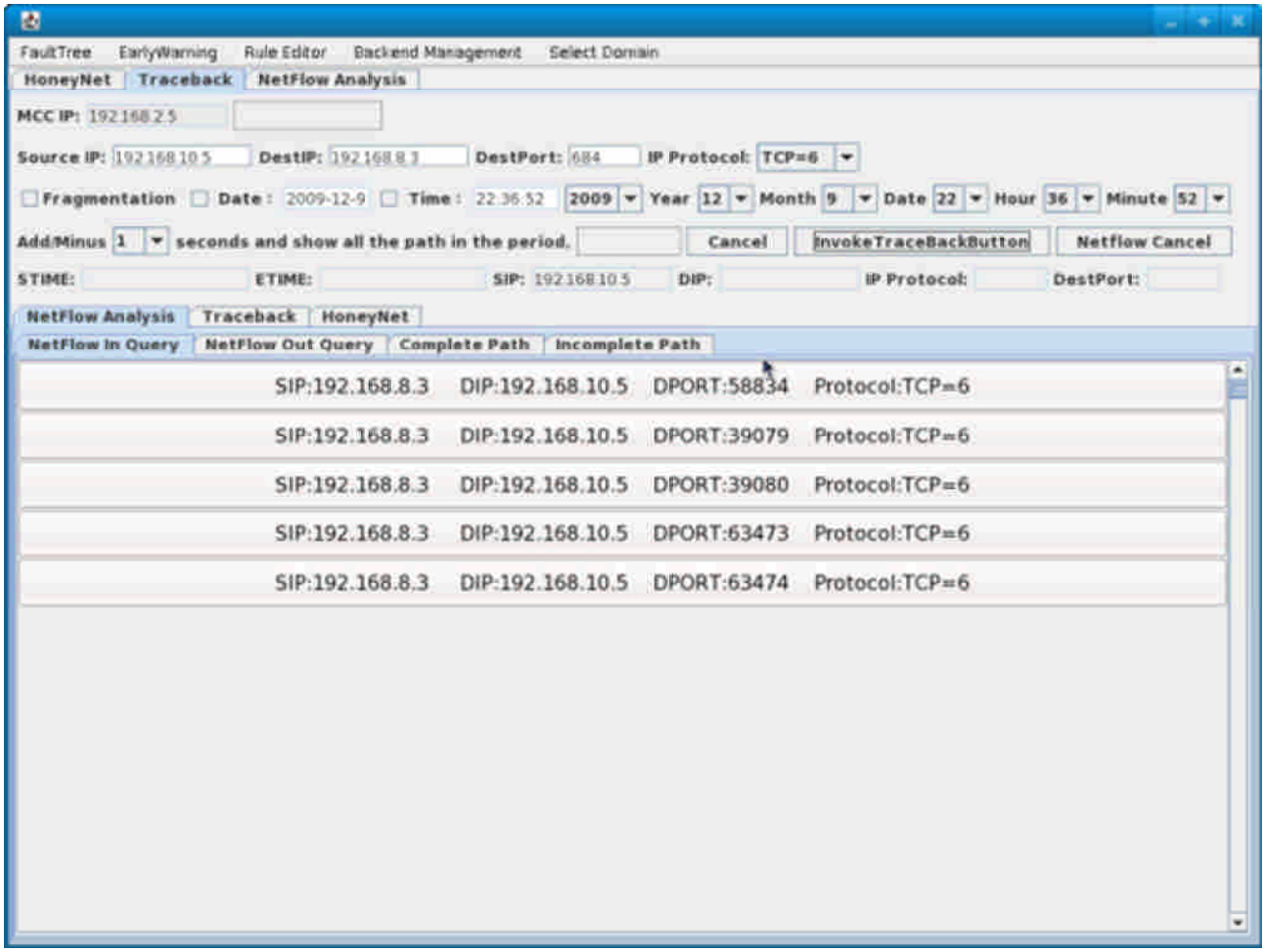


圖 A-5 Traceback 路徑呈現

3.Traceback-NetFlow Analysis

此介面呈現的是和法則子計畫的整合，當法則子計畫按下紅色異常節點主機，即會傳給 Traceback 一組 IP，Traceback 根據此 IP，呈現所有跟此 IP 相關的連線資訊，如圖 A-6，”NetFlow In Query”呈現的是跟此 IP 相關的流入連線資訊，也就是此 IP 當來源 IP。

圖 A-6 流入連線資訊



如圖 A-7，”NetFlow Out Query”呈現的是跟此 IP 相關的流出連線資訊，也就是此 IP 當目的 IP。

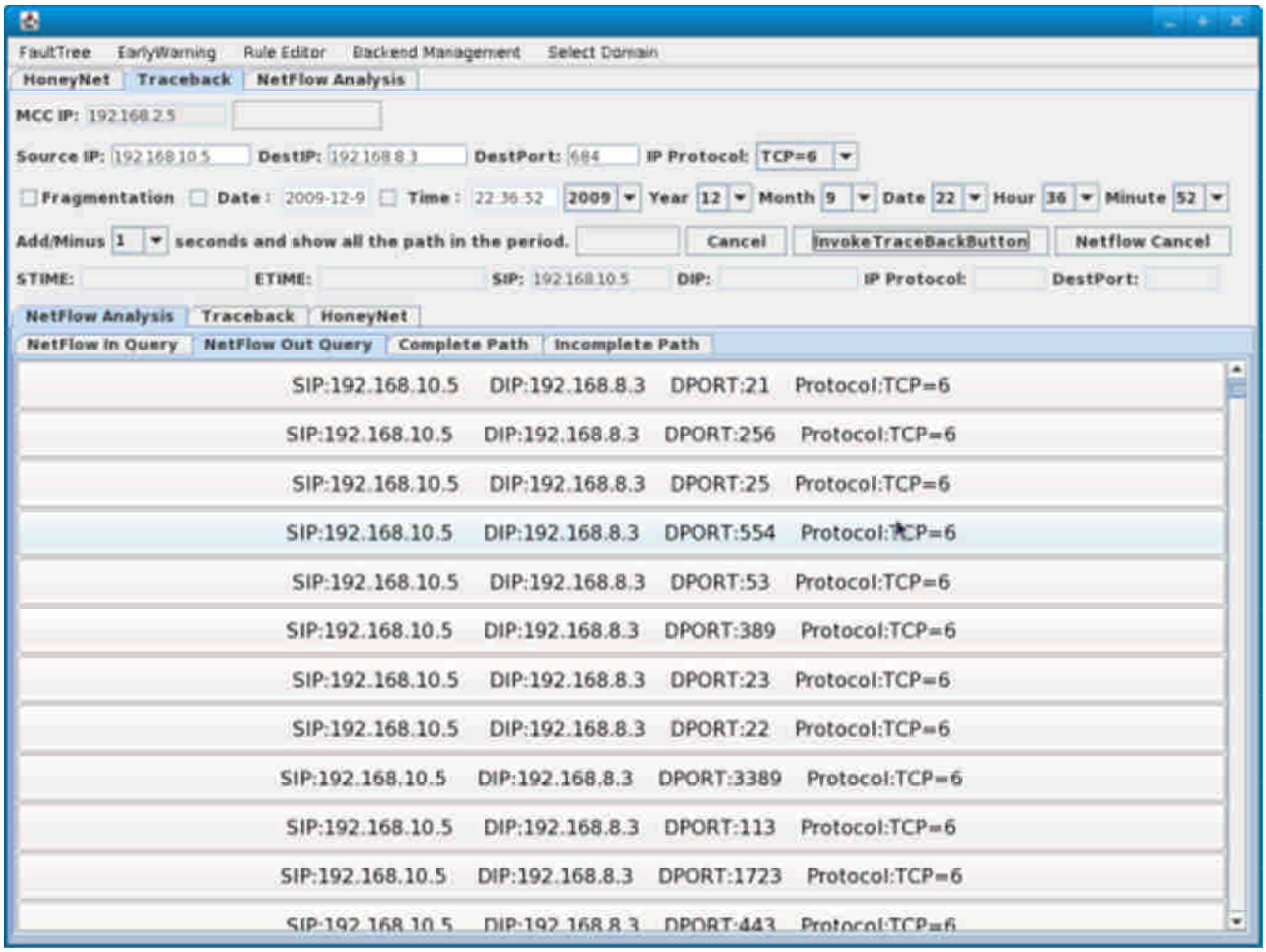


圖 A-7 流出連線資訊

點選”NetFlow In Query”或”NetFlow Out Query”介面下的某筆資料，Traceback 即會自動進行回追，點選後所有在”NetFlow In Query”或”NetFlow Out Query”介面下的按鈕皆會鎖住，如圖 A-8，

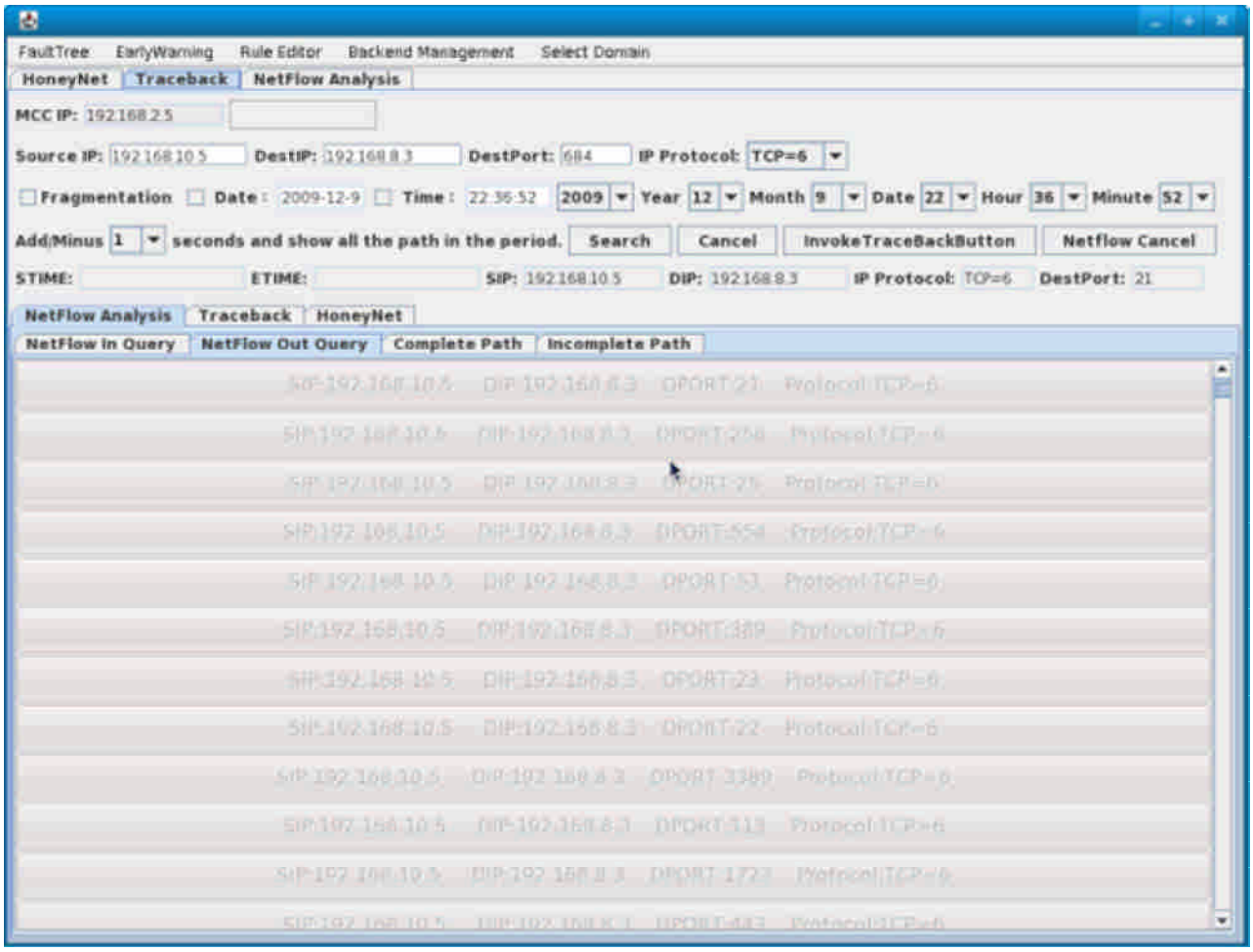


圖 A-8

可到”Complete Path”查看自動路徑重建結果，如圖 A-9，若要點選另一筆資料查詢，需按”NetFlow Cancel”，若是等了一段時間皆未呈現路徑，則此路徑可能為不完整路徑，可按”NetFlow Cancel”，即會在”Incomplete Path”呈現不完整路徑。

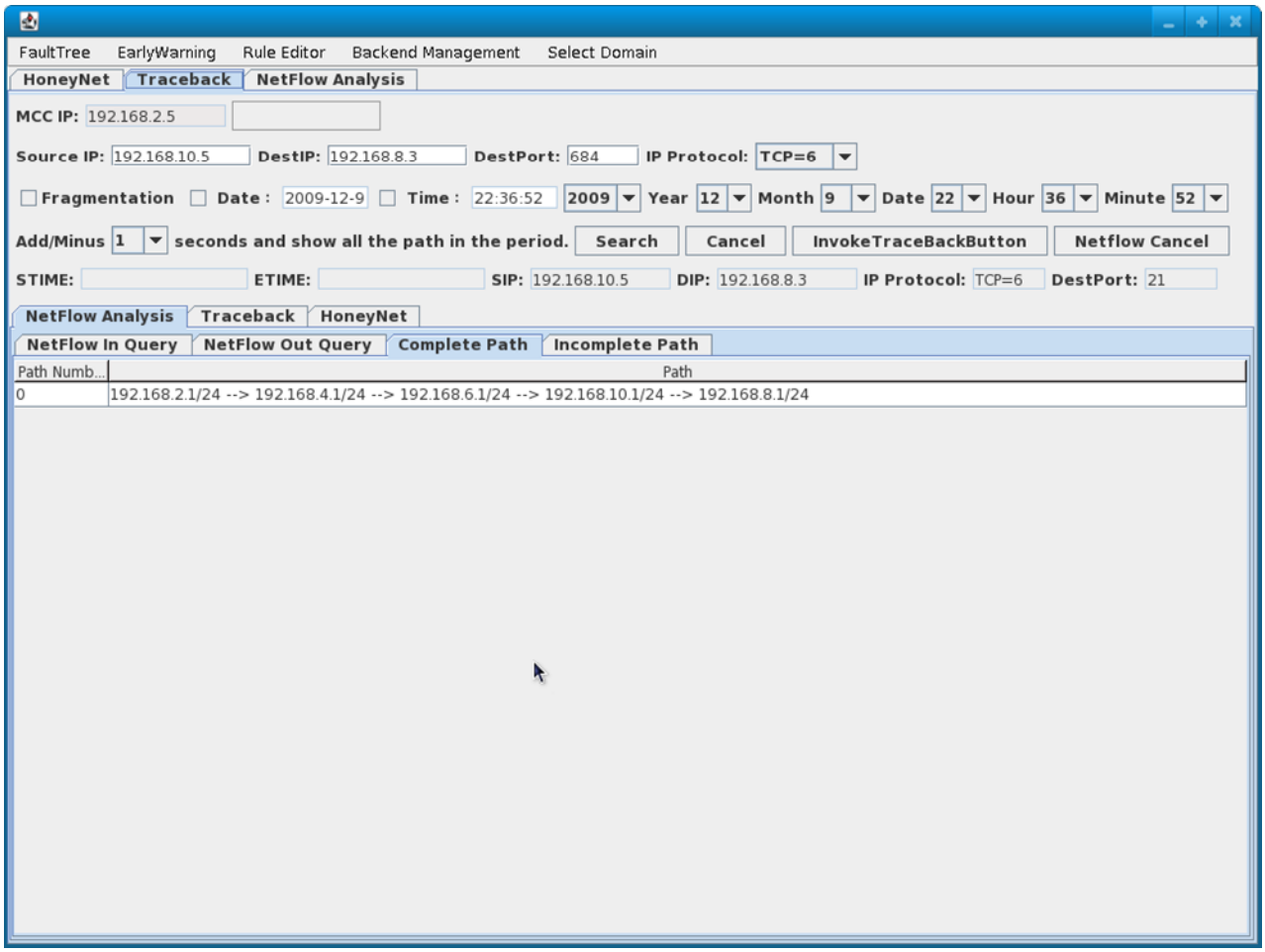


圖 A-9

4.Traceback-Honeynet

當有攻擊者連上蜜網子計畫的網站時，蜜網子計畫會自動傳送攻擊者相關資訊給 Traceback，此時 Traceback 即會自動進行回追，呈現結果如圖 A-10。

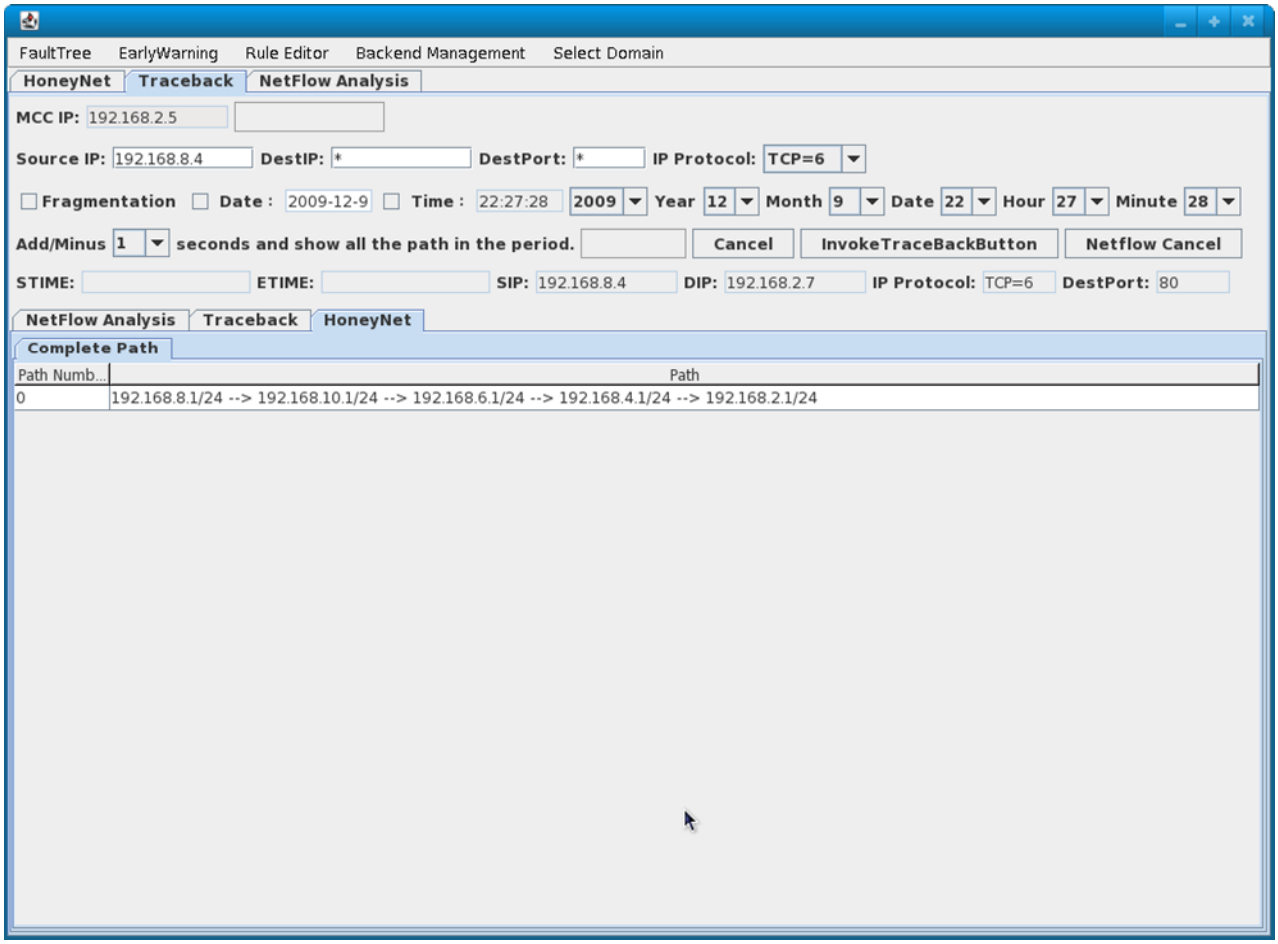


圖 A-10

附錄三、網路風險分析及預警系統操作安裝手冊

資料庫及資料倉儲工具使用說明

程式	說明
update_tid	用來更新 log 中 tid 的數值，由於 insert 新資料的時候並未對 tid 做處理，所以用這隻更新程式使之對應到對的 tid。 使用方法： ./update_tid <與更新之資料庫 IP>
log 與 log_sp	用來將資料導入資料庫的程式，log 適用於 demo 時使用的資料格式，log_sp 適用於中科院資料格式。 使用方法： 由於這隻程式沒做到 general，所以用於不同環境時需要修改內部的程式，需要改的地方有連接資料庫的地方和檔案的位置，函式分別為 mysql_real_connect 與 fin
insert_time	這是用來建立 time_by_hour 這張 table 裡的資料，可以建立從 2009~2100 間所有時間的資料。 使用方法： ./insert_time <資料庫 IP> <年>(限制在 2009~2100 之間)

表格 44、資料庫及資料倉儲工具使用說明

法則編輯器使用說明

1. 預設規則

在預設規則內，可以看見被設定為預設的規則。

如果覺得規則不在需要被訂為預設，可藉由刪除預設規則按鈕取消。

2. 規則瀏覽

在規則瀏覽中，可以看所有已增加的規則，選定左手邊規則後可以看到規則內容，若是覺得規則效果良好，可藉由新增預設規則訂為預設。

在閱讀規則時，有每日已經算好的 profile，可藉由選定日期後借由”顯示拓樸”的按鈕秀出資訊於 MCC 上。

Rule Editor

3. 使用者自訂規則

可讓使用者自訂內容，自訂方法為我們所設定的巢狀式規則，先設定主要規則，在依序設定子規則。

設定子規則時，可藉由參數微調按鈕查看此規則的風險值。

4. 群組設定

在此頁面可設定群組 IP。

5. 連接埠群組設定

在此頁面可設定群組 Port。

異常偵測及預警機制工具使用說明

1. 檔案說明：

相關程式已打包成 EarlyWarning.tar.gz ，請解壓到 ~dmsecurity/ 下，將產生一資料夾 ~dmsecurity/EarlyWarning/ ，其下包含下列檔案與資料夾：

b. bin

存放編譯完成的 java class 檔案的位置，使用 build.sh 該 script 重新編譯 EarlyWarning.jar 此可執行 jar 檔案時會使用到。

c. jar

存放可執行的 jar 檔案的位置，目錄下有 EarlyWarining.jar 和 ScoreBrowser.jar 兩個可執行 jar 檔案，EarlyWarning.jar 即為預警主程式，ScoreBrowser.jar 則是供 MCC 呼叫使用，瀏覽風險評估數值使用。

d. lib

存放本程式會使用到的其他外部 library，包含與 mysql database 連線使用的 mysql-connector-java-X.jar 和使用 Monderian OLAP Engine 會使用到的 library 等。另外，透過 build.sh 自行編譯完成的 EarlyWarning.jar 在執行時會即時參考這些 library，請互自行移動此資料夾內容，更多敘述請參考 build.sh 和 manifest.txt 之說明。

e. profile

存放特定主機的 Profile 資訊，以供 MCC 讀取使用，若該主機為 192.168.1.3，其對應的 profile 檔名為 192-168-1-3.xml。每日 EarlyWarning 程式執行完後會上傳到 CSIM 主機存放。

f. src

存放 EarlyWarning 主程式的原始碼檔案，詳見附件的 javadoc 文件。

g. .classpath

紀錄使用到的外部 java library 檔案，在使用 build.sh 手動產生 EarlyWarning.jar 時會以此為參考產生對應的 manifest.txt 檔案。

h. .project

使用 eclipse 編輯此 project 時的設定檔。

i. manifest.txt

編譯 EarlyWarning.jar 此一可執行 jar 檔案時，需透過一 manifest 檔案指定執

行之程式與相關引用外部 jara library 的位置。此 manifest.txt 可透過 build.sh 和 .classpath 兩者產生，並在 build.sh 產生 EarlyWarning.jar 時引入使用。

j. sim.xml

使用 Monderian OLAP Engine 時需要指定的 table schema，相關說明請參考資料庫儲之說明文件。

k. build.sh

此一可執行 script 是用來手動編譯可執行 EarlyWarning.jar 使用，程式上分成三個部份，第一部份生成本程式之 java class 檔案並置於 bin\ 目錄下，第二部份藉由 .classpath 檔案產生對應之 manifest.txt 文件，第三部份則藉由上述 manifest.txt 文件與第一部份所產生的 java class 檔案建立 EarlyWarning.jar 此可執行檔。

l. updateProfileAndRun.sh

此一可執行 script 會做以下三件事：使用 update_tid 此一程式更新資料庫內 log 這張表格內的 timd_id 資訊，然後執行 EarlyWarning.jar 計算風險預警值，最後將產生的 profile 檔案上傳到 CSIM 主機，並依日期存放在不同資料夾內。

2. 程式使用說明：

a. EarlyWarning.jar

本程式執行時可帶有 6 個參數進行設定，其中 simserver, csimserver, domain 三個參數是必定要指定，其餘三個參數可視需求指定。程式執行完畢會在 profile\ 下產生各監視主機的 profile，並於 standard output 印出如 ScoreBrowser 可見的風險值。

執行方法：

Usage: java -jar jar/EarlyWarning.jar -simserver <ip> -csimserver <ip> -domain <domainID> [Other options ....]

Other Options:

[-clusternumber k] [-date computeDay] [-debuglevel level]

i. –simserver <ip>

此一參數指定 SIM 主機的 IP address，一般來說 EarlyWarning.jar 會在 SIM 主機上執行，但必須指定其 IP

Address 不可使用 localhost 取代。

格式如： -simserver 192.168.6.3

ii. –csimeserver <ip>

此一參數指定 CSIM 主機的 IP address，因 EarlyWarning 程式會將有問題的主機與其連線資訊和風險值上傳到 CSIM 上管理，故需要提供 CSIM 主機的 IP address 以供連線使用。

格式如： -csimserver 192.168.2.4

iii. –domain <domainID>

此一參數指定本程式是執行在第幾個 SIM 主機，對應到 CSIM 上會取用不同的資料庫來存放本程式計算的結果。

格式如： -domian 1

iv. –date <computeDay>

此一參數指定此程式要計算特定日期的風險評估值，例如以 2009-12-25 來計算時會根據 2009-12-18 ~ 2009-12-24 之間的連線資訊來計算其風險值，並產生 2009-12-24 該天各主機的 profile。不提供此參數時，會將此參數定為執行程式的日期，並以前一週的連線資訊計算其風險值，產生前一日各主機的 profile。

格式如： -date 2009-12-25

v. –clusternumber <k>

此一參數指定要將監視的主機分成多少群，即 K-Medoids Algorithm 中的 K，若指定的 K 值比主機數還大，會自動縮小到一個可接受的程度，詳見原始碼。

格式如： -clusternumber 2

vi. –debuglevel <level>

此一參數指定要印出的輔助資訊其詳細程度，依 1 ~ 5 分成 5 個等級，預設值是 1，一般使用建議不要超過 3，印出的輔助資訊會在以 standard error 印出，不會影響

standard output 的資訊。

格式如： -debuglevel 3

b. updateProfileAndRun.sh

此程式首先會根據該執行主機上的資訊記錄以下欄位

i. User

記錄 CSIM 上存放 profile 在那位使用者的家目錄下，預設是 dmsecurity

ii. CSIM

記錄 CSIM 的主機 IP address，預設是 192.168.2.4

iii. SelfIPAddress

記錄本機的 IP address，預設透過 “ip address” 此一 unix 指令取得

iv. DomainID

記錄這是第幾台 SIM 主機，預設透過 “hostname” 此一指令取得

v. Today

今日日期，透過 “date” 此一 unix 指令取得

vi. Yesterday

前一日日期，透過 “date” 此一 unix 指令取得

vii. RemoteDir

紀錄 profile 將上傳到 CSIM 主機的那個位置，並依 profile 日期分為不同資料夾存放

viii. LocalDir

記錄生成的 profile 會存放在那個資料夾內

c. 其他環境與參數設定

i. 時間

執行前請先確定此一 SIM 主機與 CSIM 主機的時間一致，可使用 ntpdate 進行校時，使用方式：ntpdate <CSIM IP address>

ii. 監視主機群組

監視主機群組清單預設以 array 型式寫死於

HostClustering.java 此一檔案內，預設只針對 host_group 中 gid 為 8 的群組進行 clustering 與計算風險值，如有需求請自行擴充並以 build.sh 重新編譯 EarlyWarning.jar。其相關程式碼如下：

```
26
27          // the hosts we are monitoring
28          private static final int[] HOST_GIDS =
{ 8 };
29
```

3. 參考文件：

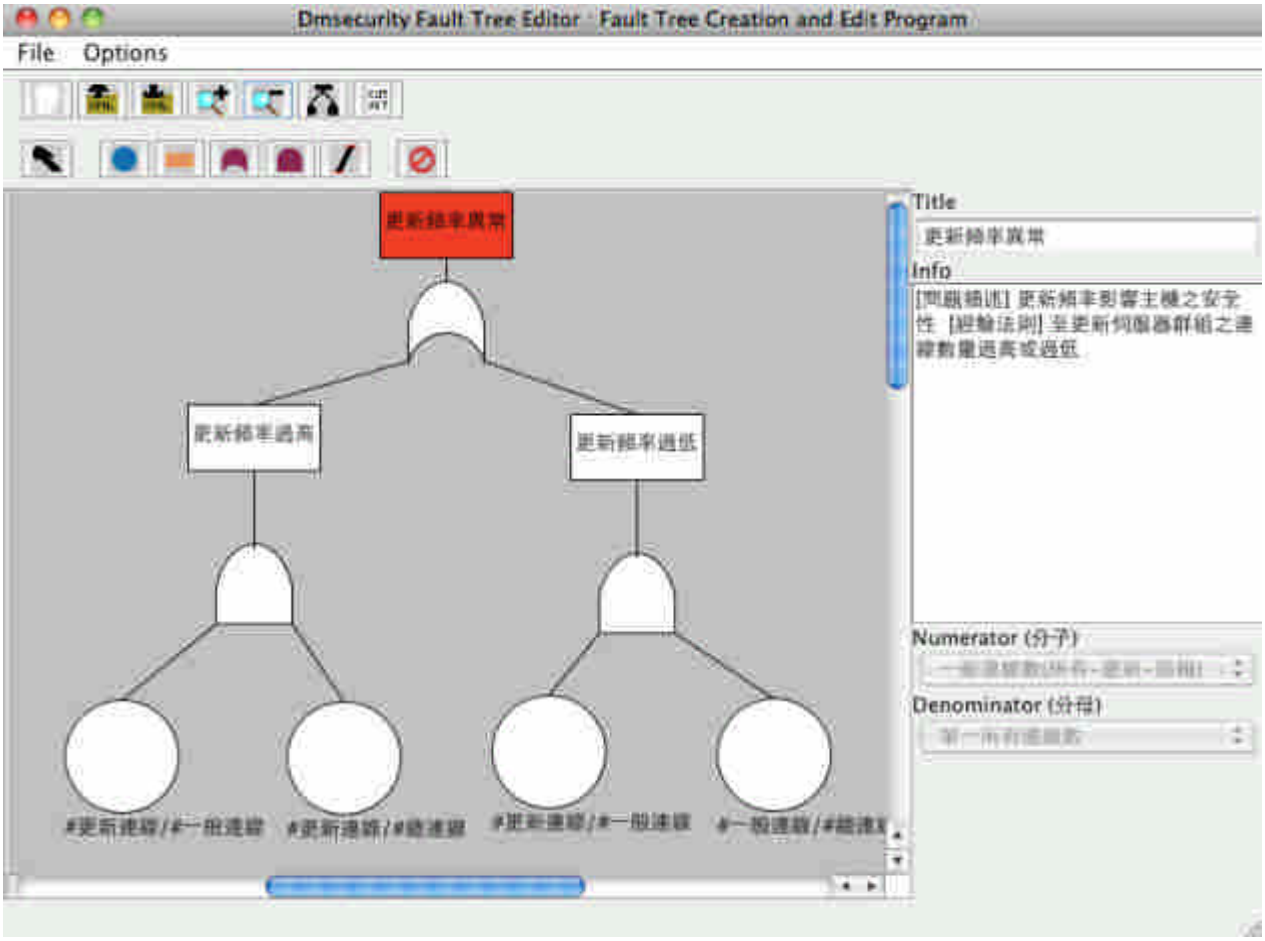
a. http://en.wikipedia.org/wiki/K-means_clustering
b. http://en.wikipedia.org/wiki/K-medoids
c. http://java.sun.com/docs/books/tutorial/
d. http://java.sun.com/javase/6/docs/api/
e. http://www.mysql.com/
f. http://mondrian.pentaho.org/
g. http://heirloom.sourceforge.net/man/sh.1.html

失誤樹風險分析工具使用說明

1. 失誤樹編輯器(FaultTreeEditor)

主要作用：

用於創造、修改及刪除失誤樹(Fault-Tree)。



失誤樹編輯器介面圖

使用方法：透過 MCC 呼叫。

選單說明：

File：

[New Page] ：新增一個空白的 FaultTree 編輯區。

[Open XML File] ：開啟一個 FaultTree XML 檔案。

[Save XML File] ：儲存一個 FaultTree XML 檔案。

[Delete XML File] ：刪除一個 FaultTree XML 檔案。

Options：

[Keep last selected component] ：保持最後選定的組件。

[Create Gate with Intermediate Event]：新增 Gate 時連帶一個 Intermediate Event。

工具列說明：

A B C D E F G

工具列說明

A：新增一個空白的 FaultTree 編輯區。

B：開啟一個 FaultTree XML 檔案。

C：儲存一個 FaultTree XML 檔案。

D：放大編輯區的物件。

E：縮小編輯區的物件。

F：自動對齊排列 FaultTree 物件。

G：秀出編輯區的 FaultTree 公式。

H I J K L M N

工具列說明

H：遊標工具，用來選擇或移動物件。(Alt+1)

I：產生一個基元事件，失誤樹中最底下的事件，可選擇產生機率值的分子分母。(Alt+Q)

J：產生一個有 Input/Output 的中介事件，除了頂上事件（樹根）沒有 Output。(Alt+W)

K：Or Gate，將下方連線的事件機率值的總和減去總積，得到的值往父節點送。(Alt+E)

L：And Gate，將下方連線的事件機率值全相乘，得到的值往父節點送。 (Alt+R)

M：可在物件間建立一個連線。 (Alt+2)

N：用來刪除物件。 (Alt+3)

資訊工具說明：

Title：使用者可自定元件的標題名稱，可空白。

Info：使用者自定的元件詳細資訊，可空白。

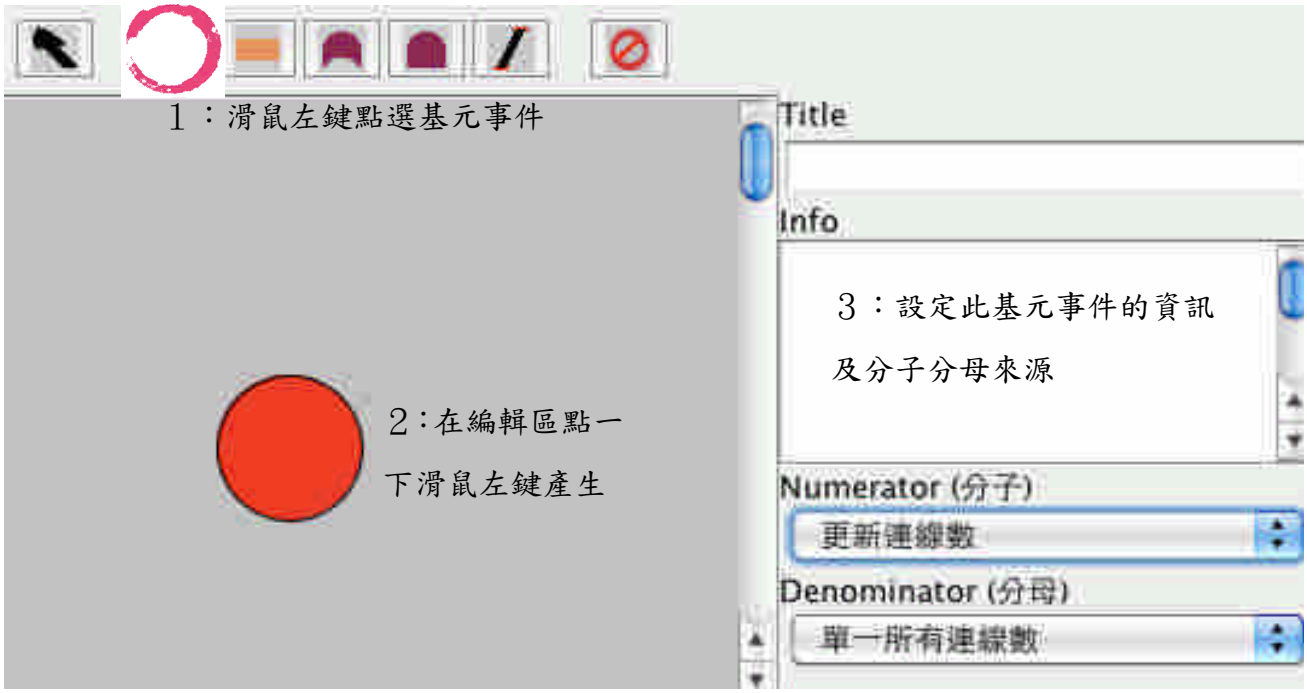
Numerator（分子）：基元事件的分子來源。

Denominator（分母）：基元事件的分母來源。

如何用 FaultTreeEditor 建一個 FaultTree：

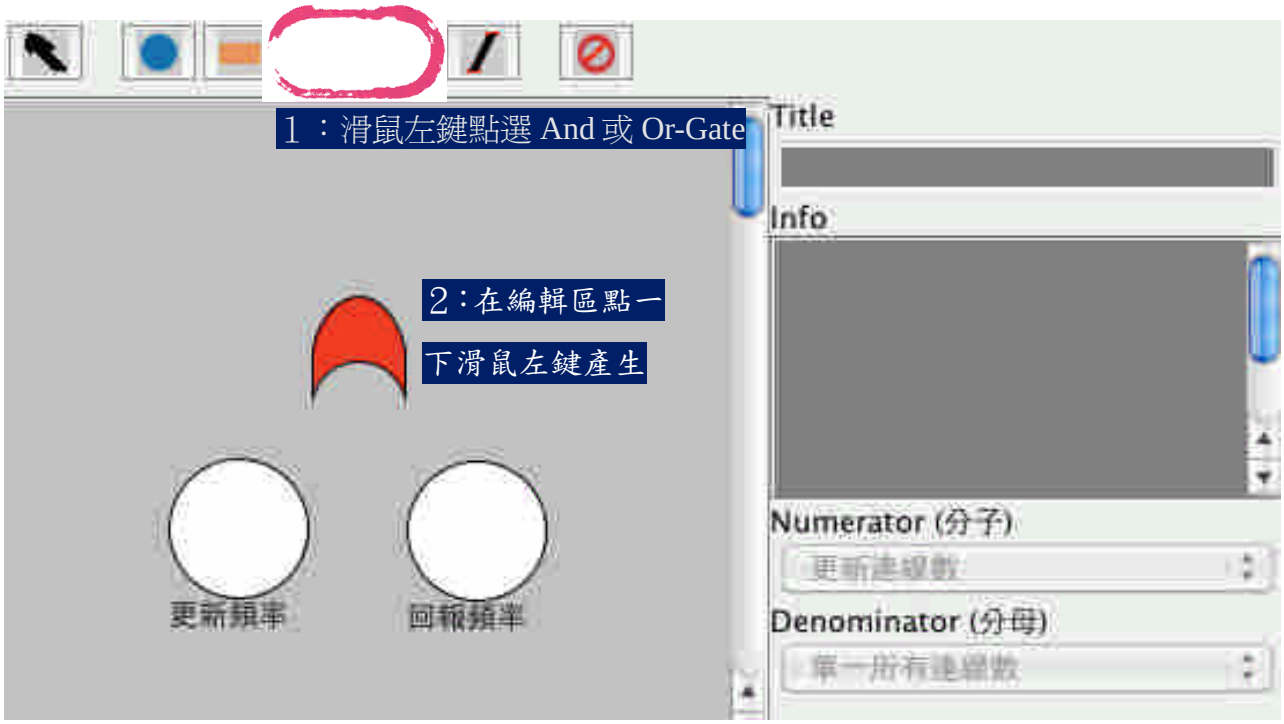
假設要產生一個「更新或回報異常」的失誤樹。

步驟一：建立基元事件



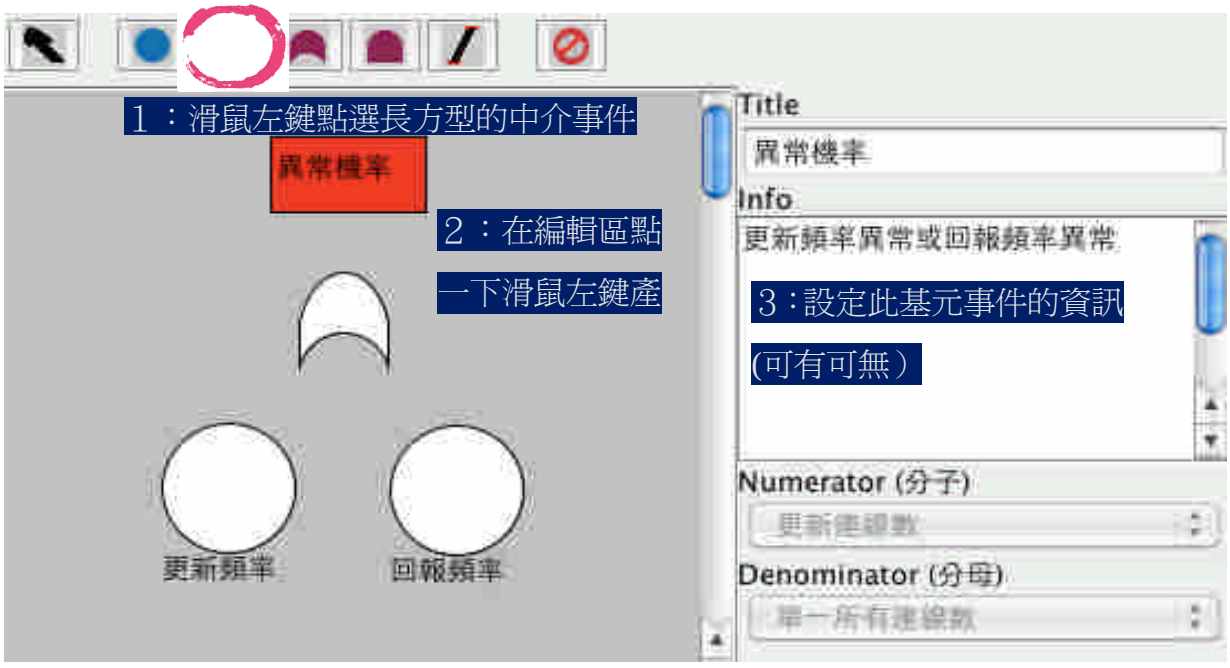
建立基元事件

步驟二：建立 And-Gate 或 Or-Gate



建立 And-Gate 或 Or-Gate

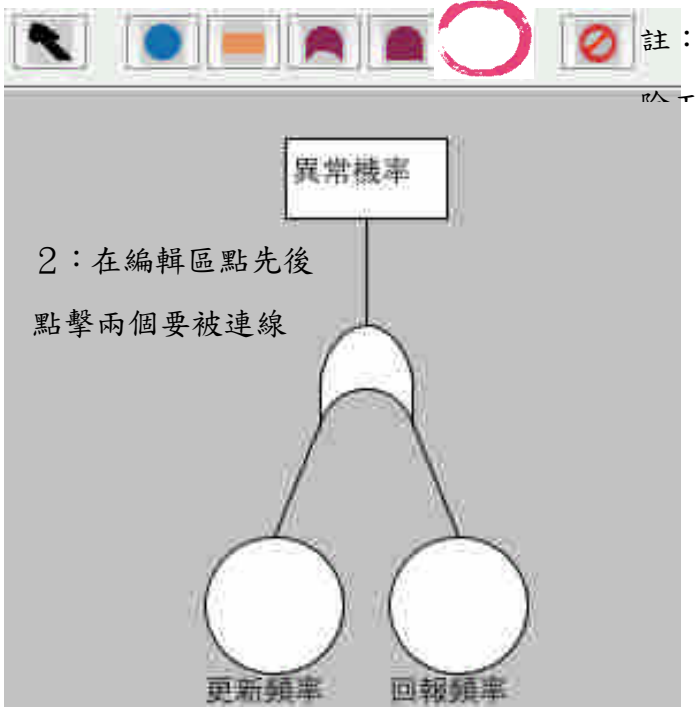
步驟三：建立頂上事件



建立頂上事件

步驟四：產生事件的連結

1：滑鼠左鍵點選連結物件



註：要刪除物件就點刪

產生事件的連結

最後存檔就可以產生一個描述此失誤樹的 XML 檔給風檢分析工具分析。

2. 失誤樹風險分析器(FaultTreeAnalysis)

主要作用：

根據分析者所定的失誤樹 XML，定時在每日計算出列管主機的風險值後，存入資料庫，計算結果可用失誤樹瀏覽工具觀看。

使用方法：

在終端機直接下指令：java -jar FaultTreeAnalysis.jar 就會開始分析當日的風險值。

專家指定日期用法：

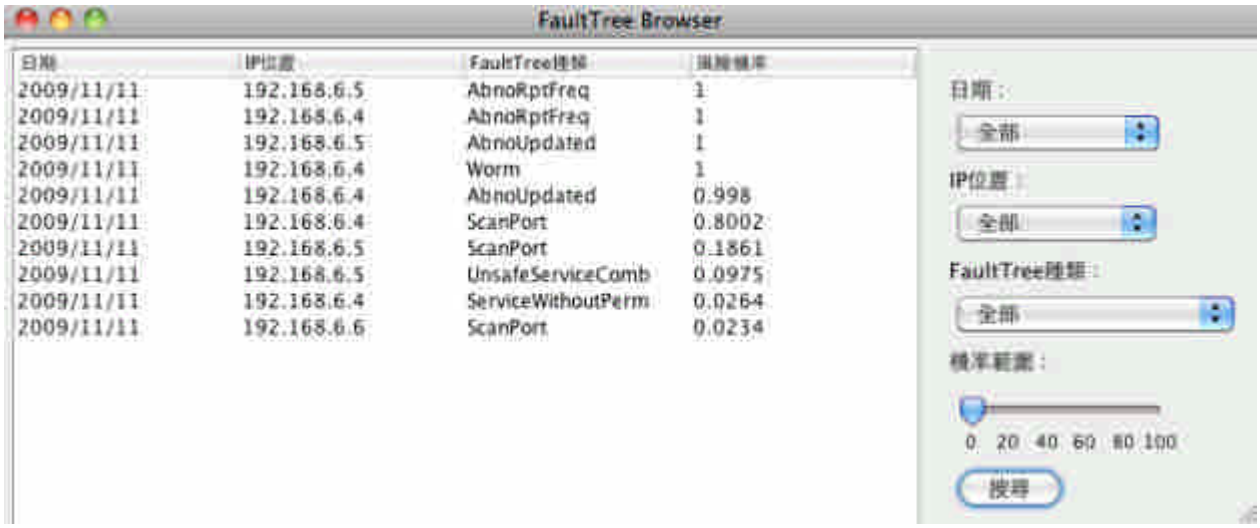
指令：java -jar FaultTreeAnalysis.jar [2009].[Q4].[11].[2].[11] 2009/11/11

↑何年.何季.何月.何週.何日↑存檔日期

3. 失誤樹風險瀏覽器(FaultTreeBrowser)

主要作用：

從資料庫抓出由 FaultTreeAnalysis 分析後產生出來的風險值資料來瀏覽。



失誤樹風險瀏覽器

使用方法：

從 MCC 呼叫。

功能：

1、可以依欄位排序

2、可以依日期、IP 位置、FaultTree 種類及機率範圍過濾。

附錄四、帳號權限管理系統操作安裝手冊

本操作手冊將會介紹如何設置 MySQL 內的資料庫。主要分成兩個步驟，第一步驟是設定資料庫，第二步驟是設定 JDBC。首先我們將會先介紹如何安裝 MySQL，並且在 MySQL 的介面建構出帳號資訊的表格。接著是敘述如何設定 JDBC，也就是如何設定和 MySQL 作連接。

STEP1 設定資料庫

先灌 MySQL，以 root 身分在終端機輸入下列指令

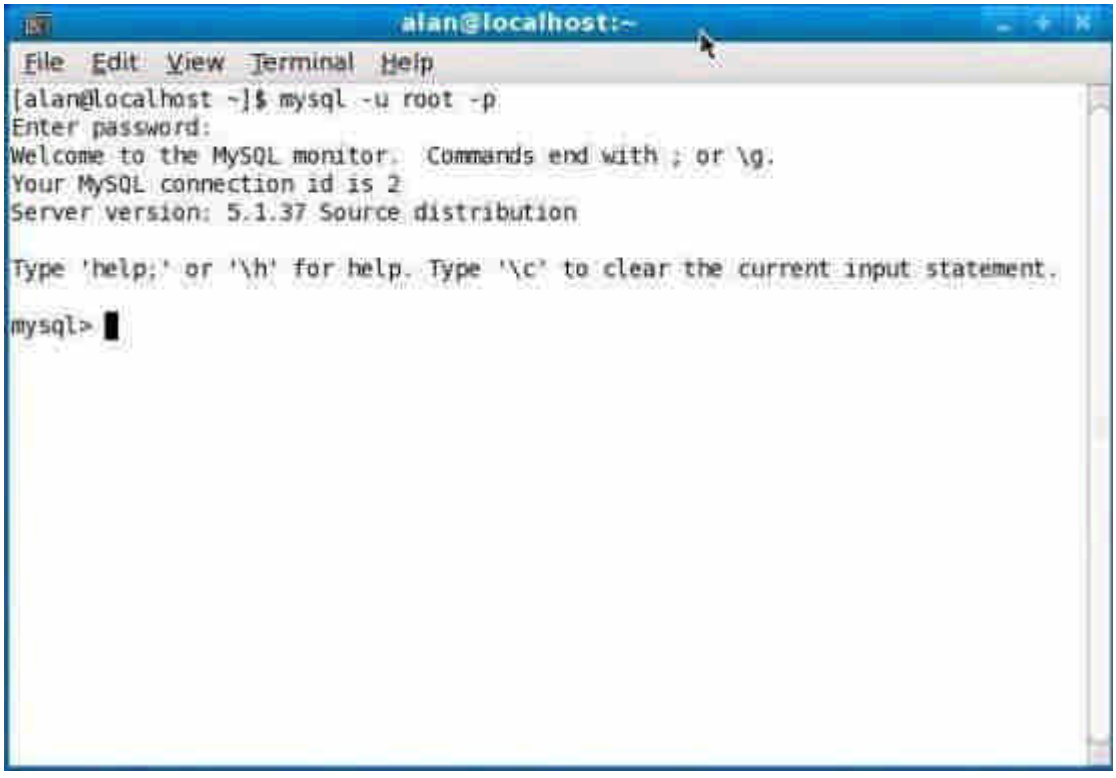
便可以安裝 MySQL，如下圖。

安裝時會要求輸入 root 使用者的密碼，請輸入密碼(以後還可以修改)，如下圖。

開啟終端機輸入下列指令

接著輸入設定的密碼便可以登入，如下圖。

登入 MySQL



我們的目標是建立如下的 database

資料庫名稱: AccountData

表格名稱:　AccountList

帳號(id) [字串型態]	密碼(pass) [字串型態]	權限(domain) [數字型態]
root	123	11111
alan	qwer	10000
bae	asdf	11000
. . . .	. . . .	. . . .

按照以下的語法逐步輸入即可完成

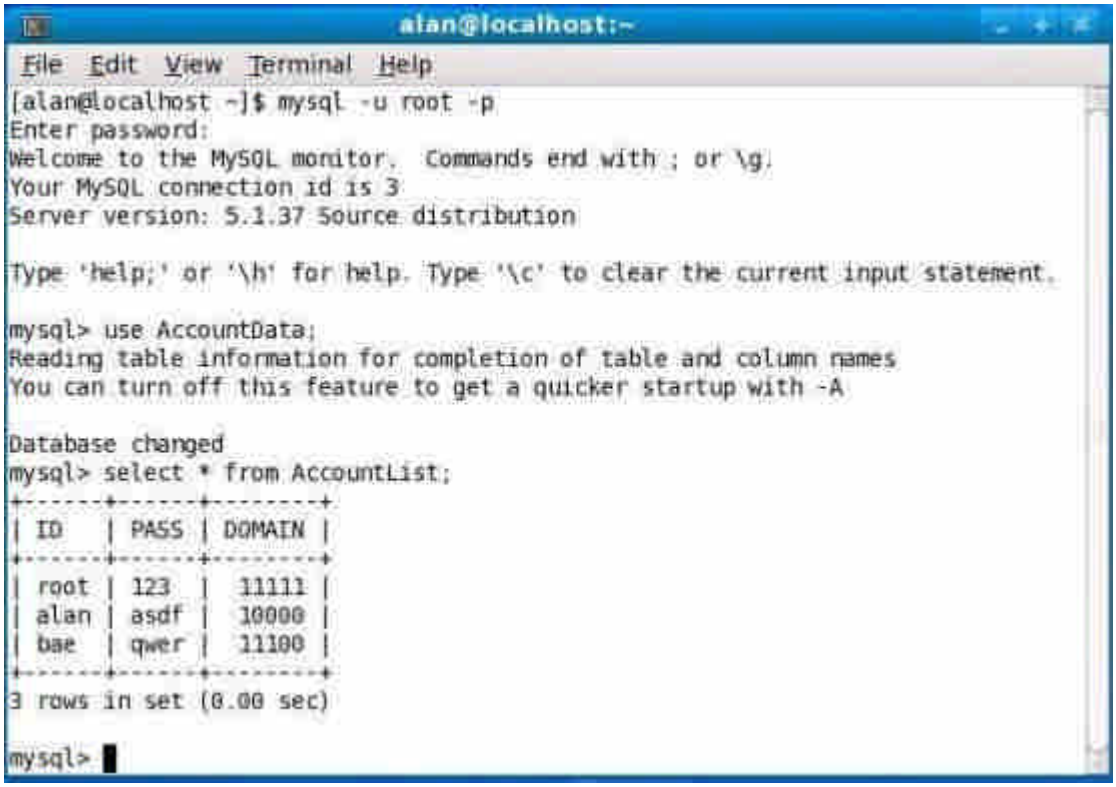
至此 table 已建構完成，接著逐筆輸入資料，我們規定每個欄位都一定要填東西，所以每個變數後面都要加 not null 的參數。

以上三行是一樣的動作，插入資料到 AccountList 的資料表。插入的欄位分別指定內容為 root、123、11111。

輸入完成後可以以下面語法作顯示:

```
select * from AccountList;                    #選擇觀看 AccountList 所有項目
```

完成建構的表格



STEP2 設定 JDBC

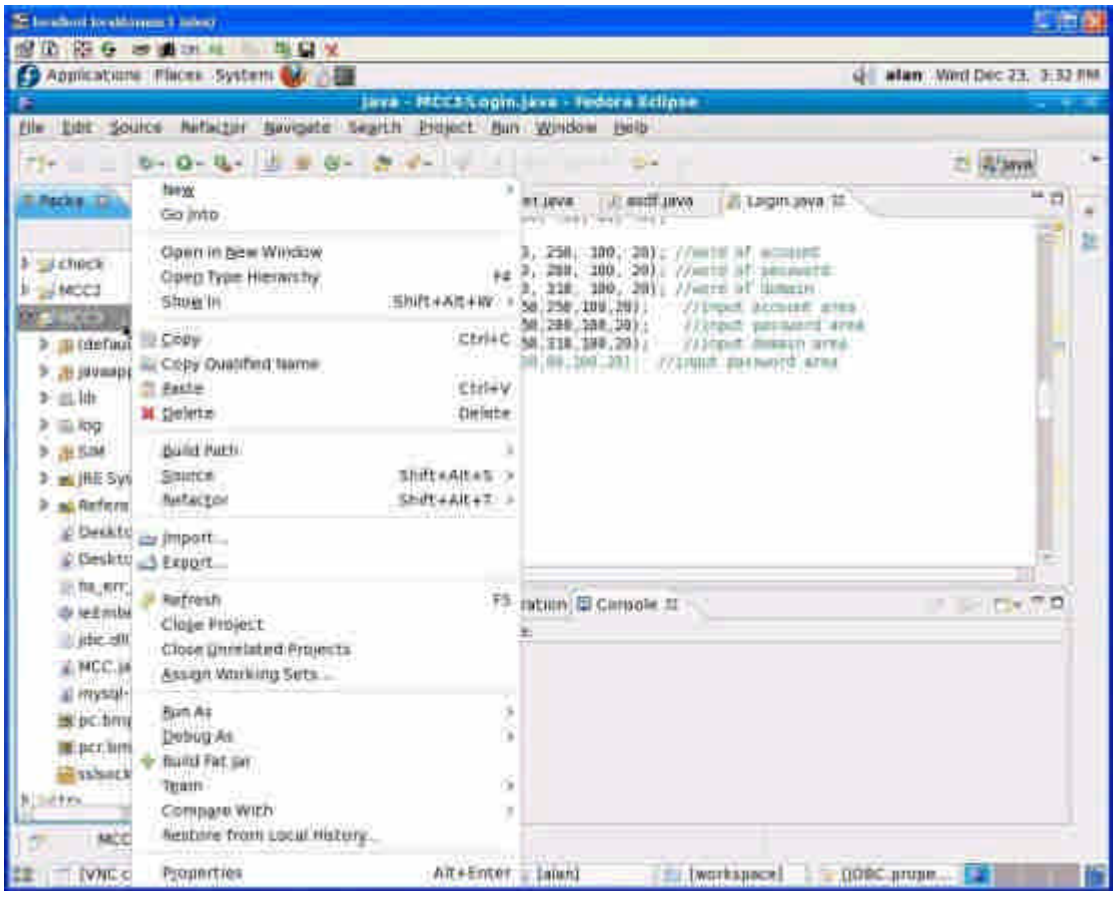
建立完表格後，接著設定 JDBC，主要必須做三件事情。

1. 設定 mysql-connector (java 版)的連接路徑
2. 設定 property 檔
3. 修改 MCC 內的 code

設定 mysql-connector (java 版)的連接路徑

在 MCC 的主封包在那個封包按右鍵，並且選擇選單中最下面的項目， properties。

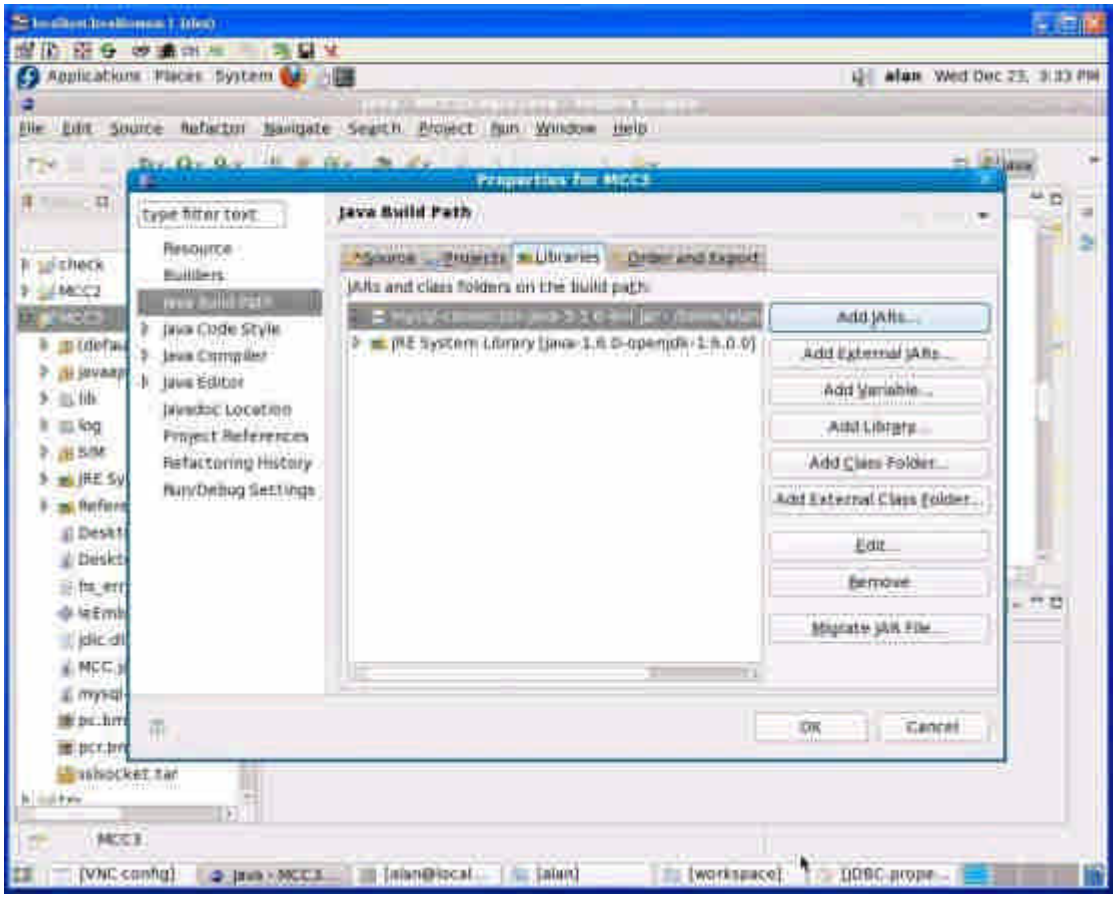
選單圖



左邊 java Build path 加入關鍵的 mysql-connector 放在 MCC3 的封包裡面

MCC_Final 底下的 mysql-connector-java-5.1.6-bin.jar

加入 connector



設定 property 檔

接著是設定在 workspace 下的設定檔。

內容如下

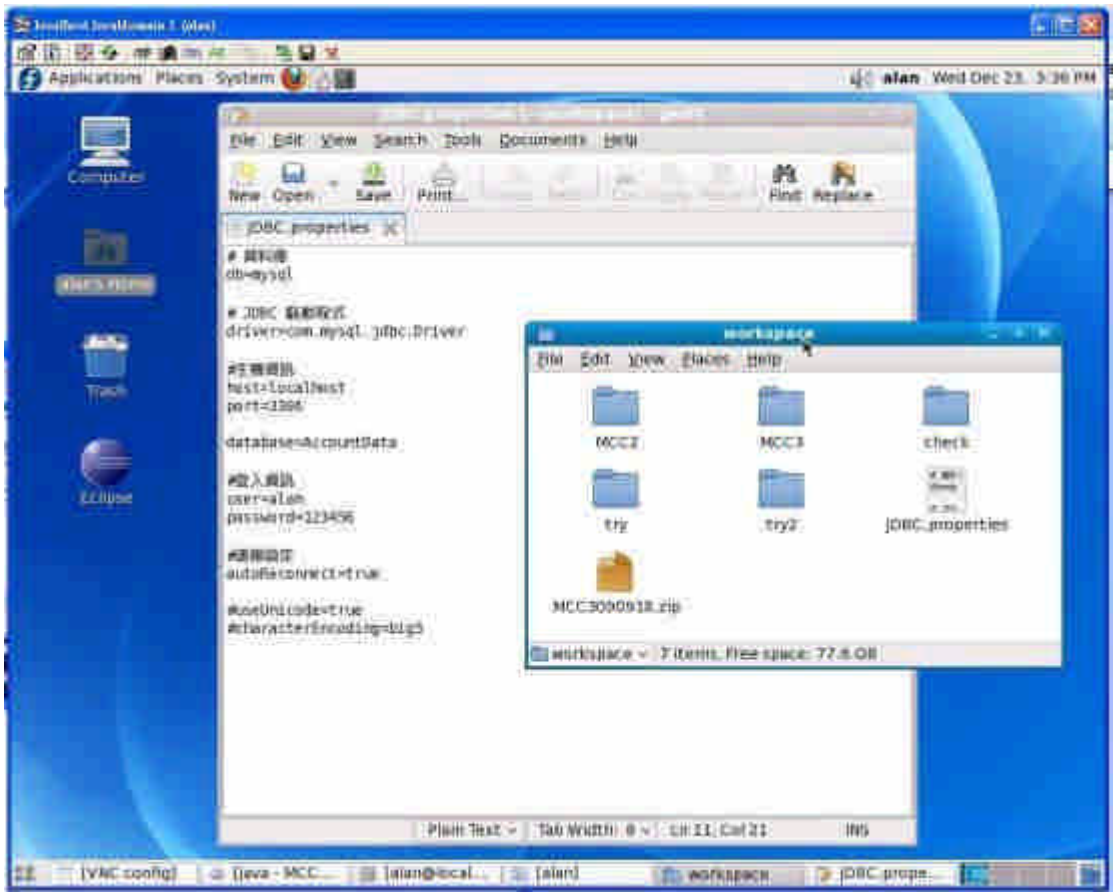
```
# 資料庫
db=mysql                        #程式內所使用的簡稱，不需修改

# JDBC 驅動程式
driver=com.mysql.jdbc.Driver    #程式內所使用的簡稱，不需修改

#主機資訊
host=localhost                  #程式內所使用的簡稱，不需修改
port=3306                       #資料庫的 port，不需修改

database=AccountData            #建置的資料庫名稱

#登入資訊
user=root                       #預設的使用者名稱
password=123456                 #安裝時設定的 root 密碼

#連線設定
autoReconnect=true              #開啟自動重新連線，不需修改

#useUnicode=true                #使用 Unicode，不需修改
```

這份設定檔可以放在使用者所設定的位置，我們預定將此份設定檔放在 workspace 之下，設定檔名稱為 JDBC.properties。

絕對路徑為 /PATH/TO/YOUR/FILE/JDBC.properties

如下圖:

設定檔



修改 MCC 內的 code

根據絕對路徑的不同，MCC 內的 code 有兩個地方要根據當下的絕對路徑作修改。我們必須修改 MCC_Final 中 src 資料夾下的 default packeage 中的 Login.java 與 AccountVertifier.java。

修改 Login.java 的 221 行

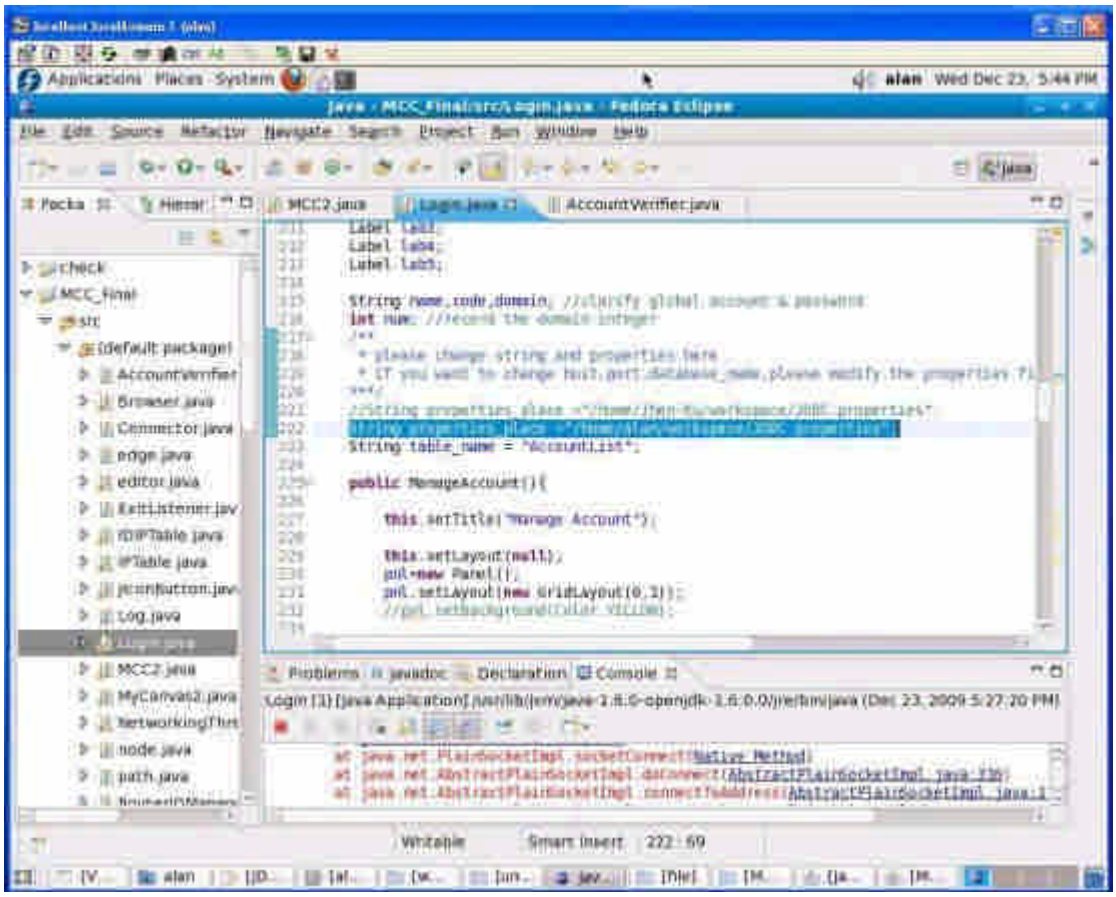
String properties_place ="/home/Jhen-Ru/workspace/JDBC.properties";

改為

String properties_place ="/PATH/TO/YOUR/FILE/ JDBC.properties";

本手冊的範例放設定檔的絕對路徑範例為/home/alan/workspace/JDBC.properties

如下圖:



接著修改 AccountVertifier.java 的第 15 行

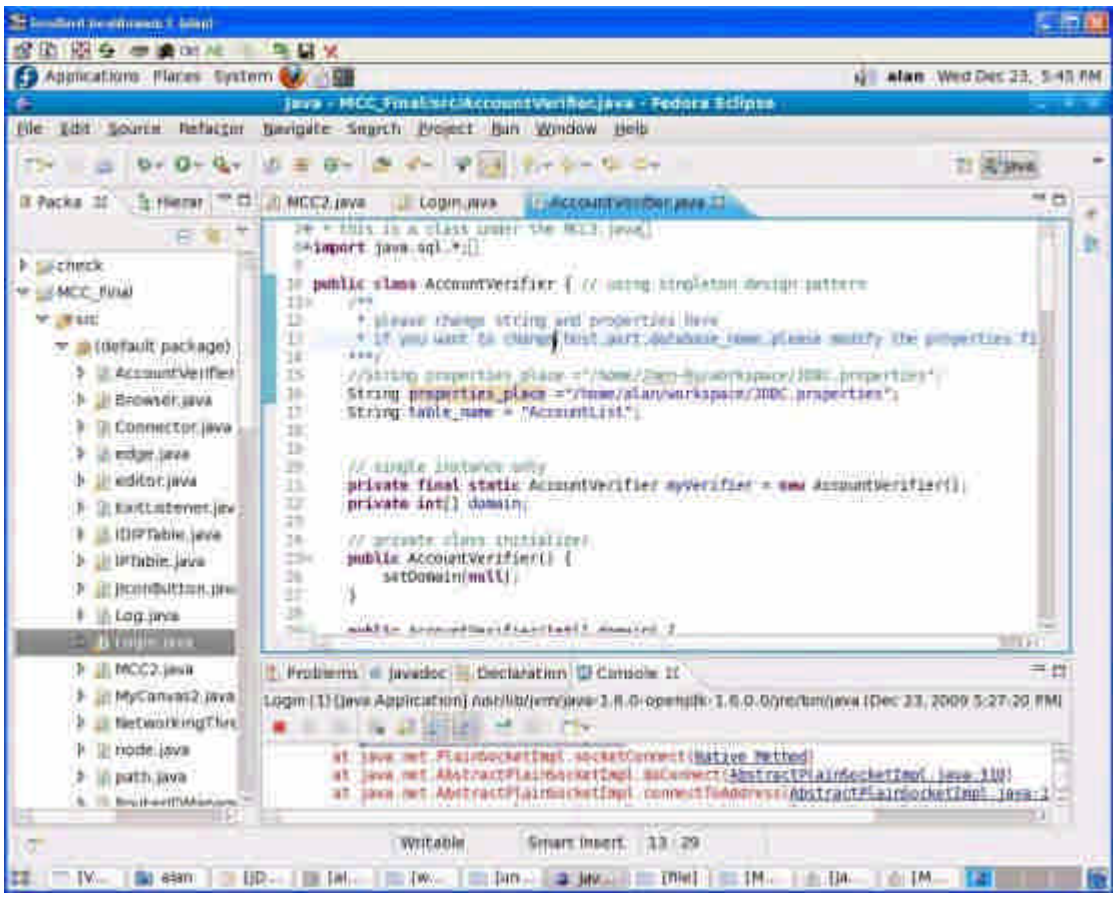
String properties_place ="/home/Jhen-Ru/workspace/JDBC.properties";

改為

String properties_place ="/PATH/TO/YOUR/FILE/ JDBC.properties ";

本手冊的範例放設定檔的絕對路徑範例為/home/alan/workspace/JDBC.properties

如下圖



完成這樣的動作之後，就完成了設定。

附錄五、中央監控系統操作安裝手冊

本操作手冊將介紹 MCC 的安裝、使用。並且指出各子計畫項目的介面。

一、 安裝

把 MCC_Final.tar.gz 複製到欲安裝資料夾下面。

輸入指令

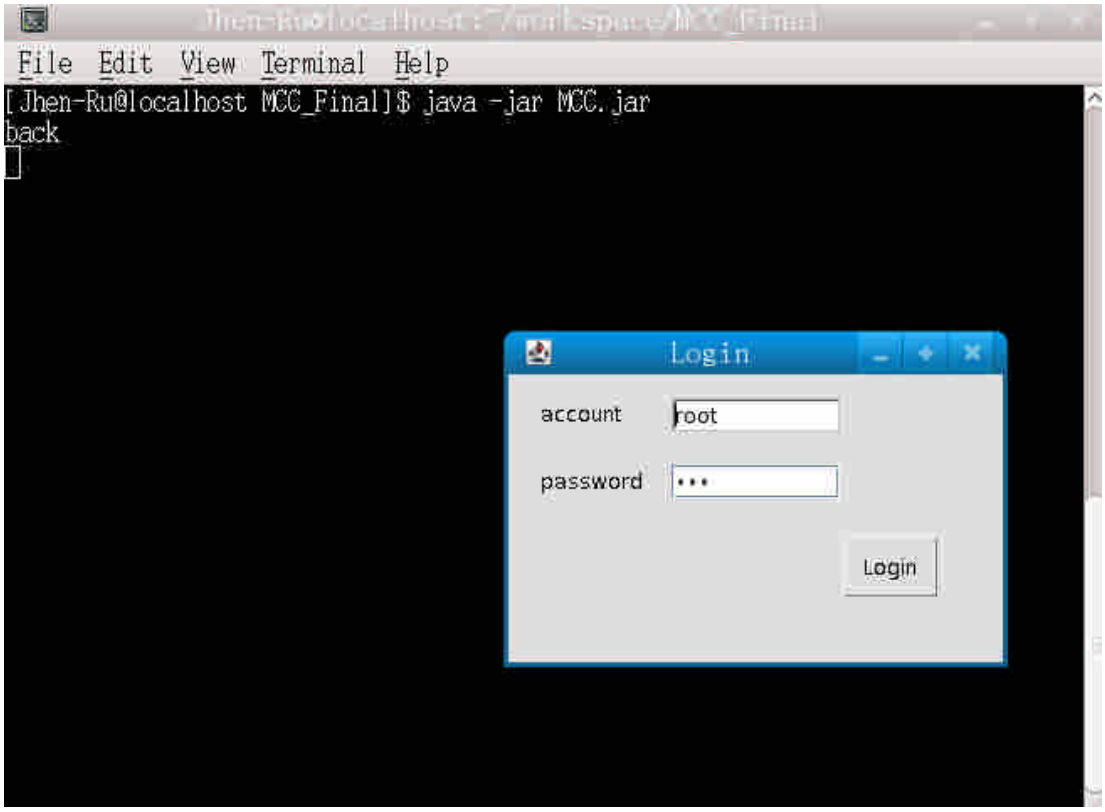
```
tar -vxzf MCC_Final.tar.gz
```

此指令將會把 MCC 系統解壓縮，並且產生 MCC_Final 的資料夾。

二、 執行

進入 MCC_Final 資料夾。

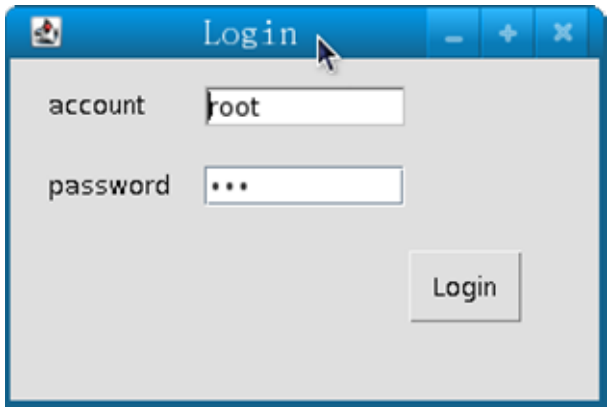
```
cd PATH/TO/YOUR/DIRECTORY/MCC_Final
java -jar MCC.jar
```



執行畫面，將會出現登入視窗。輸入正確帳號密碼之後會依是否有管理帳號權限來判斷。

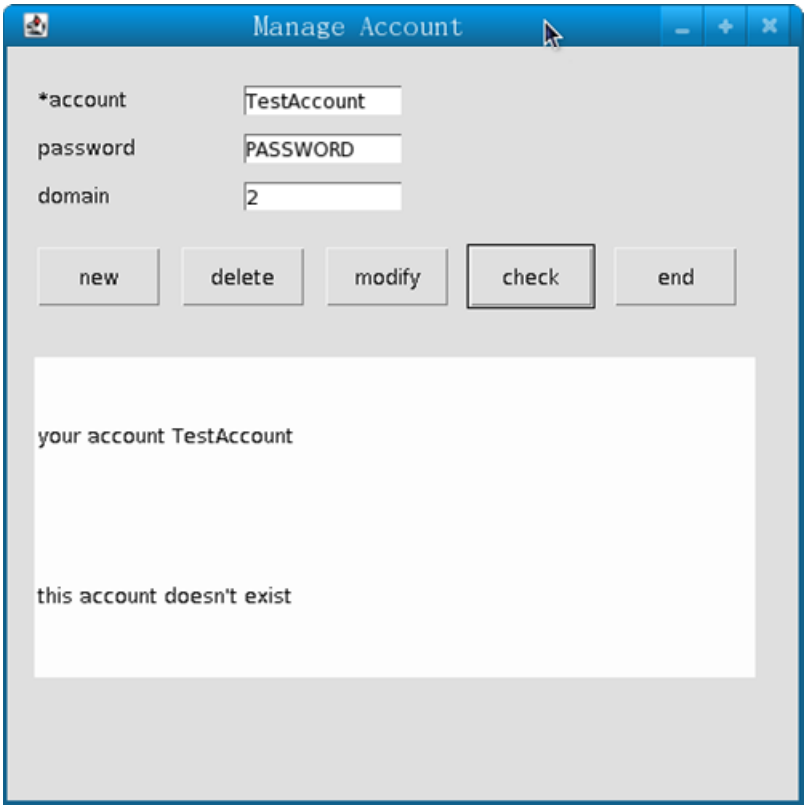
三、 畫面介紹

一剛開始會出現輸入使用者帳號密碼的登入視窗，已確定該使用者是否有使用 MCC 的權限。本截圖為了以後的呈現方面，登入時使用具有管理者權限的帳號 root 登入，密碼暫為 123。

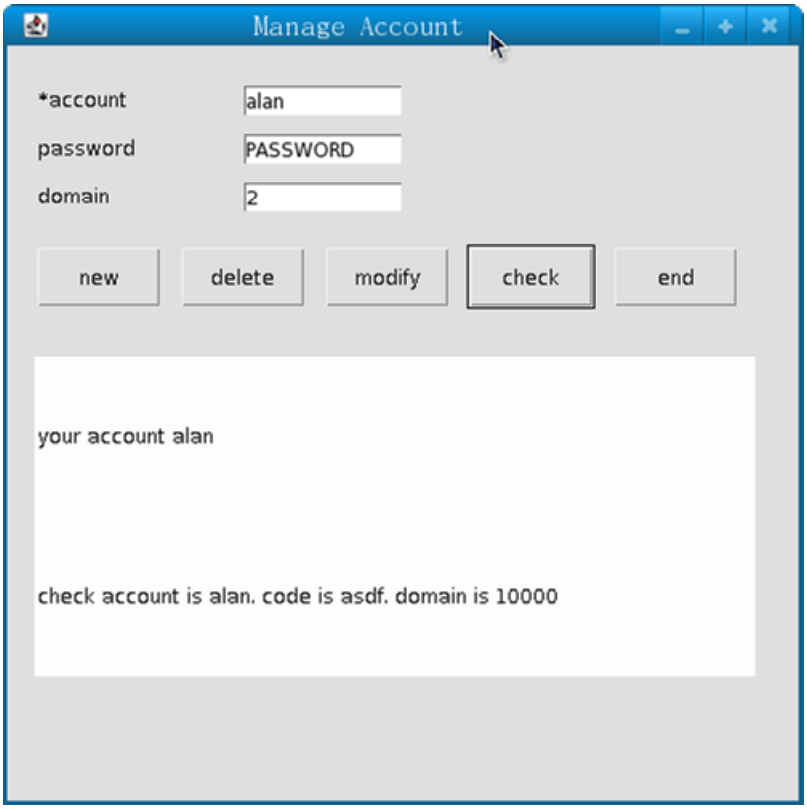


登入畫面，輸入帳號與密碼之後按下 Login，就會判斷該輸入的帳號密碼是否正確。

如果具有管理者權限的話，將會多彈出一個帳號管理者的介面。該介面用來管理可以存取 MCC 的帳號。可以查詢是否存在使用者，或是顯示該使用者的密碼。



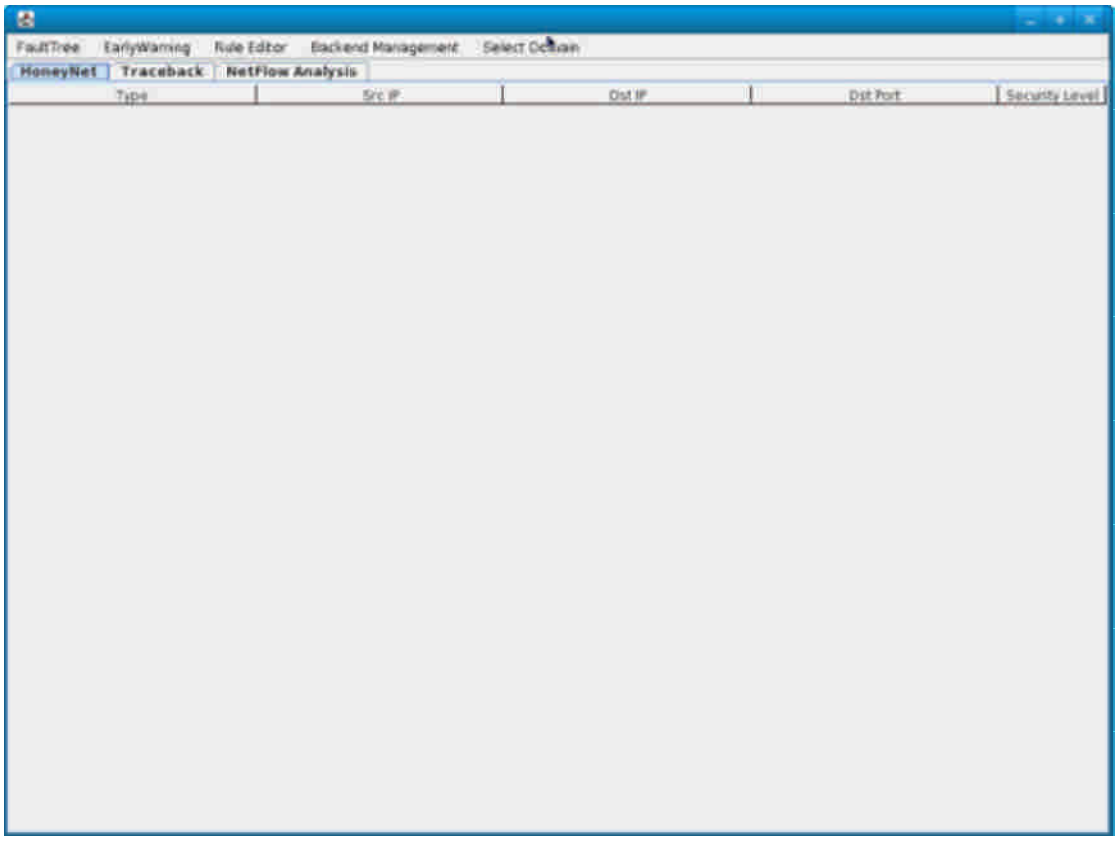
帳號管理介面輸入未存在的帳號，對話欄中會顯示該帳號不存在。



若輸入該帳號有存在的話，可以做權限的修改，存取的 domain 修改，密碼修改。

另外一個介面則是 MCC 主體，剛開始顯示的是 Honeynet 的子畫面視窗。而上面會有一排功能選單。功能選單具有下列幾項：

1 Fault Tree
  1.1 Fault Tree Browser
  1.2 Fault Tree Editor
2 Early Warning
3 Rule Editor
4 Backend Management
5 Select Domain
  5.1 Domain 1
  5.2 Domain 2
  5.3 Domain 3
  5.4 Domain 4
  5.5 Domain 5

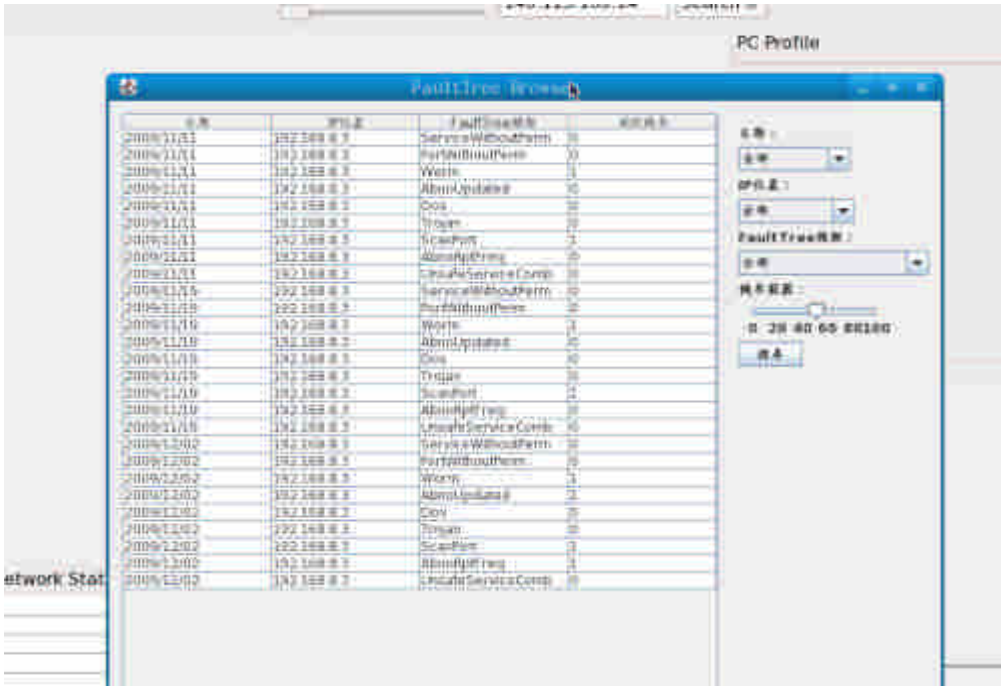


MCC 初始介面為 Honeynet 整合介面，該介面會顯示從 Honeynet 主機傳來的攻擊狀況。

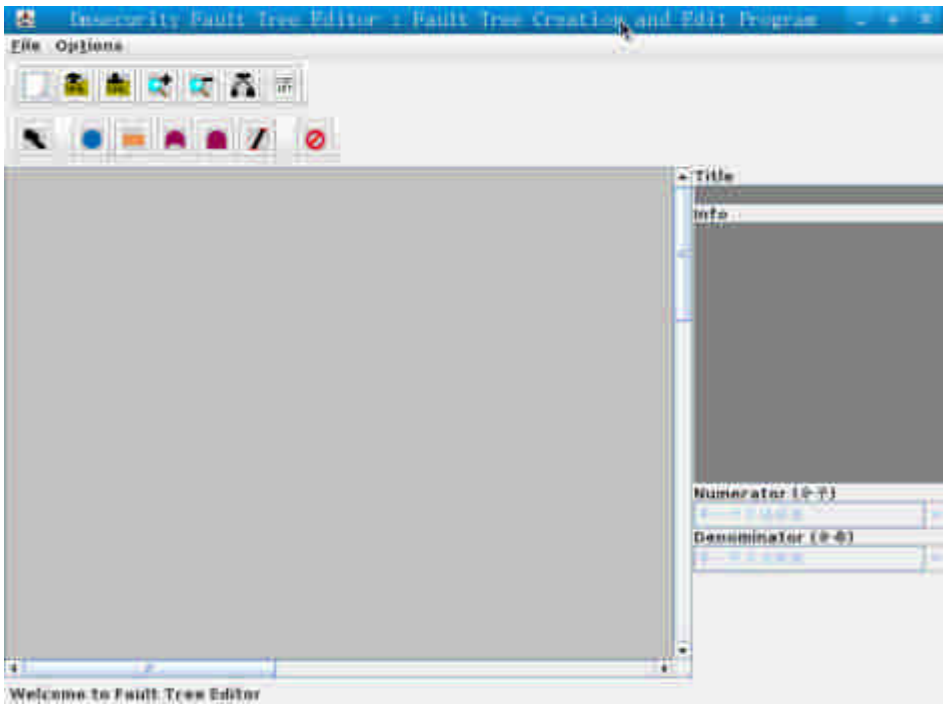
接下來逐一介紹。

1. Fault Tree

在功能選單的最左邊點選之後，會有兩個功能。分別是呼叫 FaultTreeBrowser 的介面和 FaultTreeEditor。



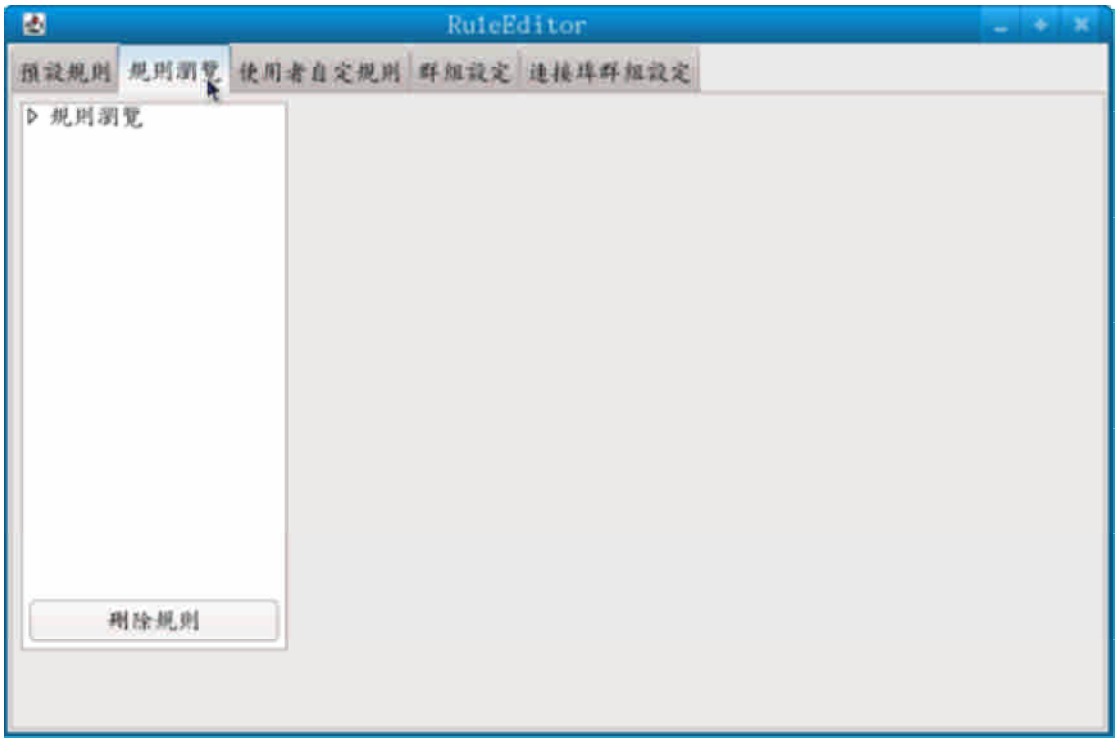
Fault Tree Browser 的介面



Fault Tree Editor 的介面。

2. Early Warning 預警系統
   點選之後將會出現預警系統的介面。
3. Rule Editor
   點選之後將會出現 Rule Editor 的主畫面。

注意：必須先至 Select Domain 功能選單中，點選一個欲修改的 Domain。



Rule Editor 出現的介面。

4. Backend Management
   會出現 Backend Management 的介面。
5. Select Domain
   點選之後會彈出五個 Domain 可以點選，此部分為單一選擇，點選新的之後將會把直傳給 Rule Editor，若無修改權限，則無法點選。

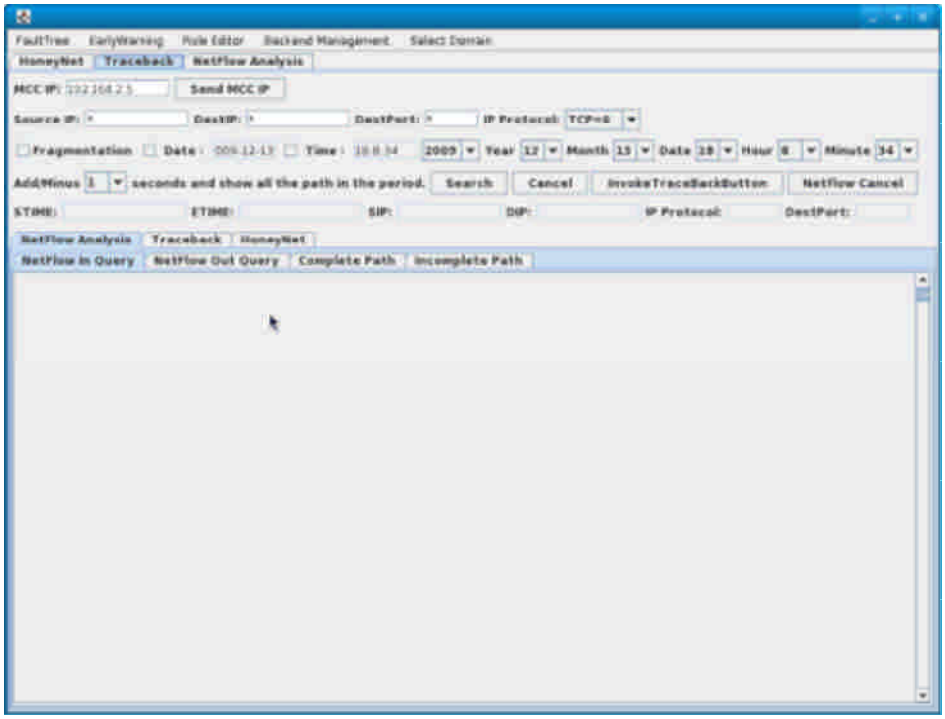
接下來將會介紹三個子計畫的畫面。(一)Honeynet 攻擊狀況介面、(二)Traceback 回追系統介面、(三)網路流量狀態介面。

(一) Honeynet 攻擊狀況介面

該介面將會顯示 Honeynet 被攻擊之後回傳的攻擊來源 IP、目標 IP、安全層級等資訊。每一筆資料將會成為新的一行，而最下方可以按下清除鍵，清除之前的紀錄。

(二) Traceback 回追系統介面

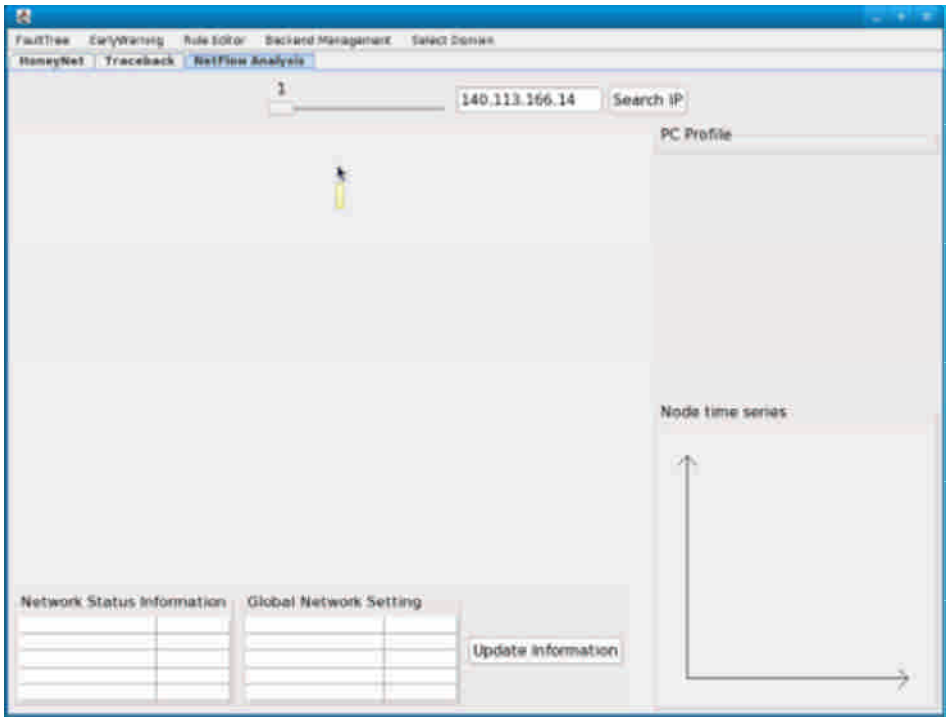
該回追系統最左上方要輸入 MCC 的 IP，彈性的指定 MCC 的 IP。按下”send IP” 方能使用 MCC 的回追功能。詳細的 Traceback 用法，將會在子計畫裡詳細的介紹，此操作手冊暫不贅述。



Traceback 的顯示介面。

(三) 網路流量狀態介面

此介面用來 Rule Editor 追蹤網路狀態中顯示節點的畫面。在 Rule Editor 中可以顯示拓撲，而此拓樸將出現在該介面中。而點選該介面中的節點，即可顯示該節點的網路狀態，並且會自動作回追該 IP 的狀態。



Network flow 顯示畫面。

附錄六、誘捕網系統設計報告書

系統綜觀

隨著 Web 攻擊行為的日增成熟，要去學習如何防範一直困擾著系統管理人員；同時，要如何去從攻擊中找出犯罪證據更是關鍵挑戰之一。現今資訊安全防護策略大都注重在已知的威脅上，根本無法因應未知攻擊，在阻擋入侵的同時，也無法真正偵測出入侵者的面貌。如今，隨著資訊科技的進步，開啟運用另一種思維來對抗，將以往被動的防禦，轉為積極學習。這項方法是所謂的誘捕系統(Honeypot)。

本研究計畫建立一個基於 web 的 honeypot，用來吸引入侵者，使他們進入受控的環境之中，並使用各種監控技術來捕獲入侵者的行為。功能上，著重於針對網頁型態下之 SQL injection 的攻擊紀錄，將相關的手法以及流程，入侵模式做完整的 log。

參考文件

[1] “SQL injection – Wikipedia, the free encyclopedia”

[2] "E.1.7. Changes in MySQL 5.0.22 (24 May 2006)". MySQL AB. 2006-05-04. http://dev.mysql.com/doc/refman/5.0/en/news-5-0-22.html. Retrieved on 2008-05-16. , "An SQL-injection security hole has been found in multi-byte encoding processing", retrieved March 20, 2008

[3] “Know Your Enemy: Sebek - A kernel based data capture tool” – the honeynet project, last modified: 17 november 2003

[4] SQL Injection (資料隱碼)– 駭客的 SQL 填空遊戲(上)(下)

[5] MySQL SQL Injection Cheat Sheet

[6] Technical Information on Encoding-Based SQL Injection Exploit

[7] 結合入侵偵測和蜜罐之分散式預警系統的設計與實現

[8] 網際網路惡意網站偵測機制之研究

[9] Honeynet Data Analysis: A technique for correlating sebek and network data

電腦軟體構型項目層面設計決策

電腦軟體發展環境選用決策分析

本計畫程式使用的開發語言為 php。選用的原因是因為 php 可以跨平台，而且開放原始碼(PHP 本身就開放原始碼)。此外，很多開放原始碼的程式，也都是使用 php 所撰寫，所以使用 php 建構網站可以節省很多開發時間。

開發完成的軟體預計安裝在 Windows XP 作業系統。本計畫的宗旨是建立 Windows 下的 Web Honeypot 誘捕網，而 XP 是市佔率最高也最普及的作業系統，因此其為最優先的考慮。

資料庫軟體使用 MySQL。原因是 MySQL 免費，它的網路承載比較少，也經過高度最佳化（Highly Optimized）。此外，應用程式透過 MySQL 做備份比起其他類型的資料庫軟體還要簡單許多。MySQL 更為各種不同的資料格式提供提供彈性的介面，好學而且操作簡單。

在本計畫中，資料監測的軟體選用 sebek。 Sebek 用來記錄蜜罐裡入侵者行為的工具，可分為客戶端和伺服器端。客戶端藉著精巧地設計隱藏於蜜罐核心(Kernel)之中，暗中蒐集蜜罐上的攻擊行為資料(執行程序、對外連線、存取系統資源等)，再將蒐集到的資料傳送至伺服器端進行資料記錄。藉由此種架構，在攻擊者入侵以後不易被察覺。此外 sebek 為一免費的資料監測軟體，容易取得與安裝。

電腦軟體構型項目行為設計決策分析

本節說明本系統將開發及為何開發哪些套件(如 MCC、SIM…)、功能模組(如資料來源截取模組)、資料庫等，提供表單 list 出每個套件所包含的功能模組，以及整套系統會有哪些 database。

I. 含有 SQL injection 弱點的 Vulnerable web application

II. Vulnerable database

以資訊安全討論區的形式呈現，為一個具有各種功能，完整的網頁介面程式。另外，在登入處理的頁面中設有 SQL injection 的弱點，引誘攻擊者入侵，藉

以記錄其所做的行為，並在 Vulnerable Database 中提供網站存取帳戶之資料，以誘騙攻擊者。

III. Entrap unit

這個部份是用來做使用者 input 的檢查，server response 的檢查以及 security level 的模式設定與切換的實作。基本上，entrap unit 是與具有漏洞的頁面綁定在一起的。若有攻擊者用 sql injection 的方式，成功執行惡意的 SQL 指令，則經過使用者 input 的檢查和 server response 的檢查，偵測到此一惡意行為，我們會將 security level 的等級調升。

IV. Manage unit

最主要的管理介面。這部份的設計為因應可以做 remote control，從遠端能夠調整 security level。

V. Record unit

為了記錄攻擊者藉由 SQL injection 入侵的所有行為，需要一個記錄的機制，來完成記錄行為的動作，這個部分會將這些行為記錄下來存到 log database 中，且基於安全的考量，log database 須和 web application 所使用的 database 分開放置。

VI. Log database

記錄攻擊者行為模式的資料庫。

VII. Sebek

Sebek 主要是用來記錄蜜罐裡入侵者行為的工具，可分為客戶端和伺服器端。客戶端藉著精巧地設計隱藏於蜜罐核心(Kernel)之中，暗中蒐集蜜罐上的攻擊行為資料(執行程序、對外連線、存取系統資源等)，再將蒐集到的資料傳送至伺服器端進行資料記錄。Sebek 是一個現有核心層級 rootkit，引入此一工具是為了防範受到非 SQL injection 的攻擊，使得 honeypot 被入侵，如此一來，我們在這個時間點之後所記錄的資料可能都是不可信的，為了找出這個時間點，以及記錄其行為模式，我們引入使用 sebek。

電腦軟體構型項目架構設計

電腦軟體構型項目組件

I. 含有 SQL injection 弱點的 Vulnerable web application

article.php
board.php

II. Vulnerable database

Vul-DB

III. Entrap unit

index.php
login.php

IV. Manage unit

lib.php
log_list.php
log_search_form.php
Blist.php

V. Record unit

log.php

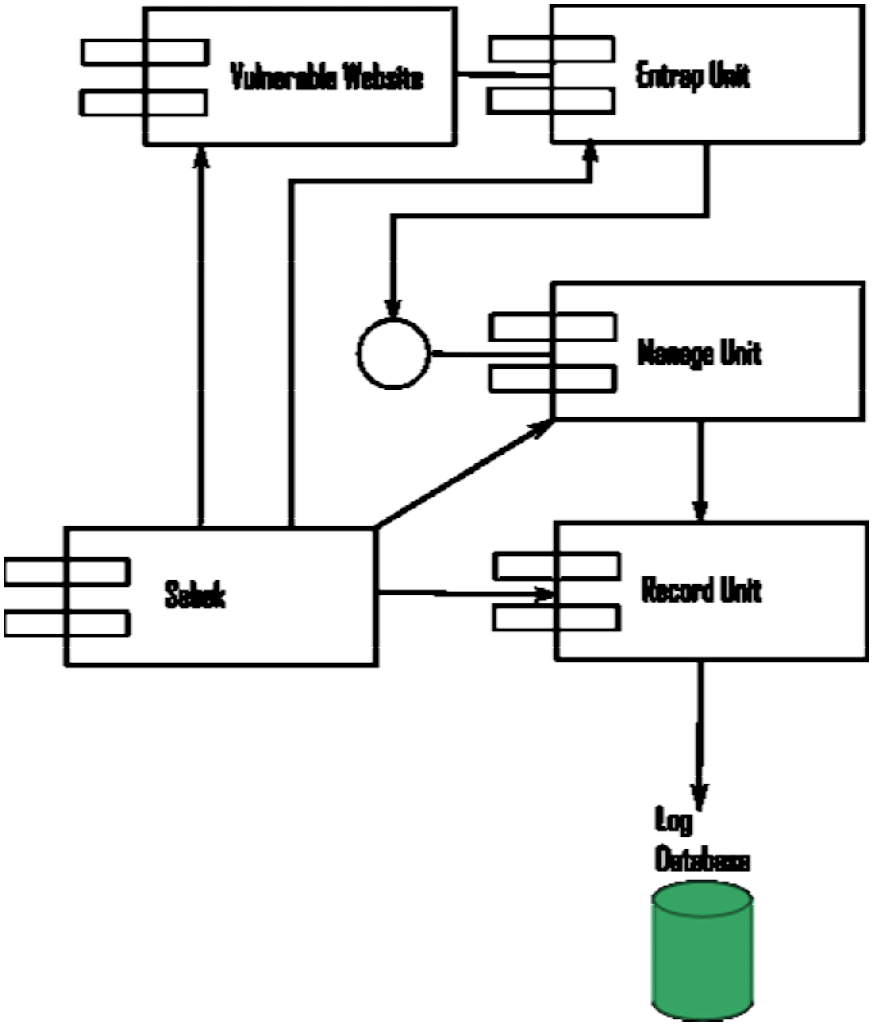
VI. Log database

Log database 記錄的欄位如下：

I. Ip
II. Seclv
III. Time
IV. Page
V. Acc
VI. Pwd
VII. Sqlstr
VIII. Sebek

各部分元件以及說明請參照 Sebek 官方網頁，上面具有完整的開放原始碼，以及其使用手冊。

元件圖(Component Diagram)：



執行的概念

I. 含有 SQL injection 弱點的 Vulnerable web application：所有程式的功能均相同，用以模擬出普通的網頁與伺服器，供使用者使用。此外，資料庫的部分則是為了讓攻擊者取出資料用。

article.php

board.php

II. Vulnerable database

Vul-DB

III. Entrap unit：廣義來說 index.php 也是含有 SQL injection 弱點的 web application。此兩者程式是用來誘捕來自 web 頁面的攻擊，判定其是否為攻擊行為，若是攻擊行為則進行紀錄以及捕捉。

index.php：為本 web honeypot 的首頁

login.php：用於使用者登入

IV. Manage unit

lib.php：主要用於調整 security level，以其與其相對應的安全措施跟過濾機制。此外並包含了一些資料庫連結，還有其他功能程式。

log_list.php：列出 log 紀錄，為一完整的管理介面。包含了 log_search_form.php。

log_search_form.php：查詢 log 紀錄。

Blist.php：Black list(黑名單)查詢與管理的頁面。之前所捕捉到具備惡意行為的主機 ip addressa 清單均可在這裡取得。

V. Record unit

log.php：將取得惡意行為紀錄的 log 送到 log database 中。

VI. Log database

Log database 各欄位的功能如下：

I. Ip – 來源的 IP

II. Seclv – 目前所在的 security level

III. Time – 事件發生時的時間

IV. Page – 事件發生時所在的頁面

V. Acc – 使用者輸入的 account

VI. Pwd – 使用者輸入的 password

VII. Sqlstr – 送到 MySQL server 執行的 SQL 查詢字串

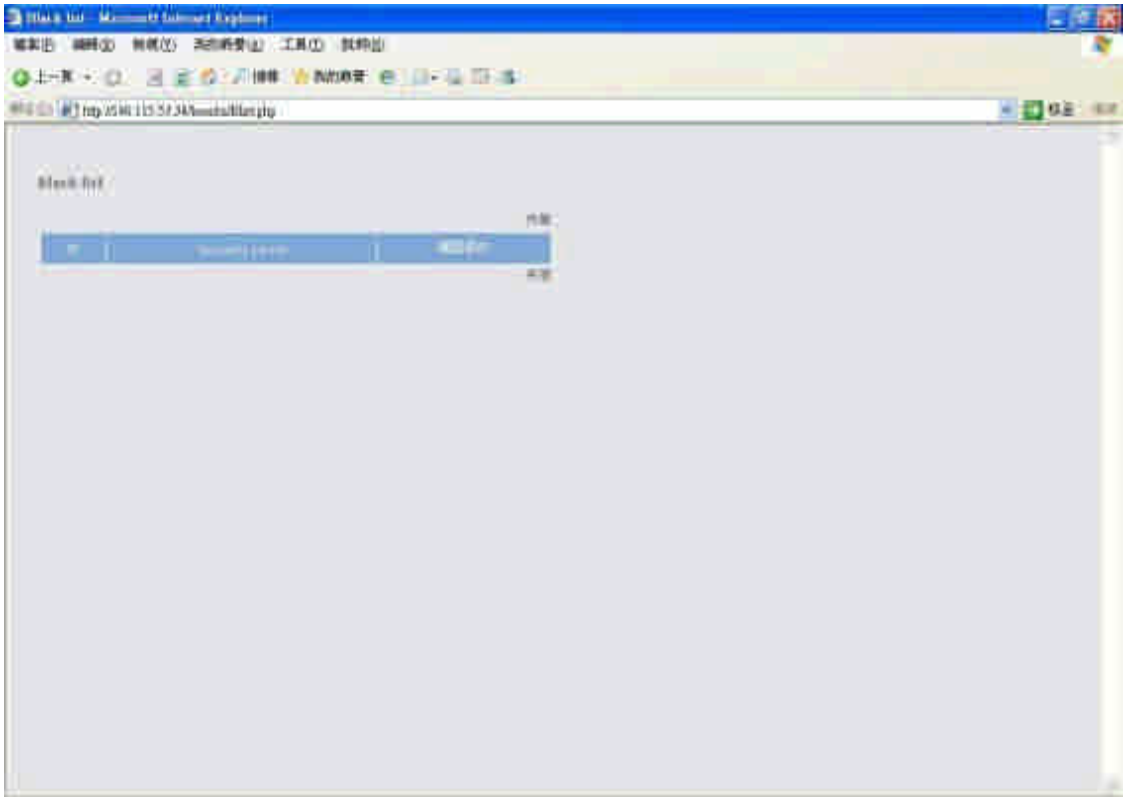
VIII.

介面設計

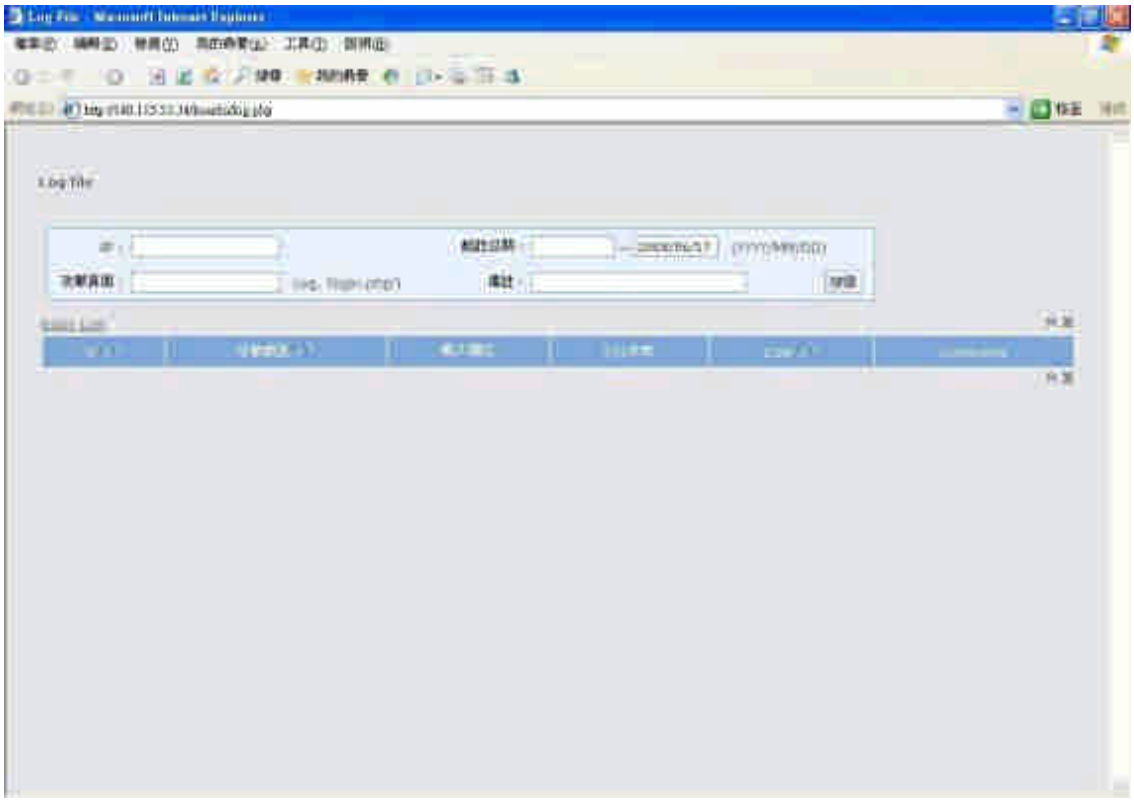
1.首頁：web honeypot 的首頁。使用者可以在此進行登入，或是由左方的連結進入討論區後登入。

2.討論區首頁: 使用者在此可以登入，並在登入後進行各項留言版的操作。

3. Black list: 管理介面，紀錄攻擊來源 ip，並可以針對此 ip 調整 security level 值。



4. Log file： 管理介面，用以查詢 log 檔案。



電腦軟體構型項目細部設計

物件導向分析(OOA)

正常使用者操作案例

內容描述

使用者由首頁的登入頁面，輸入正確的帳號以及密碼，正常的進行登入。

功能與性能需求流程(Flow of Events)

使用者在登入頁面中輸入帳號密碼，而登入頁面會產生相對應的 SQL Request 並送到 Entrap Unit。Entrap Unit 判定為合法，則直接將此 SQL Query 送至網站之資料庫，並將帳號密碼與資料庫中帳號資料作比對。比對結果正確，找到了相符合的資料，使用者在驗證過後將會獲得可存取與操作留言版的權限。

例外處置需求流程（Alternative Flow）

使用者的帳號與密碼跟資料庫中的資料不符，而使用者輸入字串並非攻擊字串，則系統回應"帳號，密碼有誤"。

特殊需求（Special Requirements）

無

前置條件（Pre-Conditions）

無

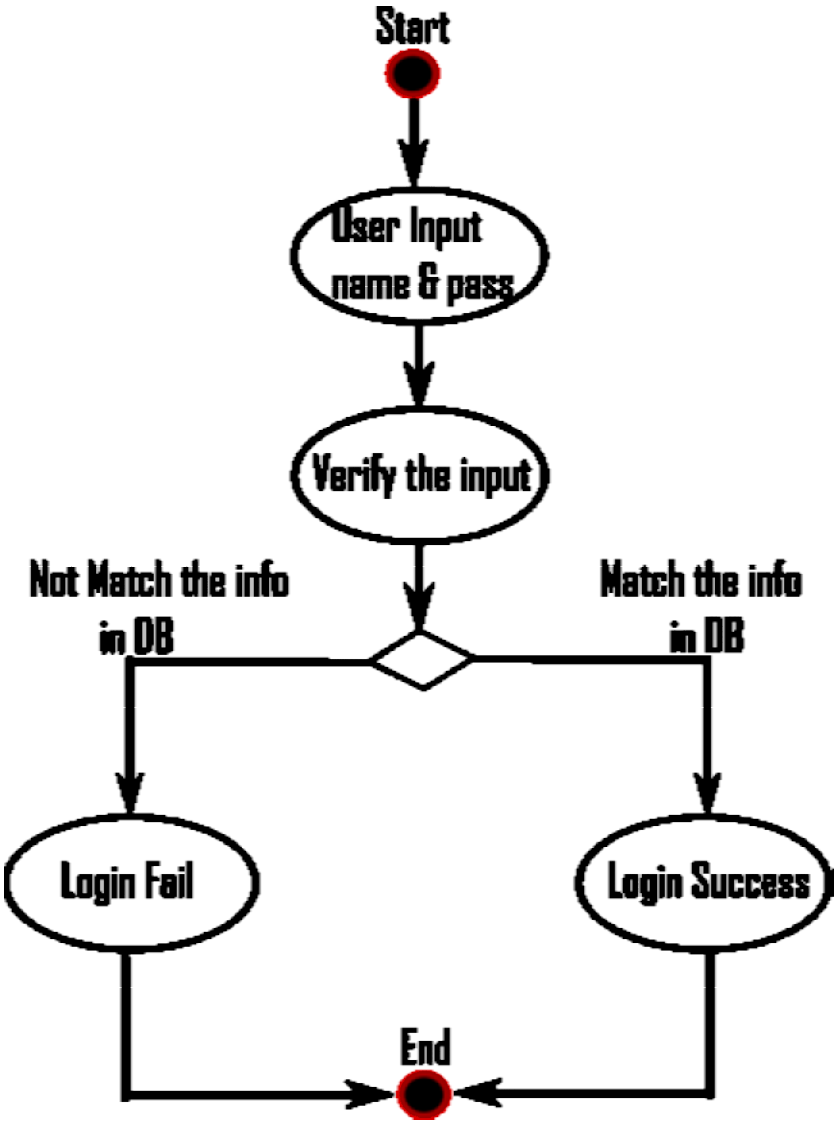
後置條件（Post-Conditions）

無

延伸點（Extension Points）

無

活動圖(Activity Diagram)



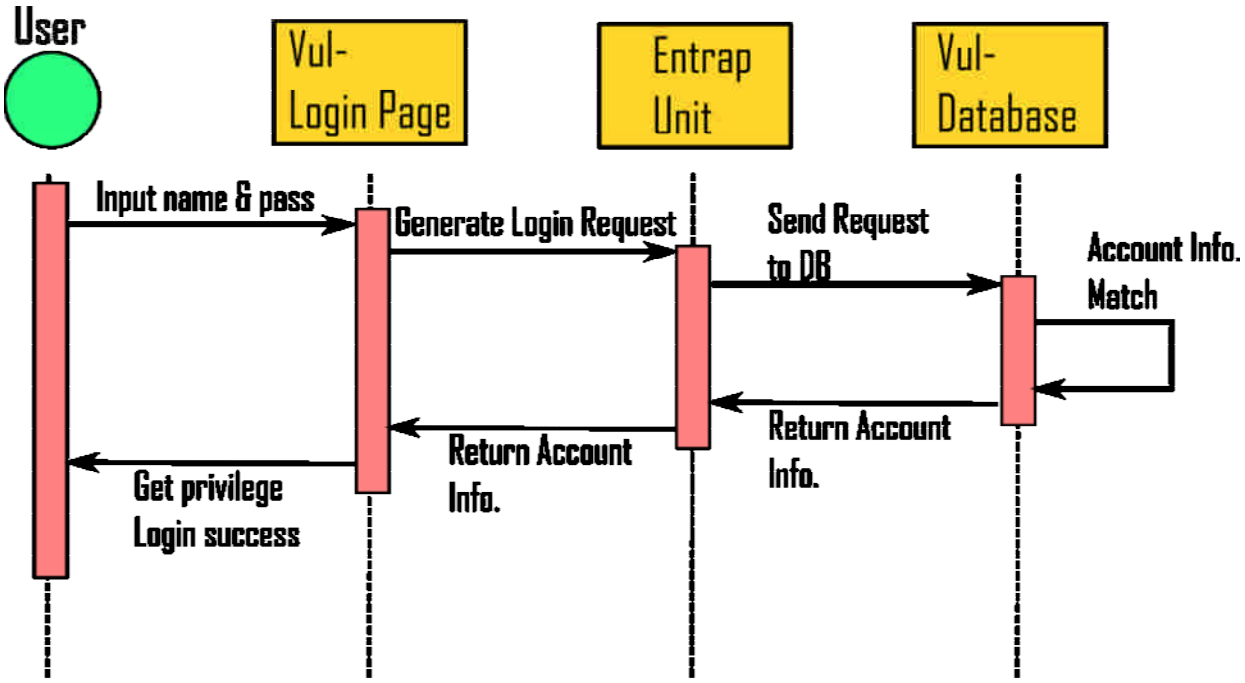
物件導向分析(Object Oriented Analysis)

主要用來描述這個使用個案會使用到了哪些程式及資料表，依序畫出循序圖、合作圖及類別圖。如下範例

為了讓操作手可以點選目標，設計了一個邊界類別(Boundary Class)RadarPanel來顯示目標和接收操作手的滑鼠控制。

為了可以儲存目標以供搜尋，設計了一個實體類別(Entity class)目標串列。

循序圖(Sequence Diagram)



類別圖(Class Diagram)

攻擊者操作案例

內容描述

使用者在首頁，進行登入時對主機進行攻擊成功。

功能與性能需求流程(Flow of Events)

使用者在登入頁面中輸入攻擊字串，而登入頁面產生相對應的 SQL Request 並送到 Entrap Unit。Entrap Unit 判定為非法的攻擊，則不僅將此 SQL Query 送至 Record Unit，同時也將此 SQL Query 送到網站之資料庫。Record Unit 會把攻擊字串送到 Log Database。

而網站之資料庫在執行完 SQL Query 後會將結果回傳給 Record Unit，Record Unit 再將此傳回資料與攻擊字串作比對，合成同一筆 log 放入獨立的 log database 中。

資料庫執行完 SQL Query 時也會同步把結果送回 Entrap Unit，再由 Entrap Unit 依序回送給攻擊者。最後攻擊者取得相對應的權限或是資料。

例外處置需求流程（Alternative Flow）

使用者在登入頁面中輸入攻擊字串，但是結果攻擊失敗，並未成功取得登入的權限。但是此時 Entrap Unit 仍然會紀錄攻擊事件與攻擊字串，並產生相對應的 log。但是此時沒有攻擊對資料庫產生效用的 log 紀錄。

特殊需求（Special Requirements）

無

前置條件（Pre-Conditions）

無

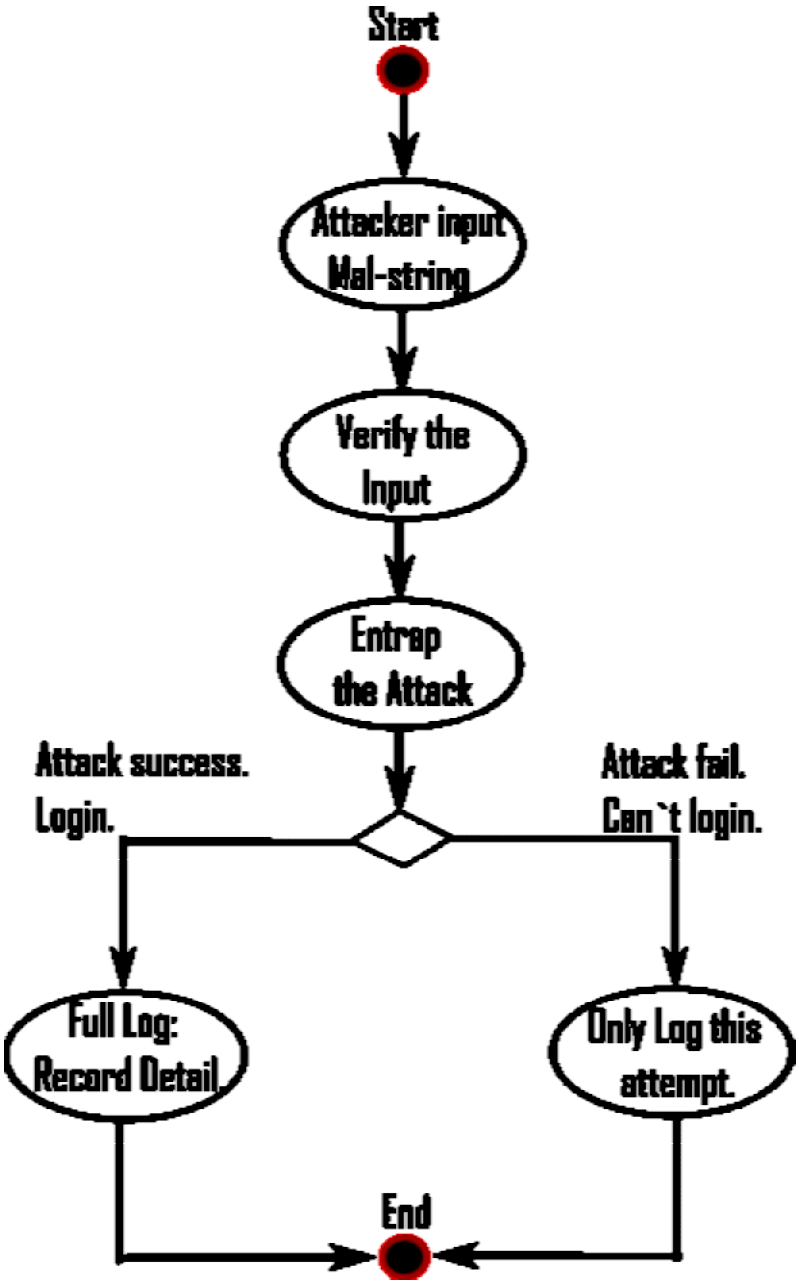
後置條件（Post-Conditions）

無

延伸點（Extension Points）

無

活動圖(Activity Diagram)



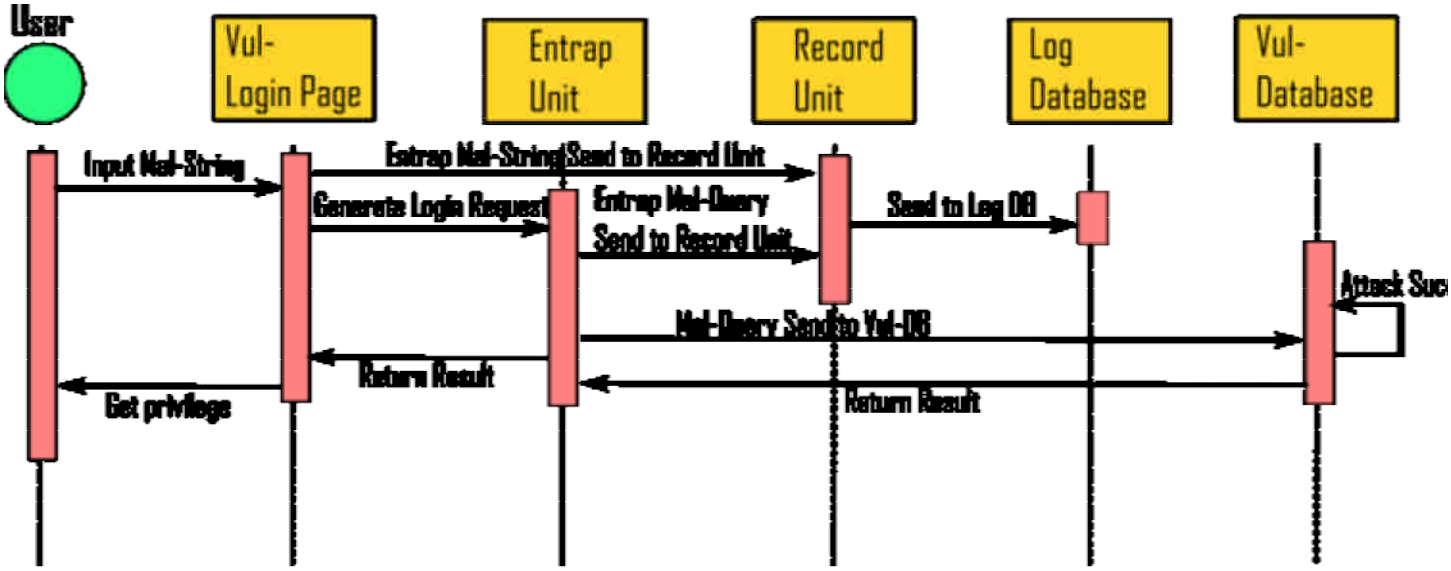
物件導向分析(Object Oriented Analysis)

主要用來描述這個使用個案會使用到了哪些程式及資料表

為了讓操作手可以點選目標，設計了一個邊界類別(Boundary Class)RadarPanel 來顯示目標和接收操作手的滑鼠控制。

為了可以儲存目標以供搜尋，設計了一個實體類別(Entity class)目標串列。

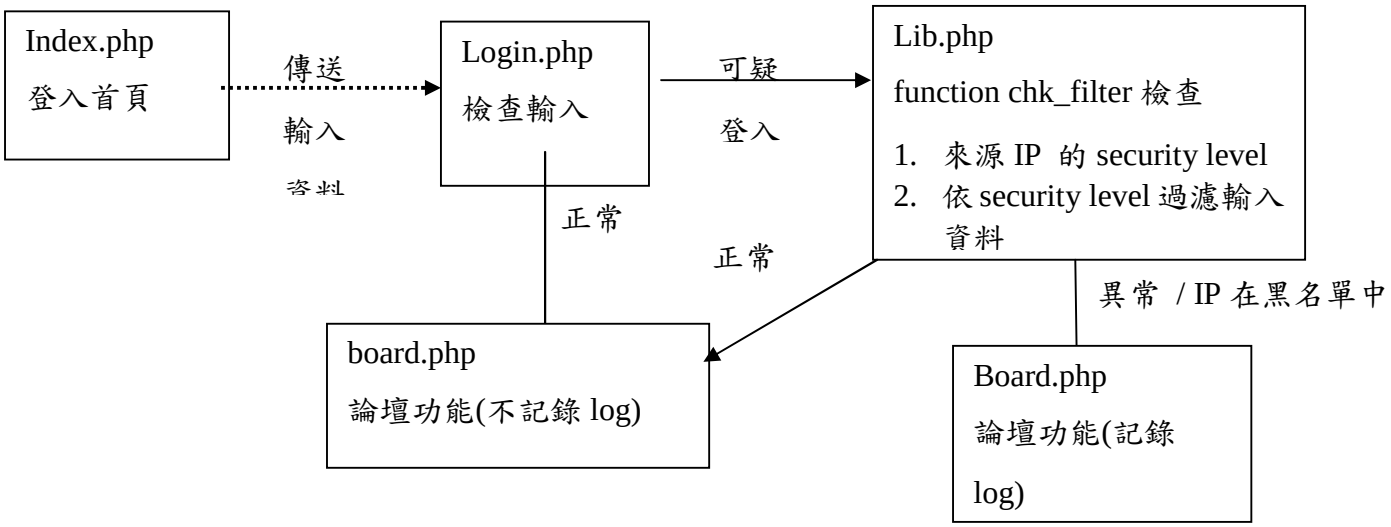
循序圖(Sequence Diagram)



類別圖(Class Diagram)

物件導向設計(OOD)

類別設計(Class Disign)



black_list: 記錄惡意來源的 IP

Field	comment
id	Primary key
ip	攻擊者的來源 IP
sec_level	此 IP 的安全等級: 0~4 (0:最低, 4:最高)

attacker_log: 每次攻擊的記錄檔

Field	comment
id	Primary key
ip	攻擊者的來源 IP
sec_level	此 IP 的安全等級 : 0~4 (0:最低, 4:最高)
time	Log 產生的時間
sql_string	使用者輸入與程式中的 SQL，結合出完整執行 SQL 字串
attack_page	受到攻擊的網頁
log_comments	預留的 log 欄位

變數、函式資料定義與說明

部分程式尚在開發中，因此沒有完整的變數定義以及函式用途。

1. login.hp ：程式專門處理登入的輸入字串，確認這個來源是否已登入過，已入登過則不處理；未登入，會將字串傳至 function chk_filter(lib.php)作進一步的檢查。

2. lib.php ：放公用函式、定義的變數

函式說明：

Function name	Input valuses (type field_name : comments)	Functionality	Reference programs
chk_mal_login	I. String $input_acc ：使用者輸入的帳號 II. Sring $input_pw ：使用者輸入的密碼 III. String $sql_acc ： SQL Query 的帳號 IV. String $sql_pw ： SQL Query 的密碼	比對使用者輸入，與 SQL 搜尋結果是否吻合，若是不同即可判斷此次登入為非法，並記錄相關資訊。	Login.php
chk_filter	HTTP_POST_variables &$_POST : an array of variables passed to the current script via the HTTP POST method.	檢查來源 IP 是否黑名單，若有 log 記錄，提昇其 security level	Login.php
get_magic_quotes_gpc	None	檢查目前系統是否已經設定處理輸入的特殊字元，若以處理則不執行下列的 addslashes_function()，避免重複過濾	Lib.php

addslashes_function	HTTP_POST_variables &$_POST : Array of variables passed to the current script via the HTTP POST method.	Security level 2：將輸入資料中的特殊字元(e.g.：『”』、『'』等)，加上『\』變為『\”』、『\'』等，因此可區分出是程式正常使用的特殊字元，或是使用者的輸入符號。	Lib.php
ini_set	I. String $varname : “display_errors” II. String $newvalue : false	Security level 3：關掉顯示錯誤訊息 ($newvalue = true 將會顯示錯誤訊息)	Lib.php
char_strip	HTTP_POST_variables &$_POST : an array of variables passed to the current script via the HTTP POST method.	Security level 4：將所有輸入內容的符號字元均過濾掉，僅留 a-z, A-Z, 0-9 等字母、數字	Lib.php

定義變數說明:

Variable Name	Functionality	Reference Program
$link	建立 Script 程式與的 MySQL 連線	article.php、Blist.php、board.php、index.php、lib.php、log.php、login.php
WEBURL	誘捕網站之 Domain name (目前為 http://140.115.53.34)	article.php、board.php、lib.php、

		log.php、login.php
not_login_yet : 0 login_error : 1 login_success : 2 mal_user : 3	使用者目前狀態 not_login_yet - 尚未登入 login_error - 帳號、密碼輸入錯誤(非惡意) login_success - 登入成功 mal_user - 登入成功的惡意使用者	article.php、board.php、lib.php、log.php、login.php

3. board.php : 論壇主畫面

4.article.php : 處理文章的新增、修改、刪除

需求追溯性

為整個系統做個總結，列出本系統所有套件、套件中的功能模組、各功能模組之組件(程式)，製作出一張需求追溯表。(含資料庫的設計)

註記

中英文名辭對照與縮寫(必要)

Honeypot：蜜罐

Entrap unit：誘捕單元

Manage Unit:管理單元

Record Unit: 紀錄單元

SQL injection：資料庫注入

security level：安全等級

remote control：遠端控制

附錄七、封包標記與追蹤系統設計報告書

系統綜觀

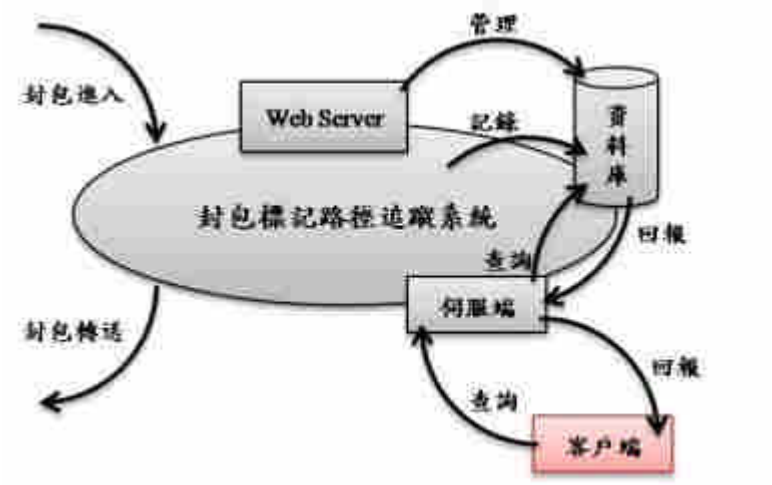


圖 1-1 系統運作

本軟體的運作如圖 1-1，封包的進入與轉送過程，封包資料的記錄，伺服器管理資料庫，客戶端與伺服端的路徑追蹤查詢與回報。

參考文件

[01] T. Baba and S. Matsuda, “Tracing network attacks to their sources,” IEEE Internet Computing, vol. 6, pp. 20-26, Apr. 2002.

[02] S. Savage, D. Wetheral, A. Karlin, and T. Anderson, “Practical network support for IP traceback,” in Proceedings of ACM SIGCOMM’00, vol. 30, pp. 295-306, Oct. 2000.

[03]A. Belenky and N. Ansari, “IP traceback with deterministic packet marking,” IEEE Communication Letters, vol. 7, pp. 162-164, Apr. 2003.

[04] A. Belenky and N. Ansari, “Tracing multiple attackers with deterministic packet marking (DPM),” in Proceedings of IEEE Pacific Rim Con. Communications, Computers and Signal Processing, vol. 1, pp. 49-52, Aug. 2003.

[05] A. Yaar, A. Perrig, and D. Song, “Pi: A path identification mechanism to defend against DDoS attacks,” in Proceedings of IEEE Symp. Security and Privacy, pp. 93–107, May 2003.

[06] A. Yaar, A Perrig, and D. Song, ”StackPi: New Packet Marking and Filtering Mechanisms for DDoS and IP Spoofing Defense,” IEEE JSAC, Volume 24, Issue 10,

pp. 1853 – 1863, Oct. 2006.
[07]R. Chen, J. Park, and R. Marchany, “RIM: Router Interface Marking for IP Traceback,” in Proceedings of IEEE GLOBECOM, pp. 1-5, Nov. 2006.
[08] Stefan Savage, David Wetherall, Anna Karlin and Tom Anderson, Practical Network Support for IP Traceback. In Proceedings of ACM SIGCOMM 2000 , pp. 295-306, August 2000.

電腦軟體構型項目層面設計決策

電腦軟體發展環境選用決策分析

軟體

(1) 封包標記軟體

- 使用何種語言:linux c
- 安裝在哪套作業系統: Linux –Ubuntu 8.04

(2) 偵測紀錄軟體

- 使用何種語言:linux c
- 安裝在哪套作業系統: Linux –Ubuntu 8.04

(3) 管理伺服器

- 使用何種語言: MySQL Database
- 為何選用此語言:因為是免費，網路承載比較少，應用程式透過它做起備份來也比較簡單，也為各種不同的資料格式提供彈性的介面，較好學，且操作簡單。
- 安裝在哪套作業系統: Linux –Ubuntu 8.04

(4) 路經追蹤軟體

- 使用何種語言: JAVA
- 為何選用此語言:因為在不同語言的平台上都能撰寫和執行。對於具備網路功能程式的撰寫，也是非常容易的。不論是一般網際網路的程式 Socket、

Email 等、伺服器網頁的程式 Servlet 等的相關套件支援可以說是非常的豐富，且使用起來也是非常的容易。

- 安裝在哪套作業系統: Microsoft Windows XP、Linux –Ubuntu 8.04

硬體

(1) MPC

- Intel E7200
- MD Gigabyte EP35-DS3L P35/ICH9
- 4GB DDRII 800 RAM
- VGA GeForce 7200 series (128M)
- 640GB HD
- PSU 350W
- DVD-ROM
- Network Interface Card(NIC) x 2

(2) MCC

- Intel E7200
- MD Gigabyte EP35-DS3L P35/ICH9
- 4GB DDRII 800 RAM
- VGA GeForce 7200 series (128M)
- 640GB HD
- PSU 350W
- DVD-ROM
- Network Interface Card(NIC) x 1

電腦軟體構型項目行為設計決策分析

本系統架構如圖 3-1，封包標記路徑追蹤系統可分為封包標記軟體、偵測記

錄軟體、管理伺服器與路徑追蹤軟體。其中封包標記軟體包括標記設定模組、標記模組；偵測記錄軟體包括記錄管理模組、偵測記錄模組；管理伺服器包括安全資料庫、使用者介面；路徑追蹤軟體包括追蹤軟體伺服端、追蹤軟體客戶端、路徑軟體伺服端、路徑軟體客戶端。

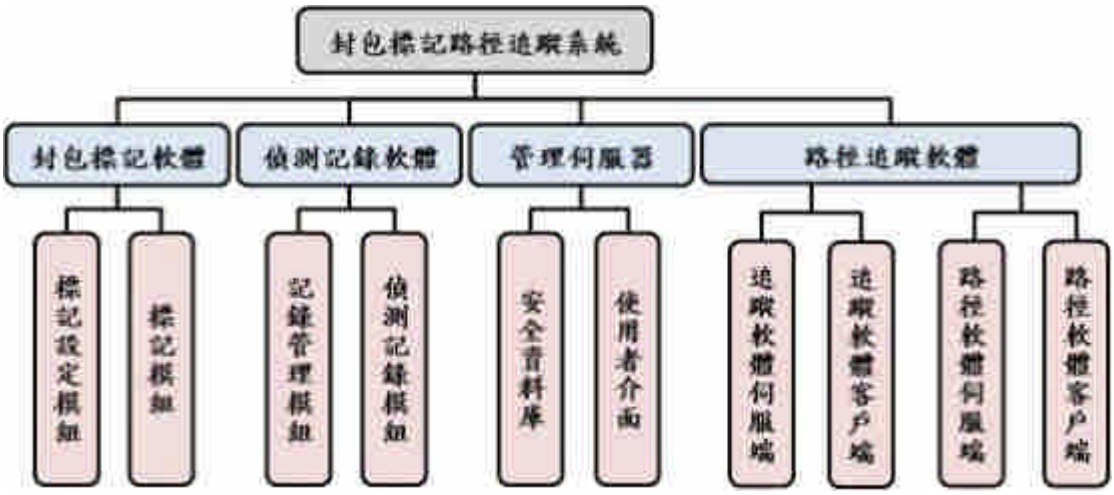


圖 3-1 系統架構

電腦軟體構性項目架構設計

電腦軟體構型項目組件

圖 4-1 展示出軟體間連結架構，在一個系統內所有軟體的關聯性。系統由 Linux 所架設出來，在 Kernel 新增變數與指令，在 Modules 修改 Bridge 模組，在網卡放置監聽程式，在使用者端架設 Server、管理資料庫、控制模組變數。

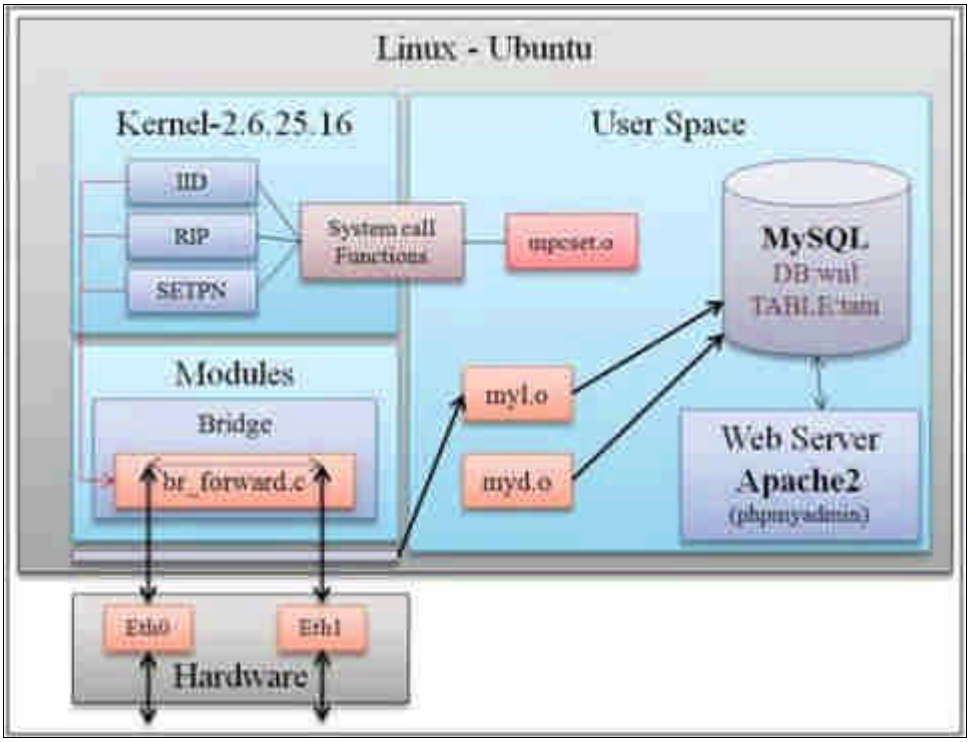


圖 4-1 軟體整體架構圖

封包標記軟體

本系統利用封包在網路上進行傳遞時，將經過本機的封包，進行封包內容的修改，再進行封包的轉送，使封包擁有標記資訊，項目包含：標記設定模組、標記模組。

標記設定模組	輸入	檢測參數檔
	處理	依據指定參數檔呼叫核心指令執行
	輸出	回報設定結果
	程式	mpcset.o
標記模組	輸入	網路封包
	處理	將封包進行標記
	輸出	標記的網路封包
	程式	br_forward.c

表 0-1 封包標記軟體項目需求概述

偵測紀錄軟體

本系統偵測從網路進入本機的封包內容，將封包內容截取出來，記錄在本機的資料庫上，項目包含：記錄管理模組、偵測記錄模組。

記錄管理模組	輸入	檢測參數檔
	處理	依據指定參數檔呼叫資料庫執行

	輸出	回報處理結果
	程式	myd.o
偵測記錄模組	輸入	網路封包
	處理	將封包資訊取出並記錄到資料庫
	輸出	回報記錄結果
	程式	myl.o

表 0-2 偵測記錄軟體項目需求概述

管理伺服器

本系統使用 Web Server 進行資料庫管理，提供簡易的使用者介面讓管理者管理，項目包含：安全資料庫、使用者介面。

安全資料庫	處理	架設資料庫並設定資料庫內容,讓偵測軟體能將記錄加入資料表中。
使用者介面	處理	簡易的 Web 頁面，讓使用者能夠簡易的去管理資料庫內容。

表 0-3 管理伺服器項目需求概述

欄位名稱	型態	欄位名稱	型態
(1)STime	DATETIME	(6)DPORT	SMALLINT UNSIGNED
(2)ETime	DATETIME	(7)IIDNUM	SMALLINT UNSIGNED
(3)SIP	INT UNSIGNED	(8)IID1	SMALLINT UNSIGNED
(4)DIP	INT UNSIGNED	(9)IID2	SMALLINT UNSIGNED
(5)Protocol	SMALLINTUNSIGNED		

表 0-4 MySQL 的 Table: tam 設計

STime 與 ETime 兩個欄位的格式型態是 DATETIME，用來記錄日期以及時間，STime 全名為 Start Time，記錄這個標記內容的起始時間，而 ETime 全名為 End Time,記錄這個標記內容的結束時間,SIP 為來源 IP,DIP 為目的 IP,Protocol 為 IP 通訊協定,DPORT 為目的 Port,IIDNUM 為封包內標記的 IID 數量,IID1、IID2 代表的是經過的 MPC 的各個 IID。

路徑追蹤軟體

本系統使用伺服端與客戶端軟體,分為兩種,一種是 MCC(客戶端)與 MPC(伺服端)間的溝通查詢，另一種是 MCC(客戶端、伺服端)間的查詢回報。讓 MPC 能夠輸入查詢條件進行路徑追蹤，取得追蹤路徑的結果，項目包含：追蹤軟體客戶端(MCC)、追蹤軟體伺服端(MPC)、路徑軟體客戶端(MPC)、路徑軟體伺服端

(MPC)。

追蹤軟體客戶端 (MCC)	輸入	資料庫查詢條件
	處理	將查詢條件送達伺服端，等待結果回傳
	輸出	查詢結果
	程式	Traceback.java
追蹤軟體伺服端 (MPC)	輸入	客戶端資料庫查詢條件
	處理	進入資料庫進行查詢,再將查詢條件送達伺服端，詢問前一個 MPC，等待結果回傳
	輸出	回報資料庫查詢結果
	程式	MPCSide.java

表 0-5 路徑追蹤軟體項目需求概述-1

路徑軟體客戶端 (MPC)	輸入	後一個 MPC 資料庫查詢條件
	處理	進入資料庫進行查詢,再將查詢條件送達伺服端，詢問前一個 MPC，等待結果回傳
	輸出	回報資料庫查詢結果
	程式	MPCClient.java
路徑軟體伺服端 (MPC)	輸入	客戶端資料庫查詢條件
	處理	進入資料庫進行查詢，取得查詢結果，若不是第一個 MPC，則再將查詢條件送達且詢問前一個 MPC，等待結果回傳
	輸出	回報資料庫查詢結果
	程式	MPCServer.java

表 0-6 路徑追蹤軟體項目需求概述-2

執行的概念

(1) mpcset.o 是在使用者環境下所使用的程式，mpcset.o 程式可進行變更核心的變數。

(2) Modules 部份，使用修改過後模組，使封包經由 Bridge 模組進行標記。

(3) 在使用者方面，myl.o 是常駐執行程式，進行封包標記的記錄，myd.o 同樣是常駐程式，進行資料庫定時清理功能。

(4) MPC 與 Server 的連線關聯圖如圖 4-2 所示。Client 透過 Traceback.java 的圖型介面，進行路徑追蹤，由 Socket 將查詢的內容傳送至 MPC，之後由 MPCSide.java 進行資料庫查詢，再將查詢結果回傳給 Client。

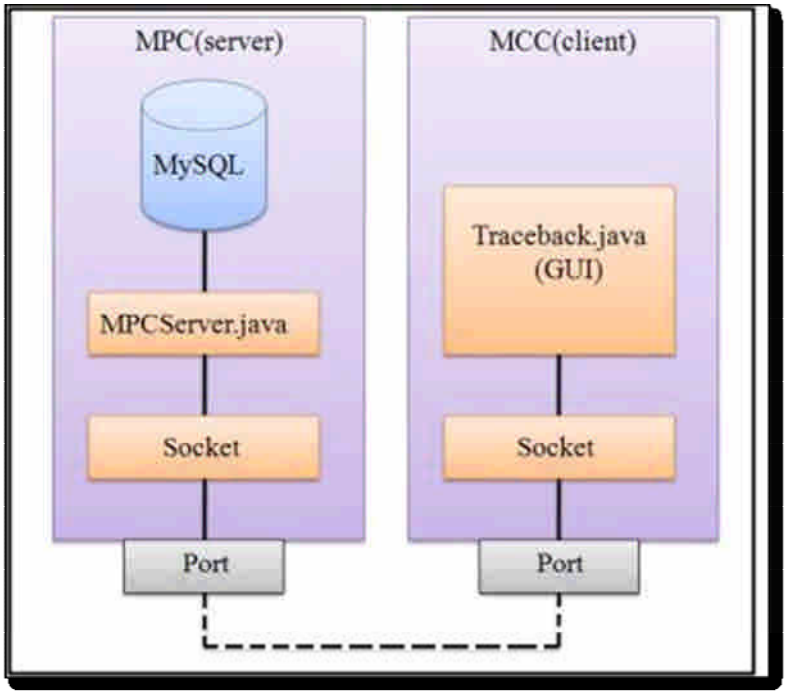


圖 4-2 MPC 與 MCC 關聯圖

(5) MPC 間的連線關聯圖如圖 4-3 所示。Client 透過 MPCClient.java 進行路徑查詢，由 Socket 將查詢的內容傳送至 Server，之後由 MPCServer.java 進行資料庫查詢，再將查詢結果回傳給 Client。

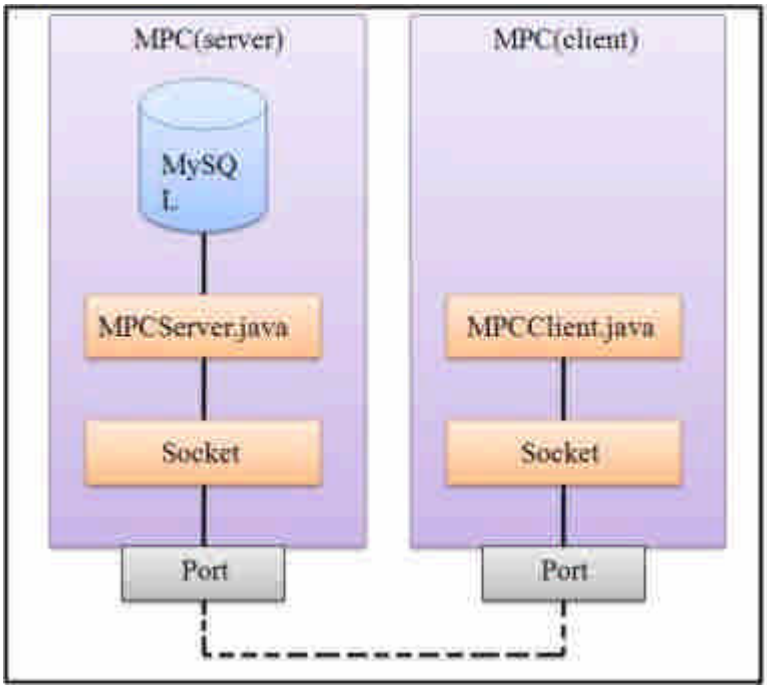


圖 4-3 MPC 間關聯圖

介面設計

系統核心設定

之前在核心部份，新增了變數以及讀寫變數的函式，使用者使用這些讀寫的函式，我們稱為使用”system call”，system call 所指的是由 user space 去讀取 kernel space 系統變數的意思，而使用者要去讀取變數，首先要在程式碼內加入 system call function 的所在位置及 library，然後由函式取得變數進行控制。

要控制的變數有一個 IID。IID 是用來控制所要標記的數值。為了使用上的方便，我們編寫了一個使用者介面的程式，原始檔為”mpcset.c”，編譯過後為”mpcset.o”，我們使用 mpcset.o 進行修改及設定 IID。在修改的過程中，我們也將修改的結果寫入設定檔”mpc.config”，當我們改變 IID 內容時，設定檔也會跟著將內容改變。

這個執行檔所能執行的指令如圖 4-4，參數使用 help 可以查詢使用方式(取消有關 rip 和 setpn 的設定)。

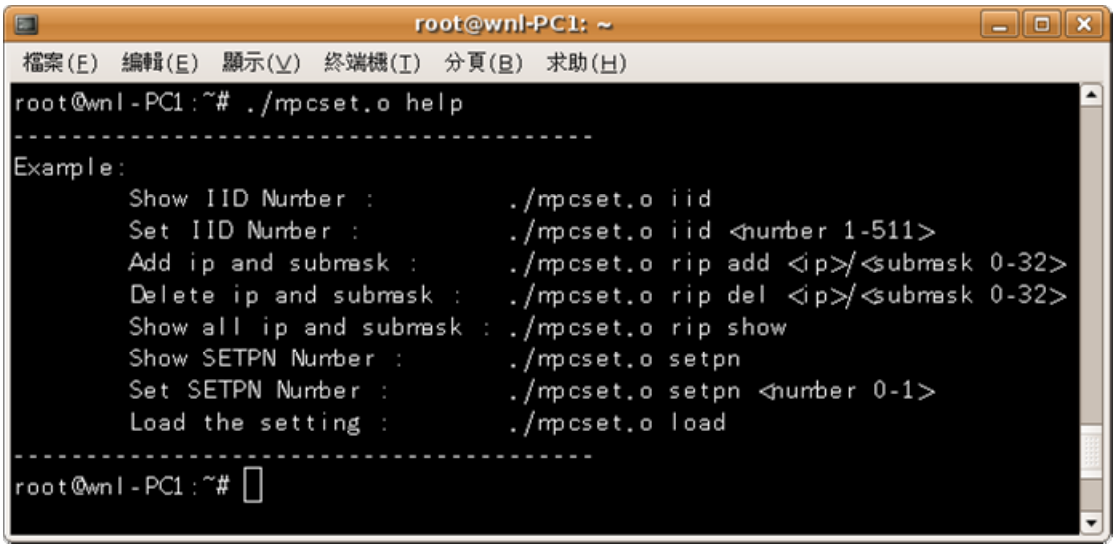


圖 4-4 執行 mpcset.o help 查詢可執行指令

使用 mpcset.o 這支程式去更改核心的標記設定。首先使用終端機進入到 mpcset.o 的目錄下(預設為~/，輸入”cd ~”)，如果需要將 IID 改變為 1，輸入”sudo ./mpcset iid 1”，觀看目前 iid 值，輸入”sudo ./mpcset iid” ，如圖 4-5。

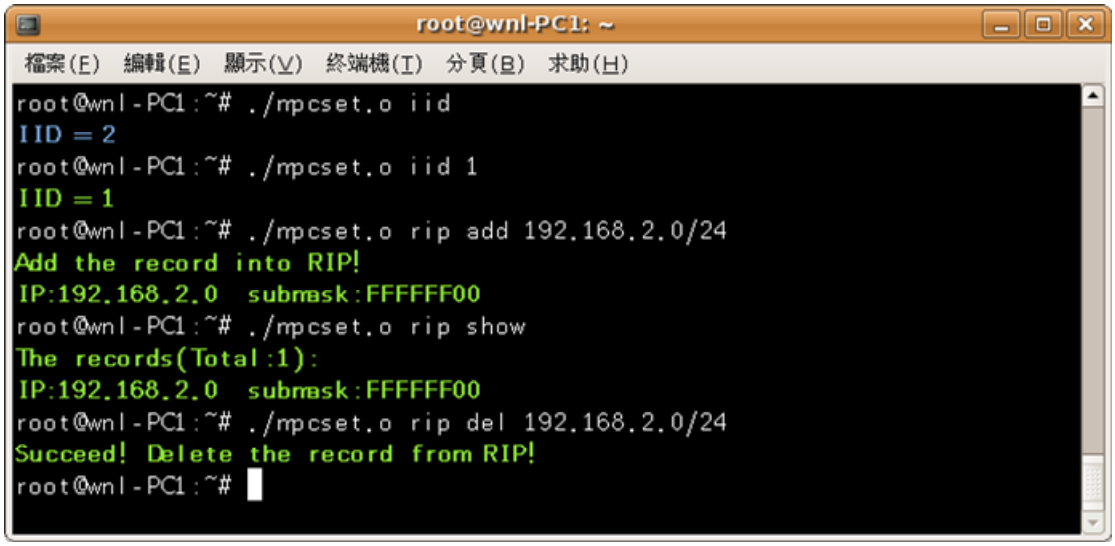


圖 4-5 執行參數 IID

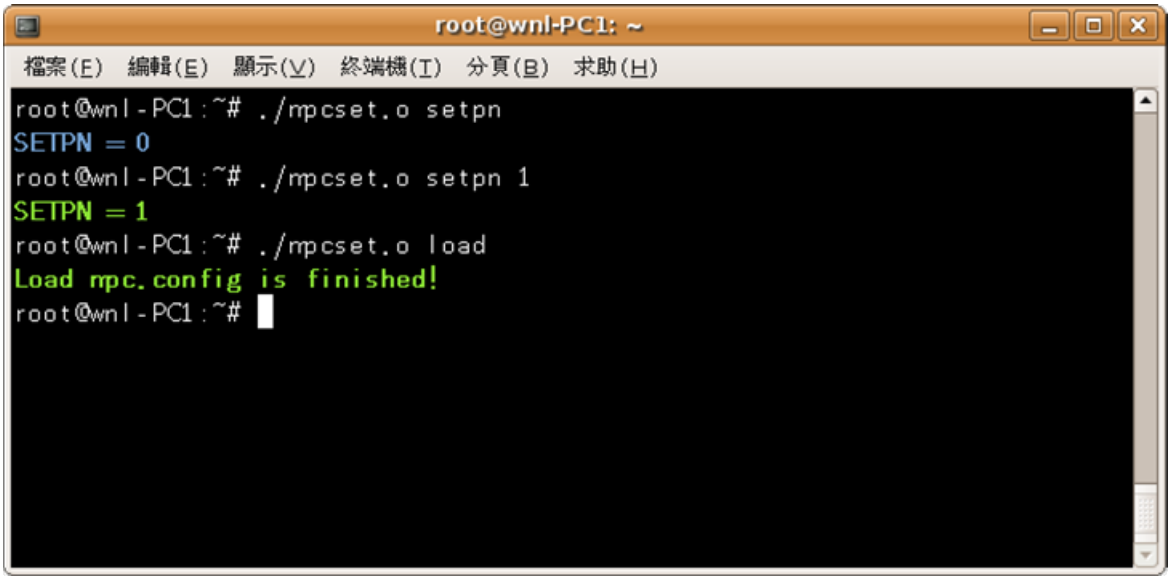


圖 4-6 執行參數 load

路徑追蹤查詢

在已有安裝 Java 的主機上，放置整個 Traceback 資料夾，然後執行 Traceback.java，輸入"java Traceback"， 就能夠執行 GUI 介面，如圖 4-7(只需 IID[1]和 IID[2])。

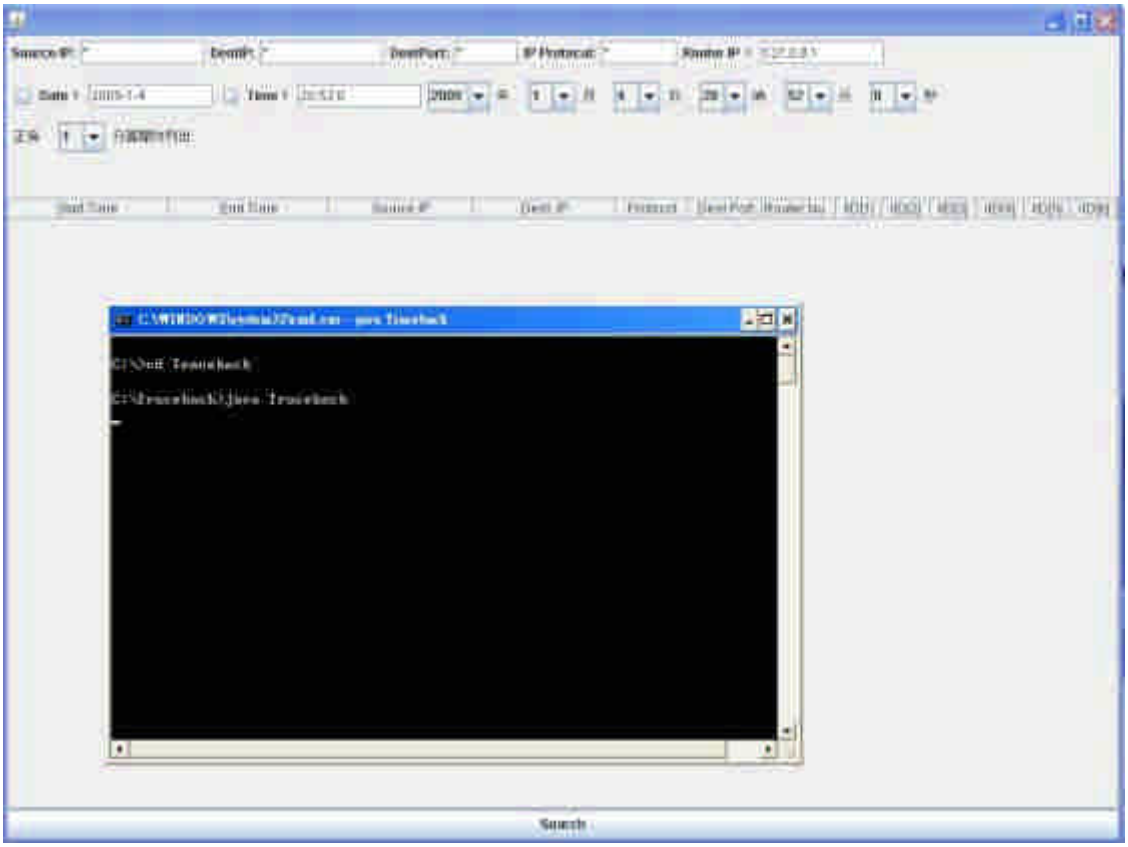


圖 4-7 Traceback 畫面

在畫面中，每一個欄位都是查詢的條件，使用”*”代表查詢時，這個欄位為不限制，Source IP 與 DestIP 欄位，放置 IP 位址(如 192.168.2.4)即可當搜尋條件。DestPort 則是放數字，其範圍在 0~65535。IP protocol 放置 6 代表 TCP 協定，17 代表 UDP 協定。Router IP 則是 MPC 的 IP，看架設時的 IP 為主。Date 跟 Time 的使用方式只有三種，Date & Time 都勾選、Date 勾選 & Time 不勾選和 Date &Time 都不勾選，只有 Time 勾選而 Date 沒有勾選是無法使用的，Date 與 Time 的選擇必須由後面的下拉選單拉選。選擇條件完成後，再按下最下方的 Search 按鈕，則可進行查詢的動作，查詢結果如圖 4-8，只是今年要呈現的是經過的 subnet 的順序。

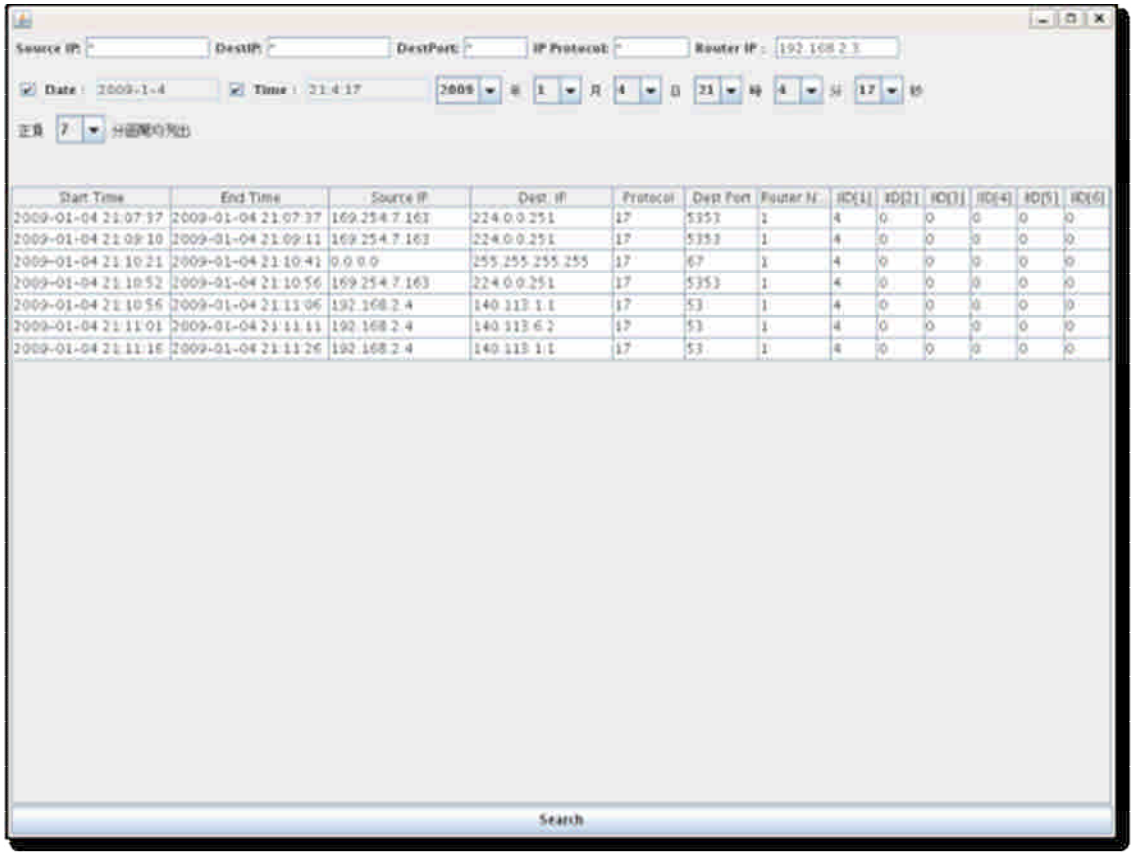


圖 4-8 Traceback 執行結果的畫面

資料庫設定

MySQL 資料庫設定主要是在於權限以及遠端存取的控制，我們使用 phpMyadmin 去進行網頁式設定。在先前的安裝設定，已經將 apache 以及 phpMyadmin 安裝完畢，所以我們可以從網頁輸入 http://127.0.0.1/phpmyadmin/ 進入的 MySQL 資料庫管理頁面，如圖 4-9 所示。

圖 4-9 phpMydmin 的登錄畫面

進入登錄畫面後，我們首先要新增我們要使用的資料庫 wnl，之後在資料庫 wnl 內新增 table 名稱為 tam，然後將欄位資料設定好，最後再進行權限的設定。

設定的順序如下：

(1) 首先建立資料庫，我們在此新增 wnl 資料庫，圖 4-10。

圖 4-10 建立新資料庫

(2) 進入 wnl 資料庫後，我們新增 table，名稱爲 tam，圖 4-11。

圖 4-11 新增資料表

(3) 設定 table 的欄位，圖 4-12。

圖 4-12 欄位設定

(4) 接下來是進入權限設定，之後新增使用者權限設定，圖 4-13。

圖 4-13 進入權限設定與新增使用者

(5) 權限新增完畢後，因爲新增了使用者，但並未讀取入權限區，所以再執行重新讀取權限讓它更新，圖 4-14。

圖 4-14 重新讀取權限

在資料庫、資料表和使用者都建立完畢後，要讓遠端電腦存取資料庫，還必須要修改一個設定檔，/etc/mysql/my.cnf 的這個設定檔內，我們要修改裡面一行為”bind-address = 127.0.0.1”，由於這行限制只有本機能夠存取資料庫，因此我們必須將它進行註解，使遠端可以存取，變為” #bind-address = 127.0.0.1”，整個資料庫的設定就全部完成了，如圖 4-15。

圖 4-15 註解原本的程式碼

電腦軟體構型項目細部設計

封包標記軟體

封包標記軟體共有 2 項之軟體單元，茲分節說明其細部設計事宜。

標記設定模組

標記設定模組之目的為標記過程中，設定封包標記的內容。

功能簡介

主要讓使用者能夠透過此模組進行標記內容的設定，設定的部份包含：標記編號設定。

處理流程

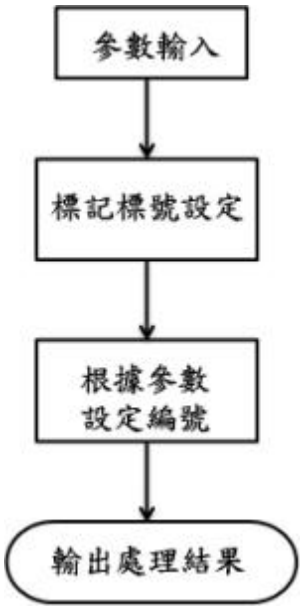


圖 5-1 標記設定模組

標記模組

封包轉送的過程中，將封包標頭進行修改，加入標記資料。

功能簡介

在封包進入本機後，將封包內的 IP Header 加入標記資訊—本機 IID 值，重新計算 Check Sum 並取代現有的內容。

處理流程

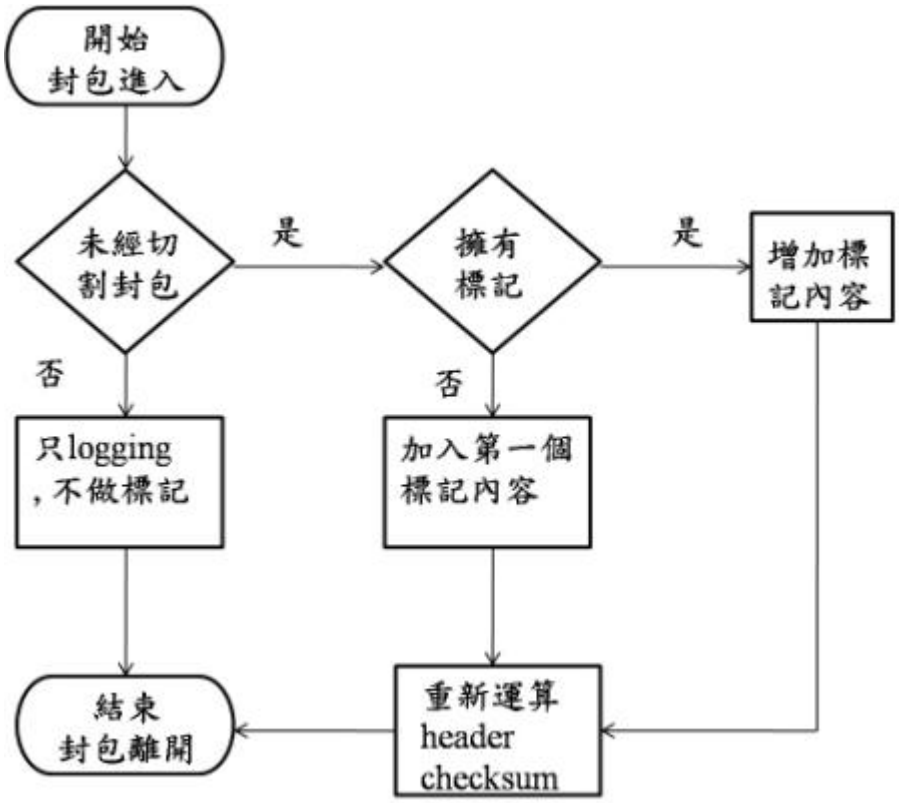


圖 5-2 標記模組

偵測記錄軟體

偵測記錄軟體共有 2 項之軟體單元，茲分節說明其細部設計事宜。

記錄管理模組

針對資料庫定期性的處理，將資料庫內容進行整理動作。

功能簡介

在資料庫中，資料長期記錄會造成資料搜尋動作減慢，需要定期進行刪減整理動作，記錄管理模組會將過舊的資料進行去除的動作。

處理流程

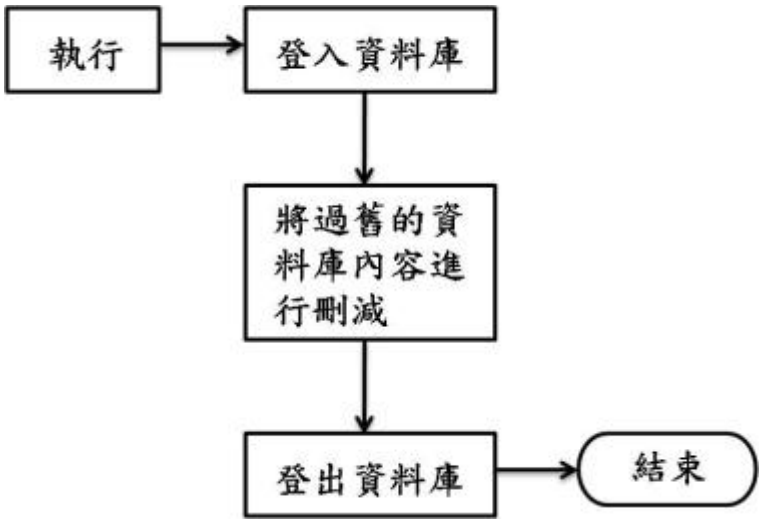


圖 5-3 記錄管理模組

偵測記錄模組

封包進入本機時，偵測記錄模組會取得封包內容，並進行記錄。

功能簡介

封包被偵測記錄模組取得後，會立即將封包進行拆解動作，取得所需要的標記資訊，並將標記資料暫存起來，一段時間後，登入資料庫，並將資料記錄到資料庫。

處理流程

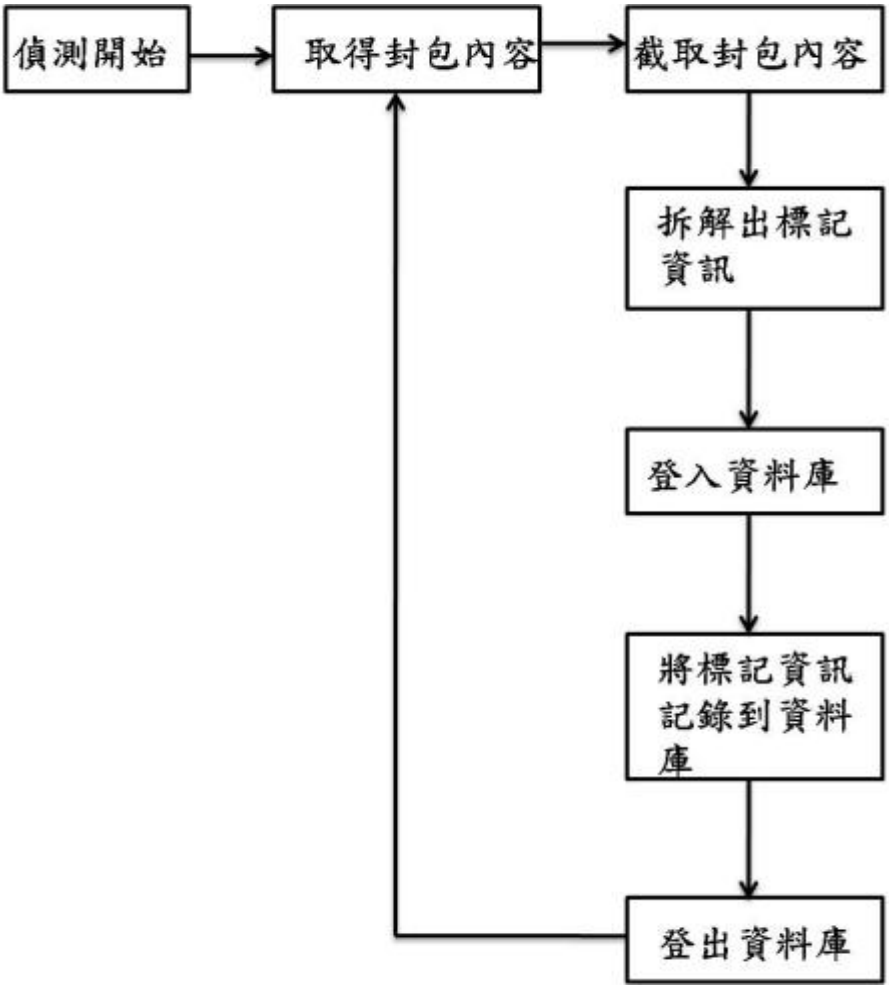


圖 5-4 偵測記錄模組

管理伺服器

管理伺服器共有 2 項之軟體單元，茲分節說明其細部設計事宜。

安全資料庫

安裝安全的資料庫，進行設定帳號密碼，讓使用者能夠使用資料庫。

安裝內容

資料庫使用 Mysql，架設 MySQL 資料庫。

使用者介面

在使用資料庫需要有簡易親和的介面，此介面為了讓管理者能夠更迅速的管理資料而安裝設定出來。

安裝內容

在伺服器方面，使用 Apache 當作 Server，然後再 Apache 設定能夠使用 php 頁面，最後在頁面上安裝 phpMyadmin 進行 Mysql 資料庫的控管。

路徑追蹤軟體

路徑追蹤軟體共有 2 項之軟體單元，茲分節說明其細部設計事宜。

追蹤軟體伺服器

主要為客戶端取得資料庫內容，客戶端無法直接取得資料庫內容，透過伺服端來取得資料庫內容，以確保資料庫安全。

處理流程

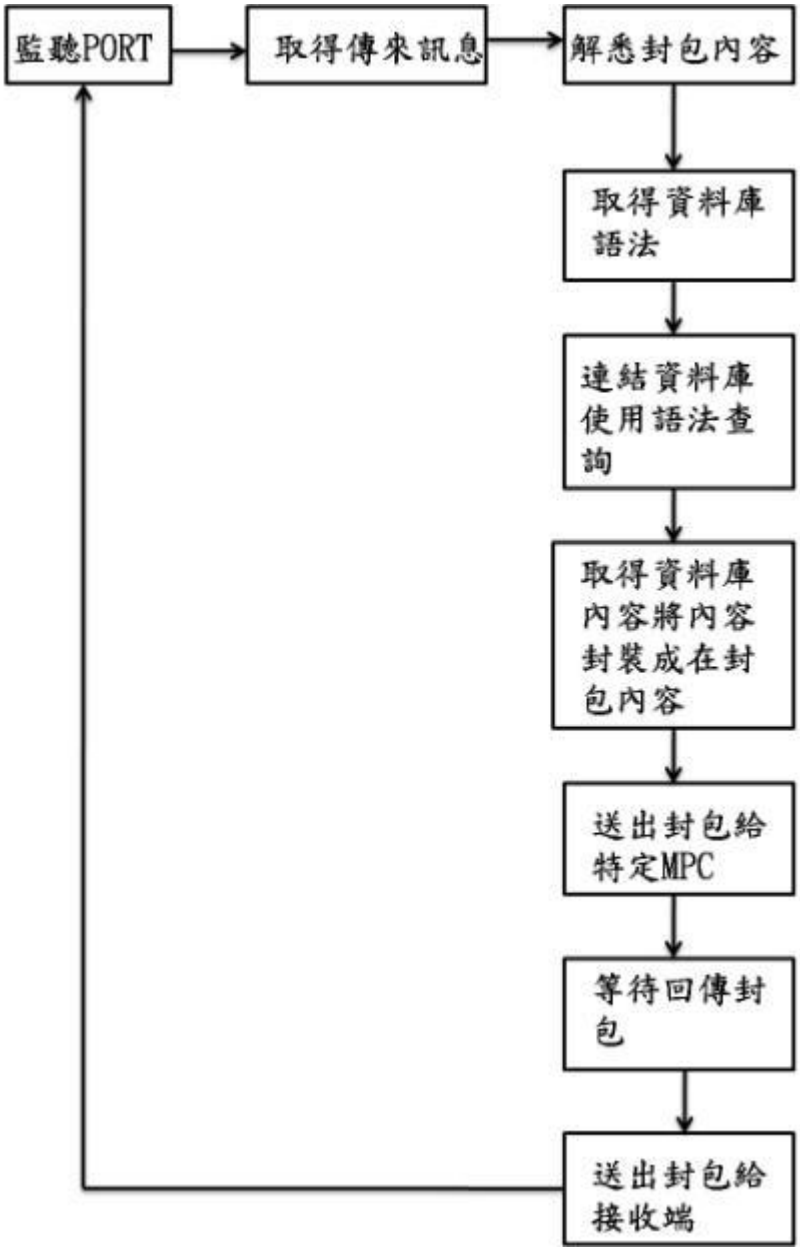


圖 5-5 追蹤軟體伺服端

追蹤軟體客戶端

客戶端設定查詢的條件，送出查詢，會將條件封裝在封包內容，之後將封包傳送至伺服端，接續等待封包的回覆；接收到伺服端所回覆的內容後，將封包內容進行解析，並拆解回原本的結果，並呈現在客戶端畫面。

處理流程

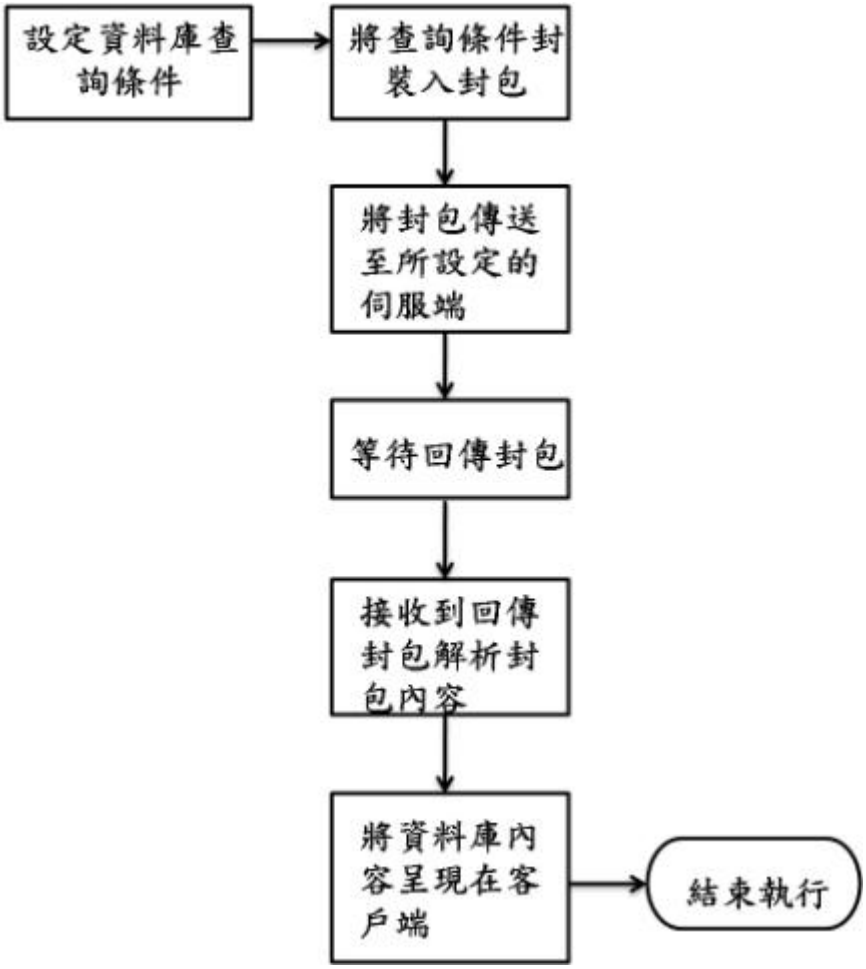


圖 5-6 追蹤軟體客戶端

路徑軟體伺服端

主要為客戶端取得資料庫內容，客戶端無法直接取得資料庫內容，透過伺服端來取得資料庫內容，以確保資料庫安全。

處理流程

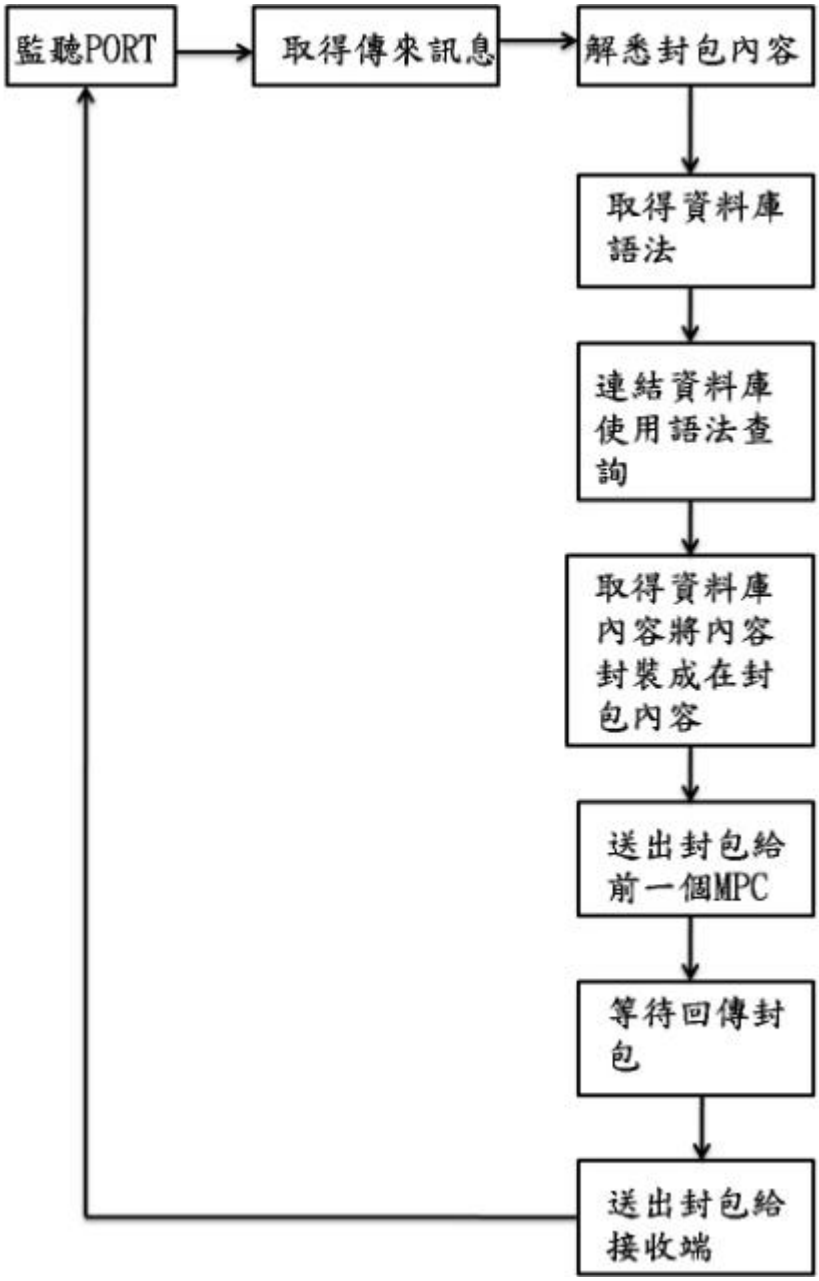


圖 5-7 路徑軟體伺服端

路徑軟體客戶端

客戶端設定查詢的條件，送出所選的條件，會將條件封裝在封包內容，之後將封包傳送至伺服端，接續等待封包回覆；接收到伺服端所回覆的內容後，回傳至客戶端。

處理流程

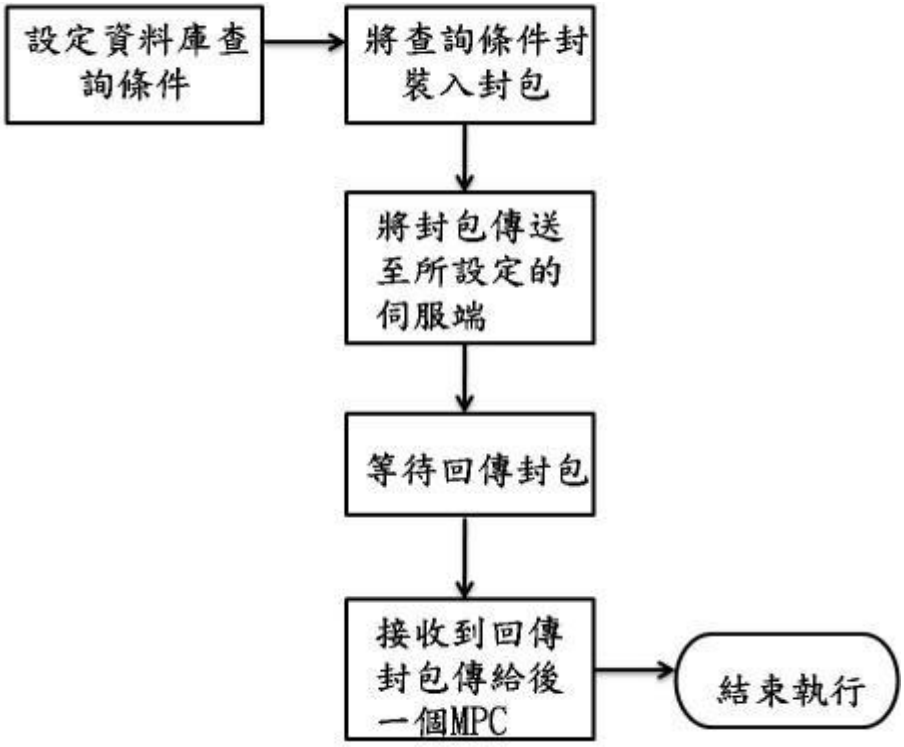

設定資料庫查詢條件
將查詢條件封裝入封包
將封包傳送至所設定的伺服端
等待回傳封包
接收到回傳封包傳給後一個MPC
結束執行

圖 5-8 路徑軟體客戶端

需求追溯性

整套系統的功能為標記封包，將標記封包內的資料存在資料內，藉由存在資料庫裡的封包標記內容，進行路徑追蹤，重建出整條路徑。

套件	功能模組	組件
封包標記軟體	標記設定模組	mpcset.c
	標記模組	br_forward.c
偵測記錄軟體	記錄管理模組	myd.o
	偵測記錄模組	myl.o
管理伺服器	安全資料庫	MySQL、table:tam
	使用者介面	Apach、 phpMyadmin
路徑追蹤軟體	追蹤軟體伺服端	Traceback.java
	追蹤軟體客戶端	MPCSide.java
	路徑軟體伺服端	MPCClient.java
	路徑軟體客戶端	MPCServer.java

表 7 需求追溯表

附錄八、網路風險分析及預警系統設計報告書

系統綜觀

本系統可以分為四大部份：履歷(profile)、法則編輯、資料倉儲及風險分析預警，如圖 1 所示。履歷部份用來為各主機建立其歷史連線紀錄並且比對已知的資安法則找出可能為異常的主機，履歷部份又可分為：profile builder、anomaly detector；資料倉儲將收集到的連線資料依各種維度進行整合，以利將來存取時能即時的拿到資料;法則編輯主要提供使用者介面來讓使用者輸入資安法則，並且將資安規則拆解成子規則(sub rule)存入資料庫，可分為：rule editor、rule browser 及 rule database；風險分析預警為主機計算其分險程度，進而做為預警之依據，其中又可分為：fault tree analysis 及 profile clustering

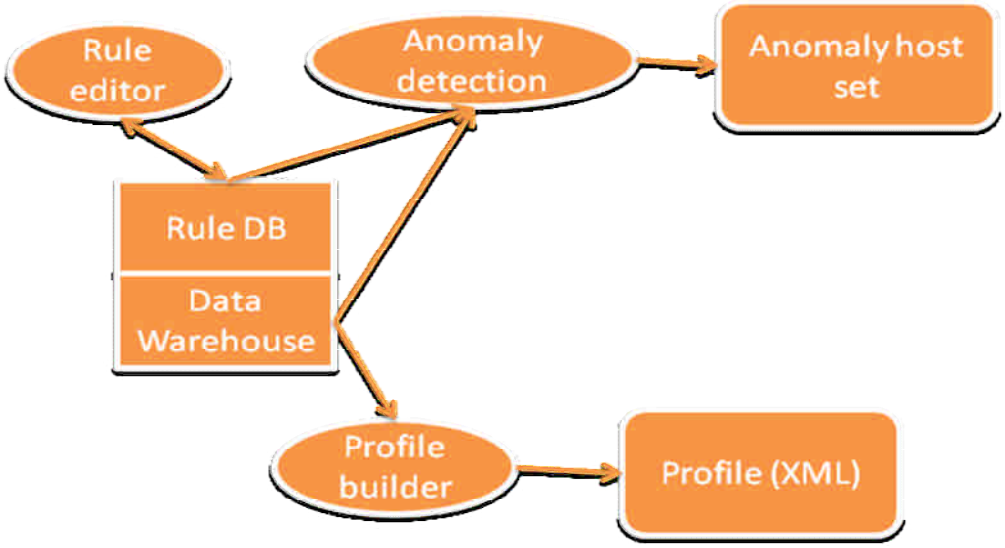


圖 1、系統架構

參考文件

[1] CVE List, http://cve.mitre.org/cve/index.html
[2] Blue Coat, http://www.bluecoat.com/
[3] Foundry, http://www.foundrynet.com/
[4] Citrix, http://www.citrix.com/lang/English/home.asp
[5] OWASP, http://www.owasp.org/
[6] C. Kruegel and G. Vigna, “Anomaly Detection of Web-based Attacks,”in ACM conference on Computer and Communications Security (CCS), 2003.
[7] X. Wang, J. Zhou, S. Yu and L. Cai, “Data Mining Methods for Anomaly Detection of HTTP Request Exploitations,”in Fussy System and Knowledge

discovery (FSKD), 2005.

[8] K.L. Ingham and H. Inoue, “Comparing Anomaly Detection Techniques for HTTP,”in International Symposium on Recent Advances in Intrusion Detection (RAID), 2007.

[9] K. Golnabi, R.K. Min, L. Khan and E. Al-Shaer,“Analysis of Firewall Policy Rules Using Data Mining Techniques,” in IEEE/IFIP Network Operations and Management Symposium, 2006.

[10] Aleksandar Lazarevic, Levent Ertoz, Vipin Kumar, Aysel Ozgur and Jaideep Srivastava, “A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection,” in SDM Conference, 2003.

[11] D.E. Denning, “An Intrusion Detection Model,” in IEEE Transactions on Software Engineering, SE-13:222-232, 1987.

[12] T. Lane, C. E. Brodley, “Sequence Matching and Learning in Anomaly Detection for Computer Security,”in AAAI Workshop: AI Approaches to Fraud Detection and Risk Management, 1997.

[13] K. Sequeira, M. Zaki, “ADMIT: Anomaly-base Data Mining for Intrusions,” in ACM SIGKDD Conference, Edmonton, 2002.

[14] V. Barnett, T. Lewis, “Outliers in Statistical Data,” John Wiley and Sons, NY 1994.

[15] C. C. Aggarwal, P. Yu, “Outlier Detection for High Dimensional Data,” in ACM SIGMOD Conference, 2001.

[16] M.F. Chiang, W.C. Peng and C.H. Lo, "Discovering Popular Co-Cited Communities in Blogspaces," in ICDE workshop on Data Engineering for Blogs, Social Media, and Web 2.0, 2008.

[17] Y. Chi, S. Zhu, X. Song, J. Tatemura, and B. L. Tseng, “Structural and temporal analysis of the blogosphere through community factorization,” in ACM SIGKDD, 2007.

[18] Alberts, Christopher J. and Dorofee, Audrey J, ”OCTAVESM Method Implementation Guide,” v2.0, Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001.

[19] C&A Systems Security Ltd, COBRA: Introduction to Risk Analysis, http://www.ca-systems.zetnet.co.uk/risk.htm

[20] International Information System Security Certification Consortium (ISC)2, Security management, http:://www.os-global.com

[21] L.C. Wuu, Chan S.H., and C.H. Hung, “Implement an IP Traceback on Linux Platform,” 2006 symposium on Open Source Technology and Application, TAIWA.

[22] H. Wang, D. Zhang, and K. G. Shin, “Detecting SYN flooding attacks,” In Proceedings of IEEE INFOCOM, 2002.

[23] S. Singh, C. Estan, G. Varghese, and S. Savage,”Automated Worm Fingerprinting,”In Proceedings of Symposium on Operating Systems Design and Implementation,(OSDI), 2004.

[24] Shigang Chen, and Sanjay Ranka, “Detecting Internet Worms at Early Stage,” IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL.23, No. 10, 2005.

[25] B. E. Brodsky and B. S. Darkhovsky., Nonparametric Methods in Change-point Problems, Kluwer Academic Publishers, 1993.

[26] J. Kang, Z. Zhang, and J. B. Ju, “Protect E-commerce against DDoS Attacks with Improved D-WARD Detection System”, IEEE International Conference on e-Technology, e-Commerce and e-Service, March 2005.
[27] Peter Mell, Karen Scarfone and Sasha Romanosky, “CVSS A Complete Guide to the Common Vulnerability Scoring System Version 2.0,” FIRST, http://www.first.org/cvss.
[28] Microsoft Corporation. Microsoft Security Response Center Security Bulletin Severity Rating System, http://www.microsoft.com/technet/security/bulletin/rating.mspx.
[29] United States Computer Emergency Readiness Team (US-CERT). US-CERT Vulnerability Note Field Descriptions. http://www.kb.cert.org/vuls/html/fieldhelp.
[30] SANS Institute. SANS Critical Vulnerability Analysis Archive. http://www.sans.org/newsletters/cva/.
[31] Yue-Lung Cheng, “Uncertainties in Fault Tree Analysis,” Tamkang Journal of Science and Engineering, Vol. 3, No. 1, 2000.
[32] Liang G. S. and Wang M. J., “Fuzzy fault tree analysis using failure possibility, Microelectron Reliability,” Vol. 33, No. 4, 1993.
[33] Misra K. B. and Weder G. G., “A new method for fuzzy fault tree analysis,” Vol. 29, No. 2, 1989.
[34] Tanaka H., Fan L. T., Lai F. S. and Toguchi K., “Fault tree analysis by fuzzy probability,” IEEE Trans. Reliability, Vol. 32, No. 5, 1983.
[35] 林盈達,林柏青,”網路安全產品測試評比-功能與效能面”

電腦軟體構型項目層面設計決策

電腦軟體發展環境選用決策分析

本系統基於可移植性(portable)使用 Java 開發程式，作業系統則為 Debian GNU/Linux，在資料庫方面使用自由軟體(open source)的 MySQL Database，firewall 則參考[35]技術報告中推薦的項目，使用 Netscreen-5GT。

電腦軟體構型項目行為設計決策分析

名稱	所屬單元	功能
Log parser	資料來源擷取系統	將 Log 存入資料庫
Profile builder	profile	利用 data warehouse 預算的資料建立 profile
Profile clustering	風險分析	依據各主機之 profile 進行分群

Anomaly detection	profile	比對 profile 和 rule，找出異常主機
Rule editor	法則編輯	提供使用者輸入自訂法則
Rule browser	法則編輯	提供使用者瀏覽已定義之法則
Create data warehouse table	資料倉儲	依據資料庫中的log建立未來將使用到的資料倉儲表格
Data warehouse	資料倉儲	將資料庫中的 Log 依據不同維度進行整合
Fault tree analysis	風險分析	計算各主機之風險程度

電腦軟體構型項目架構設計

電腦軟體構型項目組件

Log DB

SNo	序號
Datatime	日期時間
Source IP	來源位址
Source Port	來源埠號
Destination IP	目的位址
Destination Port	目的埠號

Host group

id	序號
IP	主機位址
gid	群組編號

Group

id	序號
name	群組名稱

Combination Port

id	序號
cid	組合編號

port	埠號

Available Port

id	序號
port	埠號
gid	群組編號

Port group

id	序號
pgid	群組編號
port	埠號

Rule

Rule	法則編號
Root	是否為 Root
Left	左子法則編號
Right	右子法則編號
Operation	運算方式
Description	法則描述

Sub-rule

Sub-Rule	子法則編號
Source ip	來源位址
Source Port	來源埠號
Destination IP	目的位址
Destination Port	目的埠號
datetime	日期時間
max	最大值
min	最小值
Unsafe Port	不安全的埠號組合
Source ip gid	來源群組

Source Port gid	來源埠群組
Destination IP gid	目的群組
Destination Port gid	目的埠群組

執行的概念

名稱	功能
Log parser	每天將 Log 存入 Log DB
Profile builder	將每個小時的 log 轉成 snapshot，並且將至多一個月的 snapshot 建立為 profile
Profile clustering	將 tree 轉為 feature vector，進行分群
Anomaly detection	每天比對 profile 和所有的 rules，將異常主機 list 傳給 master SIM
Rule editor	GUI 介面提供給使用者編輯、修改、新增資安規則
Rule browser	提供使用者瀏覽已定義之法則
Create data warehouse table	依據資料庫中的log建立未來將使用到的資料倉儲表格
Data warehouse	將資料庫中的 Log 依據不同維度進行整合
Fault tree analysis	計算各主機之風險程度

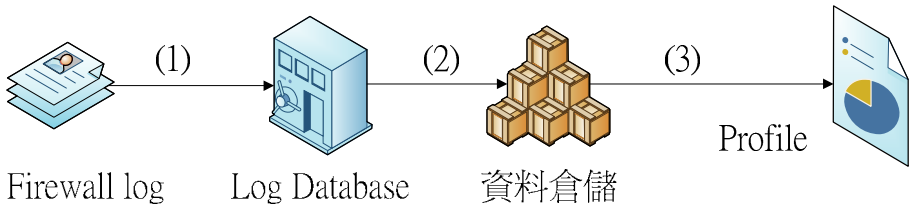
介面設計

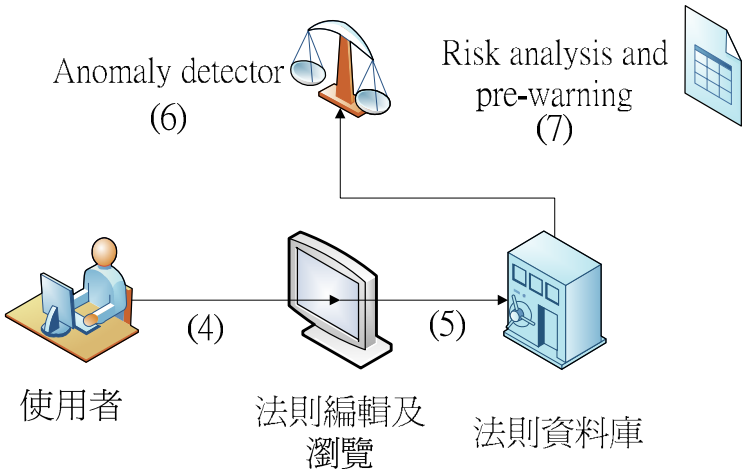
名稱	介面及參數
Log parser	無可用參數
Profile builder	無可用參數
Profile clustering	無可用參數
Anomaly detection	無可用參數
Rule editor	GUI 介面提供給使用者編輯、修改、新增資安規則
Rule browser	提供使用者瀏覽已定義之法則
Create data warehouse table	無可用參數
Data warehouse	無可用參數
Fault tree analysis	無可用參數

電腦軟體構型項目細部設計

物件導向分析(OOA)

Use Case





Log parser 依照欄位每天將 Firewall log 讀入 Log database

(1) 資料倉儲將一天的 Log 由 Log database 中讀出來，根據不同的維度建立不同時間長度的 data warehouse 表格，並且將資料加總後存入 data warehouse。
(2) Profile builder 利用資料倉儲的資料建立 profile。
(3) 使用者登入 MCC 利用法則編輯器新增、修改或刪除法則。
(4) 將法則拆解成子法則存入法則資料庫
(5) Anomaly detector 將 profile 及 rule 進行比對，將異常的主機存成 list 並且上傳給 Master SIM。
(6) Risk analysis 每天計算所有主機的風險值並且建立各主機之群組，計算其風險值。

法則編輯子系統使用案例

內容描述

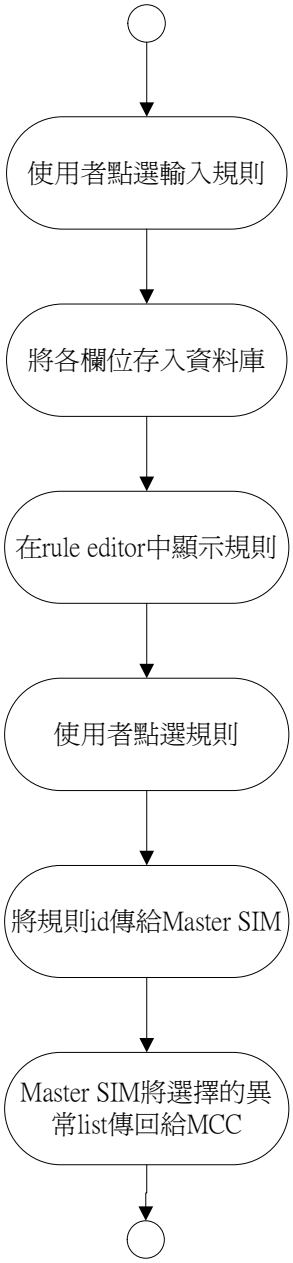
使用者經由滑鼠及鍵盤設定自定之規則，並透過 rule browser 選擇檢視違反該規則之主機連線狀態

功能與性能需求流程(Flow of Events)

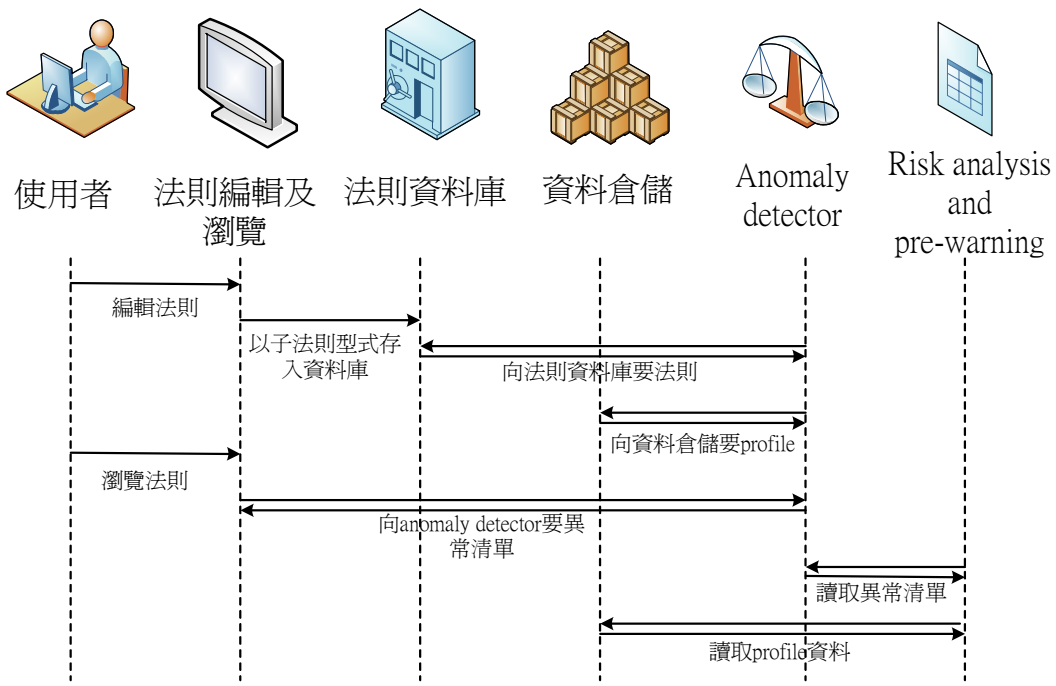
使用者由 rule browser 選定規則後，可得到 rule id，透過傳送 rule id 給 master SIM

可以得知使用者欲檢視之規則編號(rule id)。

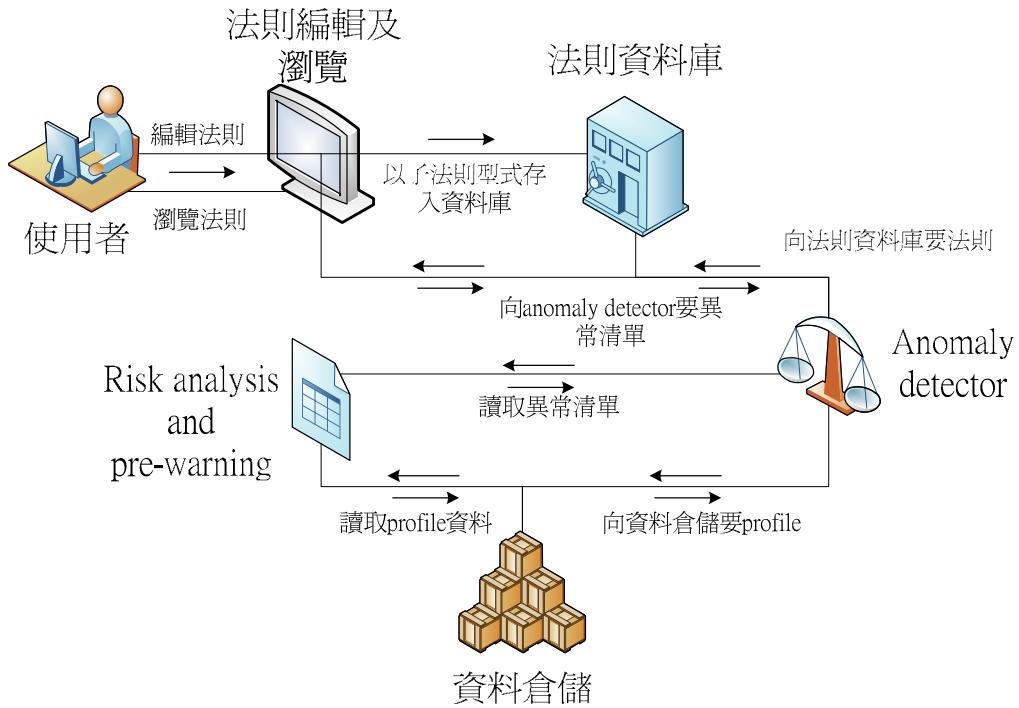
活動圖(Activity Diagram)



循序圖(Sequence Diagram)



合作圖(Collaboration Diagram)



註記

中英文名辭對照與縮寫(必要)

Profile 履歷

Anomaly detection 異常偵測

Clustering 分群

Fault tree 失誤樹

附錄九、系統界面整合文件

Traceback

There are two parts in the system. One is the MCC server, the other is the MPC server. The following is the related code file and file category bind to each server.

MCC server:

|--Architecture

|-- Traceback.java

|-- MccServer.java

|-- MccThreadTask.java

|-- MccTimerTask.java

|-- Connector.java

|--GUI

|-- NetFlowButton.java

|-- QueryPanel.java

|-- SimpleTable.java

|--MessageType

|-- MessagePacket.java

|-- Command.java

|-- MCCEntry.java

|-- TAMEntry.java

MPC server:

|--Architecture

|-- MPCSide.java

|-- ServerThread.java

|-- ConnectionThread.java

|-- Connector.java

|--MessageType

|-- MessagePacket.java

|-- Command.java

|-- MCCEntry.java

|-- TAMEntry.java

The following is descriptions of each code file.

Traceback.java:

The main GUI and create a MccServer to negotiate with MPC server and create a timer to schedule the request from the Honeynet server.

MccServer.java:

Create a thread pool and a server socket to accept connection from MPC server.

MccThreadTask.java:

Take each requests and responses from the MPC server or Honeynet server and process each request and response depends on corresponding subsystem and request message type.

MccTimerTask.java:

A thread used to execute the traceback operation according the request from the Honeynet server.

Connector.java:

Create a connection with the localhost database.

NetFlowButton.java:

A button associated with the request which come from the Netflow subsystem.

QueryPanel.java:

A panel which is used to accommodate the NetFlowButton.

SimpleTable.java:

A table which is used to show the path responses.

MessagePacket.java:

A message packet type to let the MCC server, MPC server and Honeynet server to do communication.

Command.java:

Each MccTimerTask has a command object which tells what the operation the

task needs to do.

MccEntry.java:

A message type which equal to the field of mcc table in the database.

TAMEntry.java:

A message type which equal to the field of tam table in the database.

MPCSide.java:

The main function which create a ServerThread object.

ServerThread.java:

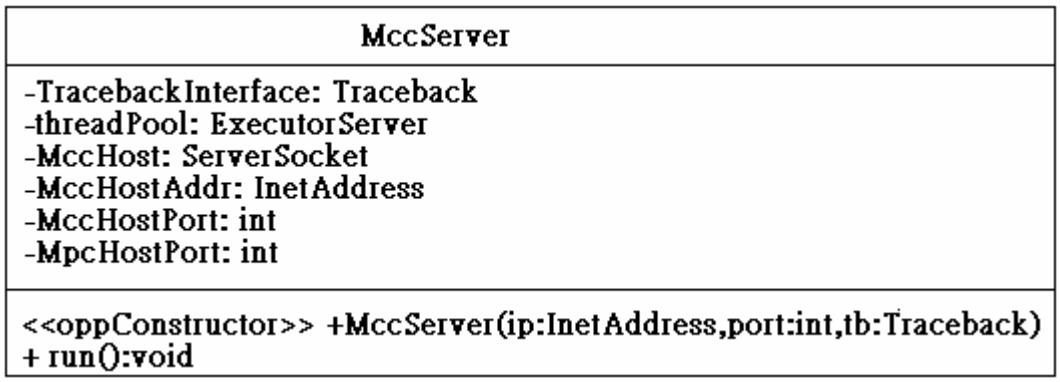
Create a MPC server socket and a thread pool to accept the connections from MCC server.

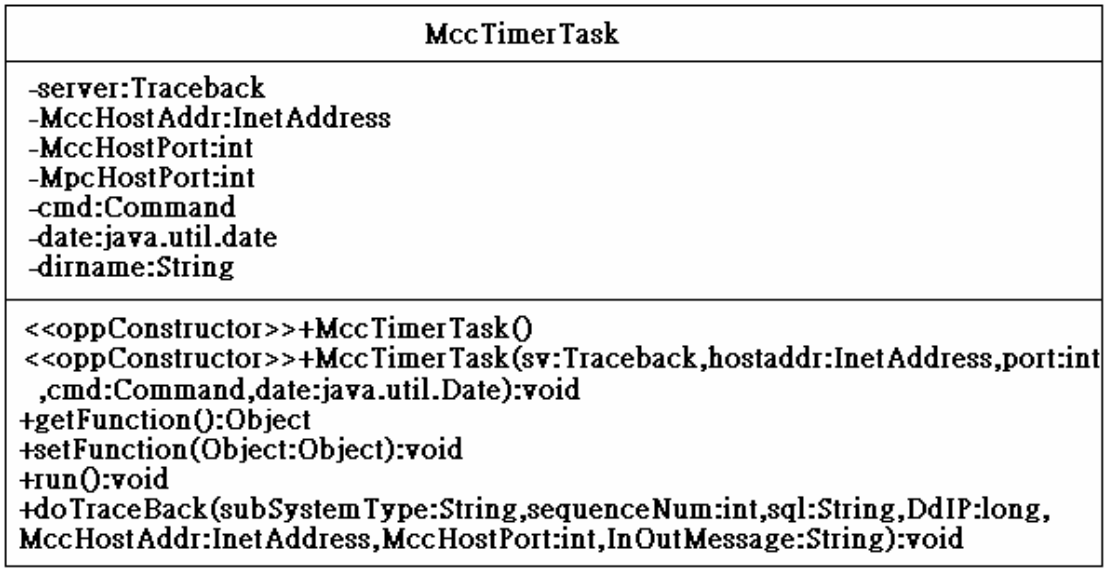
ConnectionThread.java:

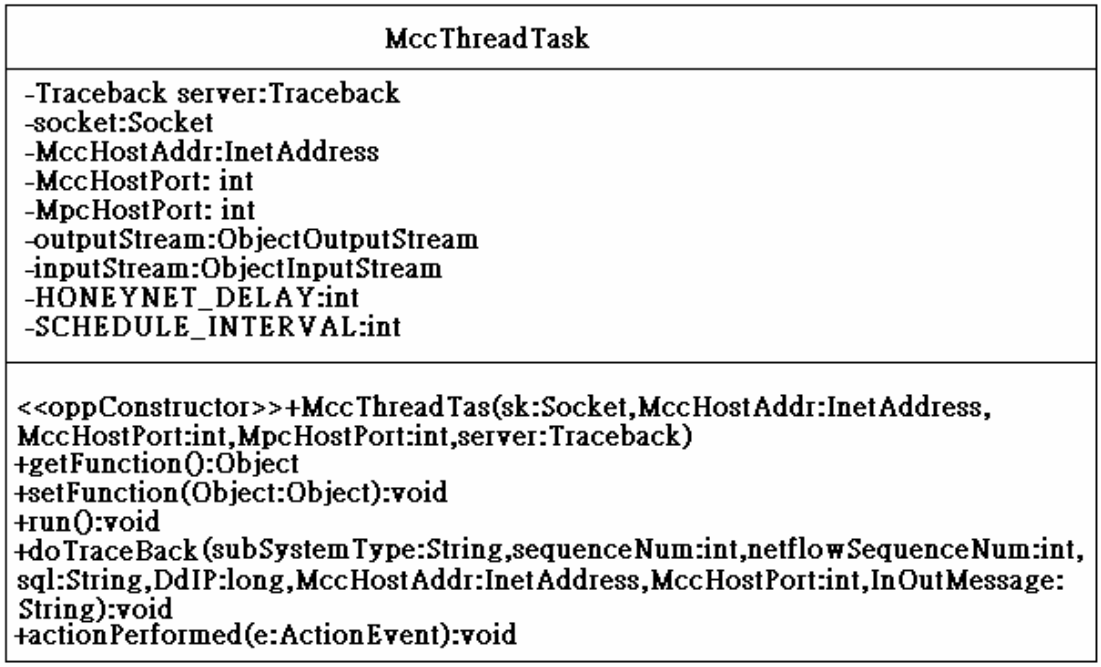
Take each request from the MCC server or the other MPC server and do some operations according to the request message type.

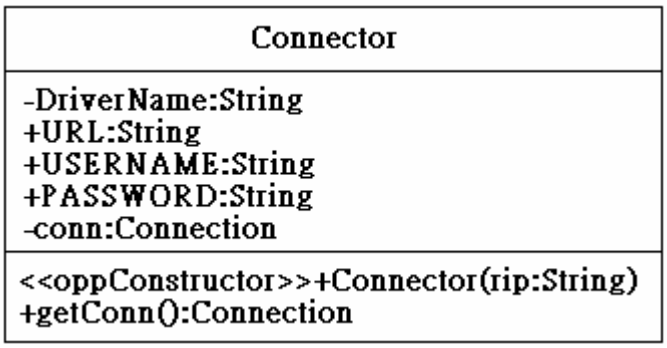
The following is each class diagram.


Traceback
-server: MccServer
-MccHostAddr:InetAddress
-MccHostPort: int
-MpcHostPort: int
-sequenceNum: int
-honeynetSequenceNum: int
-netflowSequenceNumMain: int
-netflowSequenceNum: int
-correctPathList: Vector<Object>
-errorPathList: Vector<Object>
-netflowCorrectPathList: Vector<Object>
-netflowErrorPathList: Vector<Object>
-NetflowButtonList: Vector<Object>
-honeynetCorrectPathList: Vector<Object>
-honeynetErrorPathList: Vector<Object>
-FindInPath_QueryList: Vector<Object>
-FindOutpath_QueryList: Vector<Object>
-TimerTaskList: Vector<Object>
-ServerTimer: java.util.Timer
-day:GregorianCalendar
<<oppConstructor>>+Traceback()
+getFunction():Object
+setFunction(Object:Object):void
+invokeWindowClose():void
+honeyNetSaveToFile():void
+saveToFile():void
+invokeTraceBack(sourceIP:String):int
+increaseSequenceNum():void
+increaseHoneyNetSequenceNum():void
+increaseNetflowSequenceNum():void
<<oppEvent>> +actionPerformed(e:ActionEvent):void
<<oppEvent>> +itemStatedChanged(ie:ItemEvent):void


MccServer
-TracebackInterface: Traceback
-threadPool: ExecutorServer
-MccHost: ServerSocket
-MccHostAddr: InetAddress
-MccHostPort: int
-MpcHostPort: int
<<oppConstructor>> +MccServer(ip:InetAddress,port:int,tb:Traceback)
+ run():void


MccTimerTask
-server:Traceback
-MccHostAddr:InetAddress
-MccHostPort:int
-MpcHostPort:int
-cmd:Command
-date:java.util.date
-dirname:String
<<oppConstructor>>+MccTimerTask()
<<oppConstructor>>+MccTimerTask(sv:Traceback,hostaddr:InetAddress,port:int
,cmd:Command,date:java.util.Date):void
+getFunction():Object
+setFunction(Object:Object):void
+run():void
+doTraceBack(subSystemType:String,sequenceNum:int,sql:String,DdIP:long,
MccHostAddr:InetAddress,MccHostPort:int,InOutMessage:String):void

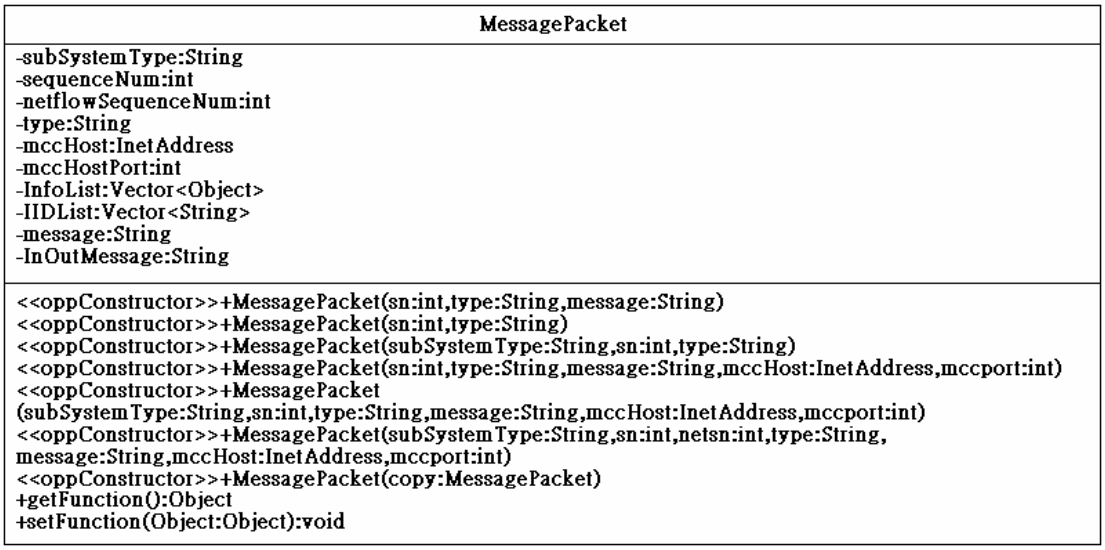

MccThreadTask
-Traceback server:Traceback
-socket:Socket
-MccHostAddr:InetAddress
-MccHostPort: int
-MpcHostPort: int
-outputStream:ObjectOutputStream
-inputStream:ObjectInputStream
-HONEYNET_DELAY:int
-SCHEDULE_INTERVAL:int
<<oppConstructor>>+MccThreadTas(sk:Socket,MccHostAddr:InetAddress,
MccHostPort:int,MpcHostPort:int,server:Traceback)
+getFunction():Object
+setFunction(Object:Object):void
+run():void
+doTraceBack(subSystemType:String,sequenceNum:int,netflowSequenceNum:int,
sql:String,DdIP:long,MccHostAddr:InetAddress,MccHostPort:int,InOutMessage:
String):void
+actionPerformed(e:ActionEvent):void

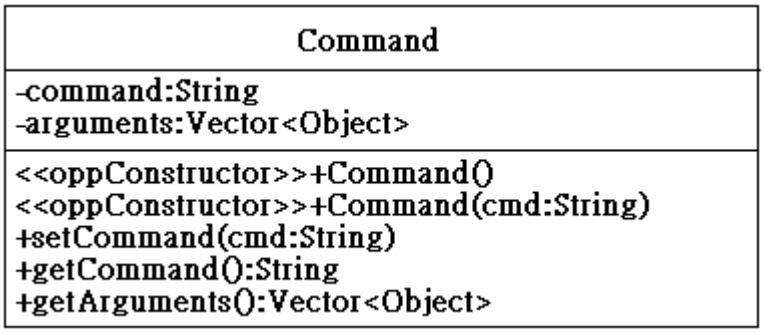

Connector
-DriverName:String
+URL:String
+USERNAME:String
+PASSWORD:String
-conn:Connection
<<oppConstructor>>+Connector(rip:String)
+getConn():Connection

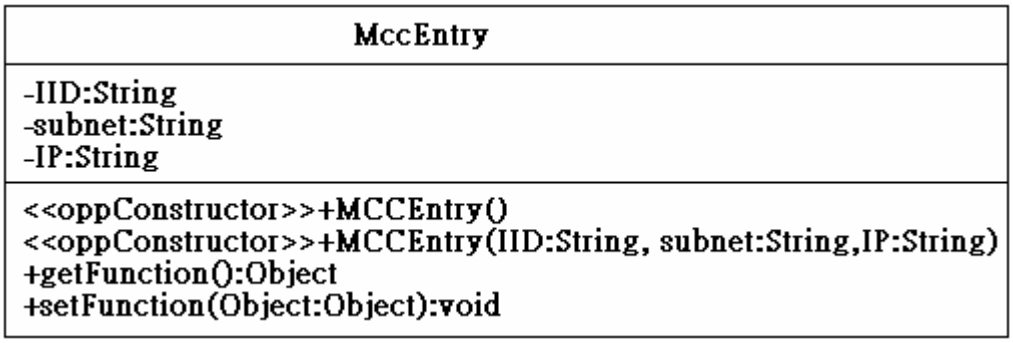
NetFlowButton
-SIP:String -DIP:String -DPORT:String -Protocol:String -subSystemType:String -sequenceNum:int -netflowSequenceNum:int -sql:String -Ddip:long -MccHostAddr:InetAddress -MccHostPort:int -InOutMessage:String
<<oppConstructor>> +NetFlowButton() <<oppConstructor>> +NetFlowButton(SIP:String,DIP:String, DPORT:String,Protocol:String,sn:int,sql:String,Ddip:long, MccHostAddr:InetAddress,MccHostPort:int,InOutMessage:String) +getFunction():Object +setFunction(Object:Object):void

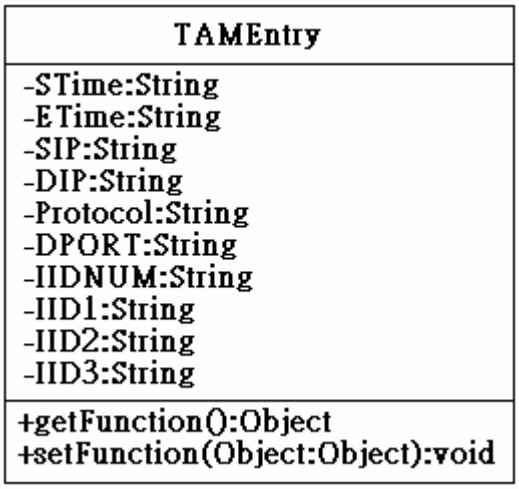
QueryPanel
-jsp:JScrollPane -jp:JPanel
<<oppConstructor>> +QueryPanel() +addComponent(obj:Component):void +removeAllRows():void

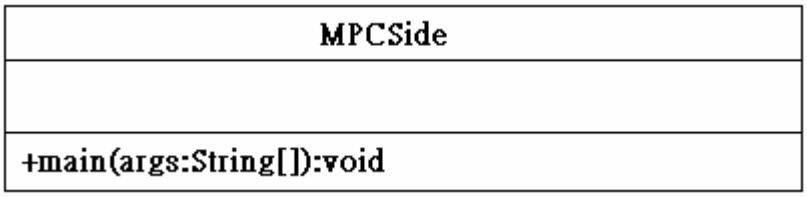
SimpleTable
+titles[]:String -DefaultTableModel model:DefaultTableModel -jt:JTable -jsp:JScrollPane -column:TableColumn
<<oppConstructor>>+SimpleTable() <<oppConstructor>>+SimpleTable(col:String[]) +addStringArray(input:String[]):void +addStringVector(input:Vector<String>):void +removeAllRows():void

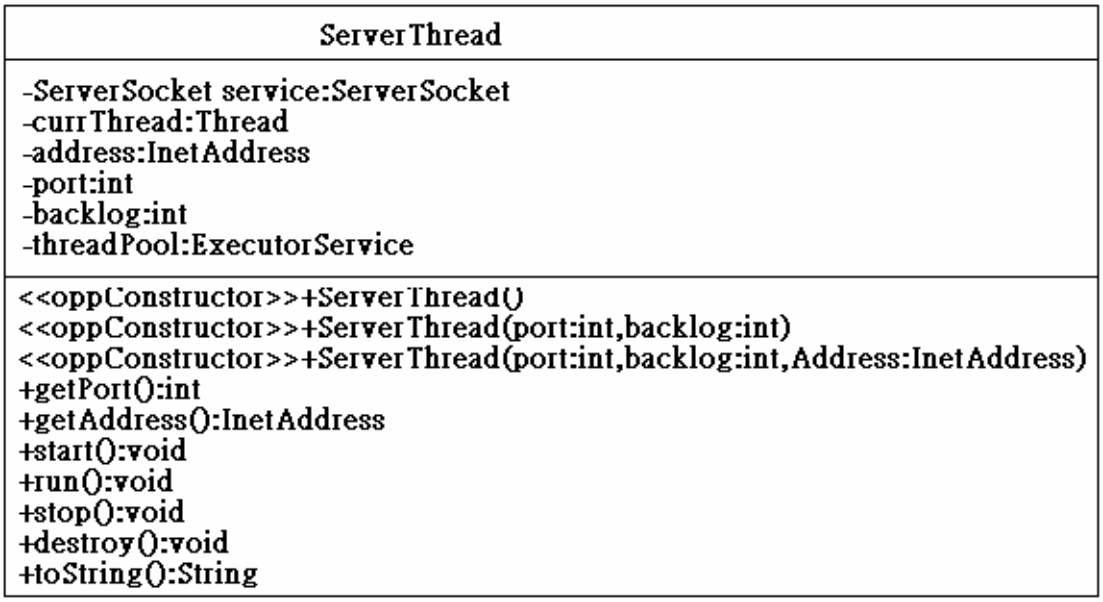

MessagePacket
-subSystemType:String
-sequenceNum:int
-netflowSequenceNum:int
-type:String
-mccHost:InetAddress
-mccHostPort:int
-InfoList:Vector<Object>
-IIDList:Vector<String>
-message:String
-InOutMessage:String
<<oppConstructor>>+MessagePacket(sn:int,type:String,message:String)
<<oppConstructor>>+MessagePacket(sn:int,type:String)
<<oppConstructor>>+MessagePacket(subSystemType:String,sn:int,type:String)
<<oppConstructor>>+MessagePacket(sn:int,type:String,message:String,mccHost:InetAddress,mccport:int)
<<oppConstructor>>+MessagePacket
(subSystemType:String,sn:int,type:String,message:String,mccHost:InetAddress,mccport:int)
<<oppConstructor>>+MessagePacket(subSystemType:String,sn:int,netsn:int,type:String,
message:String,mccHost:InetAddress,mccport:int)
<<oppConstructor>>+MessagePacket(copy:MessagePacket)
+getFunction():Object
+setFunction(Object:Object):void


Command
-command:String
-arguments:Vector<Object>
<<oppConstructor>>+Command()
<<oppConstructor>>+Command(cmd:String)
+setCommand(cmd:String)
+getCommand():String
+getArguments():Vector<Object>

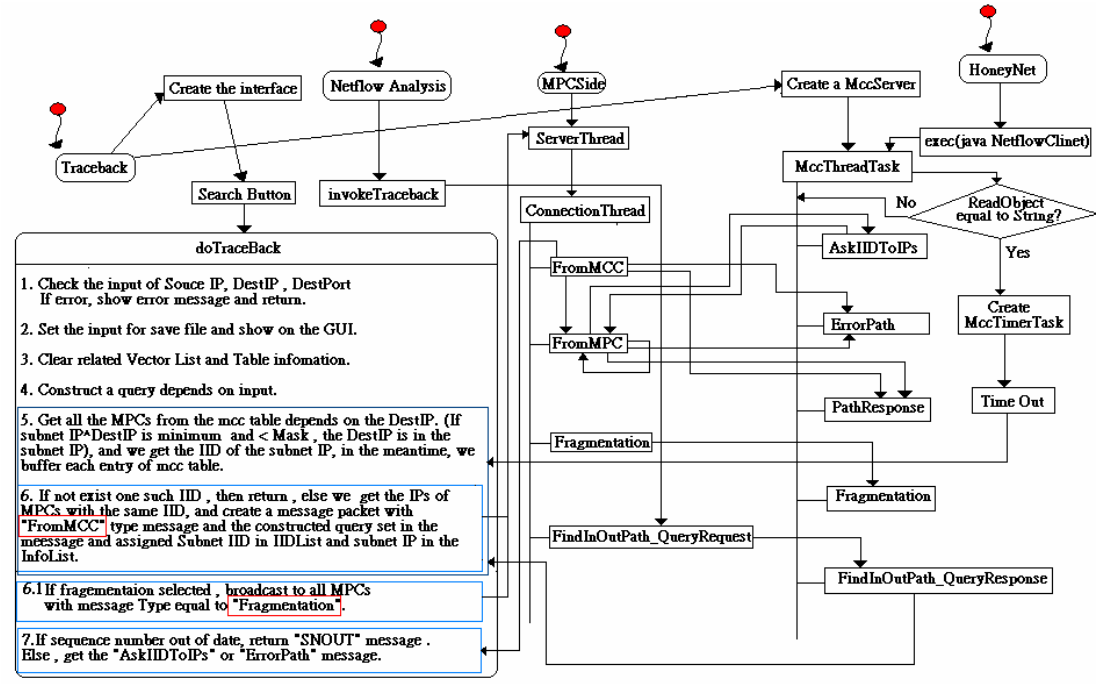

MccEntry
-IID:String
-subnet:String
-IP:String
<<oppConstructor>>+MCCEntry()
<<oppConstructor>>+MCCEntry(IID:String, subnet:String,IP:String)
+getFunction():Object
+setFunction(Object:Object):void


TAMEntry
-STime:String
-ETime:String
-SIP:String
-DIP:String
-Protocol:String
-DPORT:String
-IIDNUM:String
-IID1:String
-IID2:String
-IID3:String
+getFunction():Object
+setFunction(Object:Object):void


MPCSide
+main(args:String[]):void


ServerThread
-ServerSocket service:ServerSocket
-currThread:Thread
-address:InetAddress
-port:int
-backlog:int
-threadPool:ExecutorService
<<oppConstructor>>+ServerThread()
<<oppConstructor>>+ServerThread(port:int,backlog:int)
<<oppConstructor>>+ServerThread(port:int,backlog:int,Address:InetAddress)
+getPort():int
+getAddress():InetAddress
+start():void
+run():void
+stop():void
+destroy():void
+toString():String

ConnectionThread
-socket:Socket -inputStream:ObjectInputStream -outputStream:ObjectOutputStream -currThread:Thread -port:int
<<oppConstructor>>+ConnectionThread(socket:Socket,port:int) +stop():void +fromMCC(SQLstring:String,pkt:MessagePacket):void +fromMPC(SQLstring:String,pkt:MessagePacket):void +doFragmentationResp(SQLString:String,pkt:MessagePacket):void +findInPath_QueryResponse(destIP:String,pkt:MessagePacket):void +findOutPath_QueryResponse(sourceIP:String,pkt:MessagePacket):void +close_connection():void +run():void



Note:

1. There are three subSystemType, “Netflow”, “Honeynet”, “” (Think as “Traceback”).
2. In Traceback, it contains sequeceNum for traceback, honeynetSequenceNum for honeynet, netflowSequenceMain and netflowSequenceNum for netflow, netflowSequenceNumMain is used to count the request from netflow, netflowSequenceNum is times of doTraceback in netflow. The sequence number is used to tell the out of date of the request or response from the MPC.
3. Each subSystemType has its correctPathList and errorPathList and correct simpleTable and error simpleTable data structure. For

Netflow, it also contains FindInPath_QueryList and FindOutPath_QueryList.

4. Each coming request buffer the related query information and time stamp , and later new request one need to save the previous request do traceback results and related information into a file and into the according directory (./Traceback , ./NetFlow_Analysis , ./HoneyNet).
5. The query is set by MCC, and it set the query in message field of MessagePacket Object.

6. The previous one (MCC or MPC closer to Destination) set the IID and subnet IP of later one (MPC closer to Source) into MessagePacket Object.

7. When the system was closed, the system will save all the unsaved results and then close the system.

The following is detail description:

Traceback contains the interface and create a MccServer. MccServer is a server which use to accept all the connections from the outside. If one client connects to the MccServer, MccServer create a MccThreadTask object to serve the connection. The server thread first to read the object from the client, it's the MessagePacket Object if the client is from the MPCSide, it's the String Object if the client is from the HoneyNet. If MessagePacket, first we check the corresponding sequence number and corresponding subSystemType (Netflow Analysis, HoneyNet, "" is the Traceback). If the sequence number is out of date compare to the sequence number in the server, the server just send "SNOUT" message packet to client tell the request is out of date. If pass the sequence number check, we serve the packet according to the type field, it has four message type. "AskIIDToIPs", "ErrorPath", "PathResponse", "Fragmentation" and "FindInOutPath_QueryResponse" type.

First we describe AskIIDToIPs. MPC will get the IID1 or IID2 from tam table with the query send from previous MPC in message field of MessagePacket Object, after get IID1 or IID2, the MPC client need to ask the MCC what the IP of MPCs under the IIDs. So the MCC server will get the IIDs from the MPC client and get each entry of IIDs from the mcc table and send back to MPCs.

Now we describe ErrorPath. Each time when the MPC ask the database with the query which initially get from MCC, if the result is no answer about the query, the MPC need send the message packet back to MCC tell it I was the last one. Note that the message packet always copy the packet receive from MCC or previous MPC , and each time the MPC get the new MPCs IP(closer to source), it need set the subnet IP and IID of new MPC in the message packet and send to new MPC. For example, A (source) <- B <- C <- D (destination). D get the subnet IID of C, so it ask MCC the IP of C and it set subnet IID of C to IIDList of MessagePacket, and subnet IP to InfoList of MessagePacket, and then set the type equal to FromMPC and then send to C. If query result is no entry, it means we are the last one, so it sends a ErrorPath MessagePacket back to the MCC. We eliminate the error path which is subset of correct path when the cancel button is clicked and the error paths which not eliminated from correct paths is the real error path.

Now we describe PathResponse. When the MPC query the tam table and get the entry with A 0 0, it means its the correct path and the MPC send path response type MessagePacket to MCC. MCC save the path response.

Now we describe the Fragmentation. If MPC get the Fragmentation MessagePacket, it query to tam table get the IID1 of first one entry of the results and set to IIDList of MessagePacket and send back to MCC. MCC get the Message Packet, get the IID and query to mcc table and get the

corresponding subnet IP of the IID and show the results.

Now we describe the FindInOutPath_QueryResponse. When the user pushes the button from Netflow interface, it will invoke the invokeTraceBack function in Traceback.java. We just get the source IP, so we need to send FindInOutPath_QueryRequest to all MPCs, each MPCs will invoke findInPath_QueryResponse and findOutPath_QueryResponse . FindInPath_QueryResponse will query tam table with query DIP = sourceIP and FindOutPath_QueryResponse set SIP = sourceIP, they will get the results, the result may duplicate, so we need to reduce the duplicate. And send all the results(tamEntry is entry of tam table) back to MCC with MessagePacket type equal to FindInOutPath_QueryResponse and InOutMessage field set equal to “In” if in FindInPath_QueryReponse function, and “Out” if in FindOutPath_QueryResponse function. MCC get the message and also need to reduce the duplicate and buffer the message in FindInPath_QueryList and FindOutPath_QueryList accordingly, both are vector data structure. And then it create the NetFlowButton object with the message. So the user can push the button and the actionPerformed event will trigger and do traceback with the query information in NetFlowButton object.

Now we describe Message from the HoneyNet. The Honey will connect to MccServer and send four strings which are subSystemType (“honeynet”), SIP, DIP and DestPort. The server gets the first object (here is a String). If it’s a string, we can tell it’s from honeynet. And we get the following 3 strings (SIP, DIP, DestPort). Traceback has a timer and a TimerTaskList (vector) which use to buffer the MccTimerTask Object. We get the 4 strings and create a MccTimerTask object set with the 4 string and time stamp, and schedule the MccTimerTask in the Timer with HONEYNET_DELAY+SCHEDULE_INTERVAL, HONEYNET_DELAY (second, default is 60 second) tell the delay cause

by packet Listener program, and SCHEDULE_INTERVAL (second, default is 30 second) is the execute time of doTraceBack. If the later request come in 90 second, the scheduler will schedule with interval equal to 90 second, which means the previous one will exec 90 second and then the later one execute. If later one comes after 90 second, the later one will increase the Honeynet sequence number and save the results and information of previous one and clear related buffer and start to do traceback.

SIM

MCC

- editor.java
- Connector.java

SIM

| Profile builder
    - SIM_Calculator.java
| Early warning
    - FTA.java
    - EarlyWarning.java
    - EditDistance.java
    - HostClustering.java
    - K_medoid_Algorithm.java

CSIM

- SIMMain.java
- SIMPacket.java
- SIM_Processor.java
- Host.java
- Link.java
- Pair.java

- Profile.java

- Rule editor

Rule editor is placed on MCC. It receives the authentication identity from MCC and decides which rule database of subnet the user can access. Rule editor will directly modify the rule database of the subnet that user has access privileges. In addition, when a rule is added or updated, rule editor will perform an SSH commend to the particular SIM server for generating XML document of that rule.

  - Connector.java
    Used to connect to the rule database for the rule editor.
  - editor.java:
    Core code of the rule editor.

- SIM

The functionality of SIM server is to perform anomaly detection and then produce early warning list. However SIM does not communicate with MCC immediately, but store the anomaly list in its database and upload the early warning lists to CSIM through the SCP program.

  - SIM_Calculator.java
    Constructing profile and comparing the rules in database to generating anomaly host list

  - FTA.java:
    Performing Fault Tree Analysis

  - EarlyWarning.java
    Generating early warning list

  - EditDistance.java
    Calculating the similarity of host according to their profiles

  - HostClustering.java
    Clustering the hosts according to their edit distances through K medoid algorithm

- K_medoid_Algorithm.java
  Implement the K medoid algorithm

- CSIM

CSIM is the abbreviation of Center SIM, which collects the abnormal list from each SIM and response the request from MCC. When MCC send a request to obtain anomaly lists, CSIM will generate NodeList and EdgeList from the database in SIM server. On the other hand, when MCC send a request to obtain early warning lists, CSIM will directly response the list received from each particular SIM server.

- SIMMain.java:
  Create the network socket and listen to the requests from MCC

- SIMPacket.java:
  Define the structure of SIMPacket that used as the message format between CSIM and MCC. SIMPacket is inheriting MessagePacket and adding a member called “Object” which makes transmitting data possible.

- SIM_Processor.java
  Obtaining node and edge list from database according to the requested abnormal type from MCC.

- Host.java
  The host list data structure.

- Link.java
  The edge list data structure.

- Pair.java
  The data structure of Pair.

- Profile.java
  Constructing profile from data warehouse and convert the data format into JTree which is used to be shown in MCC.

附錄十、論文發表

Windows誘捕網情蒐之核心技術與回追技術軟體發展

趙禧綠[1]、趙梨華[2]、許富皓[3]、顏志豪[4]、蔡天浩[4]

[1]交通大學資訊工程學系教授

[2]交通大學資訊工程學系研究生

[3]中央大學資訊工程學系教授

[4]中央大學資訊工程學系研究生

摘要

日新月異的網路技術快速發展，已將網路的便利性與普及性帶到了前所未有的高峰，然而建構於這些技術之上的新型態攻擊也層出不窮。傳統的入侵偵測系統是透過以往的經驗來檢驗是否出現攻擊的跡象，但對於這種新發現的弱點便無法達到很好的保護效果。本研究之誘捕系統針對Windows作業環境採取一種不同的思維，誘使攻擊者對其進行攻擊。在攻擊的過程中，系統將秘密地記錄下整個攻擊的過程，回報並進行入侵源追蹤。這種誘捕系統提供了詳實的資料來源，將會成為資安人員在分析攻擊上的一大利器，也會對駭客造成一定程度的壓力。

在追蹤入侵來源技術方面，對於偽造來源位址的封包，由於現今網路傳輸的機制無法做到提供路徑追蹤的功能，所以本篇論文提出一個方法去設計並且實作封包標記技術，將路徑資訊記錄於封包中，能透過封包內的路徑紀錄，進行回溯追蹤，即使是偽造來源位址的封包也可回溯至來源端。本研究主要使用Identification欄位和Flags欄位的保留設定位元，將經過的每一個封包，進行封包標記，並記錄每筆標記資訊，最後利用這些記錄進行路徑的追蹤。

關鍵字：網路攻擊，蜜網，蜜罐，誘捕技術

The Research and Development of Honeynet with IP Traceback

Hsi-Lu Chao[1]、Li-Hua Chao[1]、Fu-Hau Hsu[2]、Chih-Hao Yan[2] and Tien-Hao Tsai[2]

[1]Department of Computer Science,
National Chiao Tung University

[2]Department of Computer Science,
National Central University

Abstract

Honeynet adopts a different concept, which induces an attacker to attack it. The system could record and report all the process of attack, including attackers' networking behaviors. Such systems collect detailed and extensive information. In the project, we focus on Honeynet, invasion profiling technique, and will integrate above-mentioned techniques, implement a complete Honeynet and risk estimating system, and verify its resistance by imposing common network attacks.

Traceback mechanism does not work well for packets sent from a spoofing IP address. In this paper, we propose a packet marking technique which can deal with this problem. We utilize the Identification field and reserved bit of Flags field in IP header to fill in the IID values of MPCs which packet had traveled. These marked records are buffered and periodically uploaded to a database of the MPC. The stored information can be retrieved to obtain routing paths.

Keywords: network attack, Honeynet, Honeypot, intrusion detection

1. 前言

網路誘捕技術是一種欺騙駭客攻擊的技術，它被用來吸引入侵者，使他們進入受控的環境之中(Data Control)，並使用各種監控技術來捕獲入侵者的行為(Data Capture)。網路誘捕技術的核心是設置一套具備高度隱匿性的監視機制，並記錄入侵者的活動資訊(Data Collection)，以針對其行為模式進行分析(Data Analysis)，進而透過網路追踪技術，定位到原始的入侵者(Trace Back)。

網路誘捕系統則是應用網路誘捕技術，設置數台主機甚至是一個複雜的網路系統，透過這套誘捕系統可以將駭客侵入系統的一切資訊，包括攻擊過程、使用的工具等全都記錄下來，以供分析之用。事實上，網路誘捕系統往往也具有混淆駭客選定攻擊目標的作用，以對真正運作中的伺服器，提供絕佳的掩護功能。

現行針對網路誘捕系統的研究有兩大類，其一是蜜罐(Honeypot)，另一則是所謂的蜜網(Honeynet) [1]。一般而言，蜜罐是一種具備誘捕能力的電腦系統，它藉著置放某些具備吸引力的資源，來吸引駭客的攻擊。通常蜜罐系統不修補系統的安全漏洞，以使入侵者有極大的發揮空間，以期擷取他們更多的訊息。但是由於傳統的蜜罐系統，並未與自身的網路系統有所區隔，一旦蜜罐被攻破，入侵者反而會利用這個蜜罐系統作為跳板，攻擊其他的系統，因此近年此一領域的發展，逐漸導向了蜜網的概念。

雖然蜜網也是由蜜罐所組成，但它不是單一的系統，而是一個網路。一個典型的蜜網系統通常由防火牆、路由器、入侵偵測系統(IDS)及數個蜜罐系統所組成，其中防火牆及入侵偵測系統可以對進出蜜網的資訊，執行資料控制(Control)與捕獲(Capture)的任務，並據以獲取入侵者的基本資料。蜜網內部可以設置多種類型的作業主機(如Linux、Solaris、Windows、FreeBSD等)來充當蜜罐系統，以提供入侵者一個更加真實的網路環境的感受。藉著各個主機系統所提供不同伺服器的作業環境，也將易於探索駭客所使用的工具及其手法。

有關入侵源追蹤技術方面，現今網路已擁有許多通訊協定，去進行傳輸，在網路傳輸的過程中，有一些攻擊方法專門針對傳輸設計的缺點去進行攻擊，進而達到資料的竊取、修改和偽造等行為，為了對這些攻擊採取對應政策，我們需要去偵測出攻擊和攻擊行為模式且去追蹤來源路徑。為了能夠得知攻擊者位置，制定出許多專門對路徑做查詢的協定，但是攻擊者也針對這些協定，想出相對應的方法，使得被攻擊方無法正確找出攻擊者。現存的通訊協定，大多以來源位址進行路徑的追蹤，當接收到攻擊者的封包，就能夠利用封包中的來源位址，去查詢兩端連線的傳輸路徑，但攻擊者卻可偽裝來源的位址使被攻擊方無法找出攻擊的來源，無法追回的原因是在於攻擊者的攻擊封包過來的路徑，沒有任何記錄在封包上，而是利用額外的封包去找出攻擊來源位址所經過的路徑，變成攻擊的封包與進行追蹤的封包是不同路徑，為了解決此問題，後續有人提出要對封包進行標記，在進行的路徑中，能夠在封包加上路徑資訊，使得被攻擊方能夠利用這些封包找出攻擊來源，而不再使用額外的封包去進行路徑的追蹤。

要進行封包標記的實作，除了現行的網路架構不變之外，也無法改變目前已經有的通訊協定，因此要進行標記，必須考量到能夠利用或使用的部份。以不增加網路流量為前提，就有人提出使用IP header中的Identification(ID) 欄位，而這個欄位是不常用，並非是不使用，因此還是會影響到目前的通訊協定，ID欄位正式名稱是識別碼，每一個IP封包都會有一個識別碼，當產生的數據經過網路傳送時，都會在傳送層被拆散成封包形式發送，之後封包要進行重組的時候，就是以識別碼為依據，如果覆蓋掉這個欄位，會導致原先已有的一些設定失效。除了變更已有的欄位外，另外是以IP header所保留的Options為主，使用所保留的Option設定，將所要進行的標記內容放置到自行設計的Options內，使路徑資訊以解析Options欄位就能夠取得，使用Options欄位會使封包的大小增加，但不影響目前的通訊協定是它的優點，而相反的會使網路流量增加，而且許多防火牆都防止IP Option的使用，可能導致封包在進行傳送時，遭受到丟棄。此篇論文設計並實作一使用IP header中的Identification欄位和Field欄位的保留設定位元達到封包標記與路徑追蹤兩目標，其原因有兩點：第一、經過切割的封包，對網路的效能有不利的影響，所以大部分路由器都具備有自動偵測MTU 的機制，送出封包時便已符

合MTU 的規定，封包表頭並不需要用Identification 欄位進行封包切割，據最近的研究，只有低於0.25%的網路總封包量才會經過切割而必須使用Identification 欄位[15]。第二、利用現存的IP 封包表頭欄位，而非Option等自創欄位，才能無接縫的適用於所有現存的路由器[15]，避免封包遭到丟棄和增加網路流量。且為了不影響到切割封包的重組，對無需切割封包和切割過封包採取不同處理方式和入侵源追蹤呈現方式。

2. 文獻探討

Honeynet Project 的創始人 Lance Spitzner 給出了對蜜罐的權威定義：蜜罐是一種安全資源，其價值在於被掃描、攻擊和攻陷 [2]。這個定義說明蜜罐並無其他實際作用，因此所有流入/流出蜜罐的網路流量都可能透露了掃描、攻擊和攻陷的惡意行為。而蜜罐的核心價值就在於對這些攻擊活動進行監視、檢測和分析 [3]。

資料顯示，蜜罐技術的發展歷程可以分為以下三個階段：

從 90 年代初期蜜罐概念的提出至 1998 年左右，“蜜罐”僅止於一種概念，通常由網路管理人員運用，透過欺騙駭客達到追踪的目的。這一階段的蜜罐指的是一些真正被駭客所攻擊的主機和系統 [4]。

約自 1998 年起，蜜罐技術開始吸引了一些資訊安全研究人員的注意，並開發出一些專門用於欺騙駭客的工具，如 Fred Cohen 所開發的 DTK（欺騙工具包）、Niels Provos 開發的 Honeyd 等，同時也出現了像 KFSensor、Specter 等一些商品化的蜜罐系統。這一階段的蜜罐亦可以稱為「虛擬蜜罐」，即這些蜜罐工具能夠模擬成虛擬的作業系統和網路服務，並對駭客的攻擊行為做出回應，從而欺騙駭客 [5]。

虛擬蜜罐工具的出現使得佈署蜜罐變得方便許多。但是由於虛擬蜜罐工具存在著互動性(Interactive)低、較容易被駭客識別等問題，從 2000 年之後，研究人員更傾向於使用真實的主機、作業系統和應用程式來建置蜜罐，但與之前不同的是，融入了功能更為強大的資料捕獲、資料分析和資料控制的工具，並且將蜜罐納入到一個完整的蜜網體系中，使得研究人員能夠更方便地追踪侵入到蜜網中的駭客並對他們的攻擊行為進行分析。尤其是一些研究用的蜜罐具有較高的交互性，被設計成能夠捕獲駭客的鍵擊記錄，瞭解到駭客所使用的攻擊工具及攻擊方法，甚至能夠監聽到駭客之間的交談，從而掌握他們的心理狀態等資訊 [6]。

蜜網實質上與傳統蜜罐技術的差異在於，蜜網系統構成了一個駭客誘捕的網路架構，在這個架構中，我們可以包含一個或多個蜜罐，同時保證了網路的高度可控性，以及提供多種工具以方便對攻擊資訊的採集和分析。此外，虛擬蜜網亦可建置於虛擬作業系統軟體（如

VMWare 和 User Mode Linux 等）使得我們可以在單一的主機上實現整個蜜網的體系架構。虛擬蜜網的引入使得建置蜜網的代價大幅降低，也較容易佈署和管理 [7]。

封包標記的機制，最早提出的是 Probabilistic Packet Marking (PPM) [8]，主要的設計是藉由收集標記封包的內容去建立出攻擊端到受害端之間的整條攻擊路徑，在這個概念被提出後，根據不同的標記方式分為三種實作。

第一種是 PPM-node append[9]，與其他 PPM 實作方式比較是最簡單又基礎的作法，標記方法是將每一台路由器的位址都寫入傳送的封包中，當一個封包到達接收端時，接收端可依據封包路由器資訊所寫入的順序及位址，直接建立出封包所傳送的整條路徑。它所使用的標記是將路由器的位址寫入 IP Option 欄位，每傳送到一台路由器，就接續上台路由器標記位置添加上去。這樣的作法經過大量的路由器時，會造成封包增加的量太多，而造成網路傳輸額外的負擔增大，並且無法判定是否還有空間可寫入，攻擊者也可在 IP Option 中增加偽造的位址而誤導路徑的建立。

第二種是 PPM-node sampling[9]，從 PPM-node append 的實作進行了修正，標記方法是在封包的標頭保留一個 node 欄位(32-bit)，每經過一台路由器時，會由一定的機率 P，當 P 大於一個臨界值時，才會將路由器的位址寫入 node 欄位，將空間不足的問題解決及網路傳輸的負擔減少。但是由於是採用機率 P 進行標記，因此接收到距離為 d 的路由器的位址機率為 $P(1-P)^{d-1}$，每次接收端只能接收到其中一台路由器的位址，因此需要收集大量的封包才能達到路徑追蹤的效果。而路徑追蹤則是需要傳送 sample 封包來進行排序動作，來達到建立有次序的整條路徑。這個方式針對單一攻擊者擁有很大的成效，但是不適用於有多條攻擊路徑的攻擊者，而且也需要大量時間去進行排序的動作。

第三種是 PPM-edge sampling[9]，再為 PPM-node sampling 進行改進，增加了 edge 的資訊在封包上，變更為新的 start、end 和 distance 三個欄位。Distance 欄位是記錄所標記的路由器到受害端的距離，start 欄位是記錄所標記的路由器的位址，而 end 欄位是記錄標記後的下一個路由器的位址，所以當一個封包被決定標記時，會將路由器 A 的位址寫入 start，distance 設為 0，到達下個路由器 B 時，會將路由器 B 的位址寫入 end 欄位，而 distance 欄位每經過一個路由器就會加一，因此就能夠推出路由器的距離。這次改良變成可以針對多重路徑攻擊者，除了由 distance 欄位了解它在攻擊路徑上的位置，也可由 start 及 end 建出路由器前後的關順序，加強了排序的時間也繼承了 PPM-node sampling 的優點。

相對於建立完整的路徑，也有以找出最靠近攻擊者的路由器為目標，而不建立出完整的

路徑，而 Deterministic Packet Marking（DPM）[10]就是以此為目標，DPM 相對於 PPM，它是最容易實作，也是最不增加路由器負擔的一種作法，它不需額外保留欄位來進行標記，而是使用現有的 IP 標頭內的 17bits(ID field and the reserved 1-bit flag)，當封包一進到網路時就讓它進行標記，標記的方法是將原本的路由器位址切割成兩部份(每個 16 bits)，每次標記時，放入不同的部份，而放入的內容可以由 flag 欄位內的 0 或 1 來判定，當兩個部份都收集到後，就可以組合出原本的路由器位址，推導出最靠近攻擊者的那台路由器的位址，而不找出整條路徑。優點是不用每台路由器都進行標記的動作，大大減少路由器的負擔，但是多個攻擊者使用同一來源位址或每次攻擊封包設定不同的來源端位址時，無法有效找到攻擊者。

根據 DPM 的缺點，DPM-with address digest[11]則改進了 DPM，將原本的 DPM 概念加上雜湊函數與 ingress 路由器來區別攻擊者，欄位變成 Address fragment、Hash digest 以及 Index 三個，將原本的 IP address 切割成更多區段，而使用雜湊函數讓所有封包通過相同的路由器擁有相同的 identity，而受害端利用這個 identity 去進行 IP address 組合。延展了 DPM 的優點，增加能夠區分多個攻擊者，但必然的摘要碰撞將導致有些合法來源端會被誤判，成了它的缺點。

在前兩個主要目標都是在於建立路徑，然而也有封包標記主要是為了阻擋 DDoS 攻擊而進行設計，主要目的在於區分出是來自相同的傳輸路徑，Pi Marking Scheme[12]就是以此為目標，使用了 IP header 內的 ID field 欄位(16 bits)，將 ID 欄位分為 16/n 個區段，當封包每經過一台路由器，就會將路由器的 IP address 中的 n bits(n=1~2)寫入 ID 欄位其中一個區段，因此 ID 欄位一次可記 8 台或 16 台路由器的個別 n bits，而放入的區段是由 TTL 的數值除以 16/n 的餘數所計算出來的，之後藉由 ID 欄位的值，就可判斷出是否來自不同的來源封包攻擊，若是判斷出是惡意攻擊的封包，就由相同的 ID 欄位去進行封包過濾的動作，使惡意攻擊的封包無法傳送，達到防止 DDoS 攻擊。但是只要中途有一台路由器沒有支援 Pi Marking Scheme，而 TTL 經過這台時，會進行遞減，而使標記的動作跳過 n bits，而造成誤判的結果。

接續 Pi Marking Scheme 的方法，StackPi Marking Scheme[13]為改進 TTL 缺點，而變成採用堆疊的方式將資訊標記上去，以達到改善不支援的路由器問題。其它的設計方式都與 PI 相同，而在 ID 欄位滿時，將舊的標記丟棄，加入新的標記進去。也就是放置的方式就像排隊一樣，先來的在前，後來的在後，而滿時就會將前面的丟棄，而補進後面的，如此一來在 Pi Marking Scheme 的 TTL 缺點就改善了。

Router Interface Marking (RIM) [14]，RIM也是源於PPM的概念，去建立出攻擊者到受害端的攻擊路徑，RIM機

制在每個封包經過每台路由器時，會根據某一個機率*P*之下，去進行標記的動作，標記的動作主要是將路由器進入的介面 (Interface)、相對應的距離 (Hop數)以及收集這條路徑上其他路由器的資訊，在以不增加封包額外的overhead，額外的訊息封包，以及影響路由器的負擔前提之下，達到接收端能夠由這些所收集的封包，建立出不同的惡意攻擊的封包路徑，以樹狀的方式去呈現出來，因此可以針對DDoS的攻擊，做出攻擊路徑的重建，並且能夠採取額外的措施去防止攻擊。

3. SQL Injection的弱點攻擊研究

SQL injection 是常見的一個重大威脅，SQL injection 是利用輸入特殊命令，讓系統將之與標準的資料庫查詢程式和資料合併在一起，送給資料庫管理系統執行，因此有一段有害的程式碼被正常的程式碼包裝起來形成『隱碼』，直接對資料庫存取資料或進行破壞，進而造成資料庫損毀或資料流失。SQL injection 本身不僅可以由網頁上的欄位(HTTP POST method)發起攻擊，也可以直接從附加在 URL 變數以 HTTP GET method 送出。此外，更有相當多的變形(如子字串組合)與編碼手法(如 unicode)，攻擊者可以技巧性的繞過網站上之防護措施。因此針對未知之 SQL injection 攻擊字串之分析實為一大挑戰。

3.1 SQL Injection的弱點形式

(1) 未做適當的字元(串)過濾：

當 web application 的設計者沒有對來自 user 的 input 做適當的字元(串)過濾時，它的 input 會經由 web application 合成 SQL 查詢字串的程式，合成非預期，甚至是有害的 SQL 查詢字串。

以下面的 SQL 查詢字串的合成程式碼來說明：

```
statement = "SELECT * FROM users WHERE
name = '" + userName + "';"
```

情況下，若以下列字串

```
a' or 't'='t
```

就會合成出下列的 SQL 查詢字串：

```
SELECT * FROM users WHERE name = 'a'
OR 't'='t';
```

認證的話，由於't'='t'永遠是 true，將導致身份認權限。

許一次的呼叫可以執行多道 SQL 的查詢命令，有些 SQL 的 API 有做到限制這樣的行為，像是 php 的 mysql_query 這個 SQL 查詢的 API，這樣可以避免攻擊者嵌入獨立且與原本程式無關的 SQL 查詢字串。

(2) 不正確的變數型態處理

SQL injection 也有可能發生在沒有做正確的變數型態處理的 web application 中，例如：

```
statement := "SELECT * FROM data WHERE
id = " + a_variable + ";"
```

要是一個正整數的值，以查詢與其相符的 id 的資

否為整數的檢查時，當下列的字串

執行 drop(delete) users 這個 table，產的 SQL 查詢

可能存在弱點，像是 MySQL server 的

用 unicode 的編碼方式，繞過程式設計者的過濾，

(4) blind SQL injection

當 SQL injection 的結果不會呈現在網頁的頁面上時，攻擊者有可能使用以下的方式來做 injection 有沒有成功的檢查：

I. 條件的respones

其中一種 blind SQL injection 的方法是在 select 敘述句中加上永遠成立，或是永遠失敗的條件式，如下所列：

```
SELECT booktitle FROM booklist WHERE
bookId = 'OOk14cd' AND 1=1;
```

藉此來判別是否有 SQL injection 的弱點或是 SQL

injection 是否成功。

II. 條件的errors

這個種類的 blind SQL injection 在 where 條件式成立的時候會產生 SQL 的 error，例如：

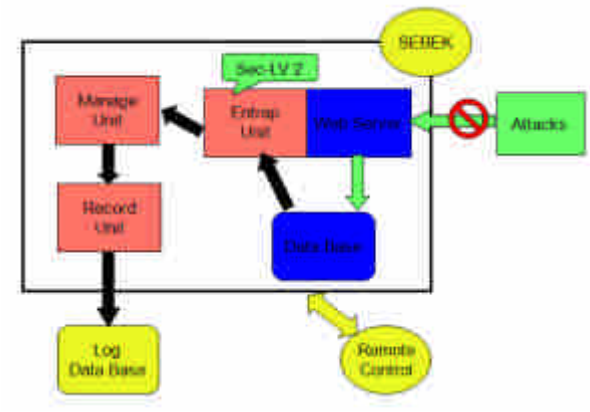
```
SELECT 1/0 FROM users WHERE
username='Ralph';
```

會造成除以零的例外，而攻擊者能夠藉此來找出

行的 query，用 web server 回傳結果時間延遲來判斷是否有 SQL injection 的弱點或是 SQL injection 是否成功，快速回應可能表示 SQL injection 失敗，過一段時間才回應可能表示 SQL injection 的注入碼有被成功執行。

4. 蜜網主機與誘捕機制之設計

本研究之實作環境，採用 VMware 虛擬機軟體架設一台蜜網主機及三台誘捕主機（蜜罐），並分別於各誘捕主機中佈設數個具備誘捕機制的網站。蜜網系統整體架構如（圖一）所示，其中蜜網主機(Net Controller)包括三個網路介面，eth0 連接外部網路，eth1 連接蜜罐系統，而 eth2 屬於一個秘密通道，連接到一個監控網路。這種架構的系統核心是建構在資料控制和資料捕獲的基礎功能之上，再納入了遠端 GUI 管理及資料分析整合的功能，形成一個具備情蒐能力的誘捕環境。



圖一、蜜網系統架構圖

蜜網系統架構如圖一所示，主要分為幾個部分：

a. 含有SQL injection弱點的web application & database

本研究計畫為建立一個網頁型態之 honeypot，以資訊安全討論區的形式呈現，在登入處理的頁面中設有 SQL injection 的弱點，引誘攻擊者入侵，藉以記錄其所做的行為。

b. Entrap unit

這個部份是用來做使用者 input 的檢查和 server response 的檢查以及 security level 的模式設定與切換的實作，若有攻擊者用 sql injection 的方式，成功執行惡意的 SQL 指令，則經過使用者 input 的檢查和 server response 的檢查，偵測到此一惡意行為，我們會將 security level 的等級調升。

c. Manage unit

這部份的設計，為因應可以做 remote control，從遠端能夠調整 security level。

d. Record unit

為了記錄攻擊者藉由 SQL injection 入侵的所有行為，需要一個記錄的機制，來完成記錄行為的動作，這個部分會將這些行為記錄下來存到 log database 中，且基於安全的考量，log database 須和 web application 所使用的 database 分開放置。

e. Log database

記錄攻擊者行為模式的資料庫。

5. Security Level

一個惡意使用者想要攻擊網站的應用程式，無法繞過的限制就是必須要有可以輸入攻擊字串(input data)的欄位(input fields)，以及接收字串的程式。所以我們為了降低監控的成本，將整個網站僅留下登入畫面可供使用者輸入帳號以及密碼(其他頁面將不接受未登入用戶的任何輸入)，當有惡意使用者嘗試利用 SQL Injection 等手法來繞過 honey pot 網站的登入檢查時，此程式會紀錄來源 IP Address 並且分析其行為(input data 與 input field 之間的關連性)，來即時決定 Security level。security level 越高，代表網站對於入侵者的手法越難以防範；此時則會將程式碼的檢查難度提升，讓入侵者認為網站管理者已對程式漏洞做修補，並誘使他採取其他方法來入侵網站。

(1) security level 1 (record the log)

當用戶登入時，誘捕程式除了一般的帳號密碼檢查之外，會額外地取出 Database 內的資料再次與登入資料做比對。例如：正確資料為 「peter / 1234」，攻擊資料為 「“ or1=1 / 1234」，攻擊資料雖能以 peter 身分登入，但藉由再次比對帳號/密碼，即可區分正常與惡意的登入。當發現到惡意登入，誘捕系統就會記錄此次攻擊的 log：來源 IP、時間、攻擊的頁面、

攻擊字串等。並且將此 IP 放入黑名單當中，當同樣 IP 下次登入時會依照之前分配的 security level 來阻攔。

而攻擊字串會先以 HTMLencode 將特殊字元轉換成 HTML 的編碼符號字元，以避免攻擊字串可能含有 SQL 語法，直接寫入 Database 可能會造成錯誤。

(2) Security level 2 (add slash)

當 security level 提升到 2 時，會檢查此使用者所有輸入的資料中是否含有 quote (')、double quote (")、backslash (\)等字元，將此類字元前面加上 backslash(\)。因此在網站程式處理輸入資料時，將會看到(\')、(\")或(\\)，並且將其視為一般的符號，不會被誤用來中斷正常的 SQL 語法。

(3) Security level 3 (filter all error message)

攻擊者有時候會因為輸入字串過濾了特殊字元，導致無法成功的進行 SQL-Injection，通常會改以 Blind SQL-Injection 的方式，藉由 SQL Server 傳回的錯誤訊息來猜測 Database 的各種資料，如資料表名稱、欄位名稱等。因此，設計了 Level 3 用來阻擋 SQL Server 送出的錯誤訊息。

(4) Security level 4 (replace all special characteristics)

這是目前的最高安全度，會將所有輸入字串內的任何特殊字元都去除掉，也就是只會接受數字(0~9)和字母(a-z, A-Z)。以語法上的認知，這些字元是無法組成任何攻擊字串；但是，禁用符號雖然可以保證安全，卻無法達到 Web application 的多樣性，所以一般網站並不會使用此類的極端手法。

6. 封包標記之方法

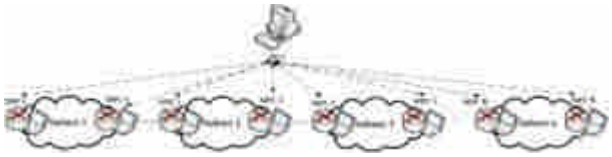
為了能夠達到少數封包也能夠進行追蹤的效果，以及不影響目前的通訊協定，我們自訂了封包標記方式以及記錄方式，去達到路徑追蹤的目標。

封包標記分為四個部份來說明，第一部份整體架構，說明整個網路環境的設置，第二部份標記設計，說明標記內容各個欄位的設定，第三部份標記偵測與記錄，說明偵測的方式與記錄的內容，第四部份是搜尋與結果，呈現畫面與路徑的相關設計。

6.1 設計環境

為了使封包進行標記，在每個子網的 gateway 附近，加入一台能夠進行標記的主機，我們稱它為 Marking PC(MPC)，當封包傳遞經過 MPC 時，會將封包的標頭進行標記動作，然後

再將修改過後的封包進行轉送的動作，放置 MPC 的優點是不需更改路由器的設定，能夠以原本的網路拓樸去進行設定，而封包能夠增加標記資訊到封包上。

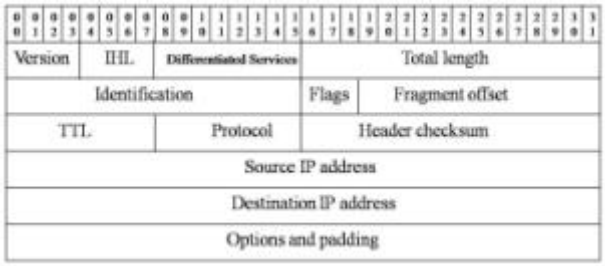


圖二、架構圖

不都在路由器間架設 MPC (如圖二所示)，只在子網的 gateway 附近架設 MPC，是基於部建成本的考量，而且這樣的架設由於不會影響原先的設定，可以將 MPC 放置在任意地方。架設的環境不只可在路由器附近，也可以在防火牆或其他設備之間，能夠任意放置 MPC 位置，並且進行標記，也不需要連續的放置，是設計上的最大考量。

6.2 Option 封包標記

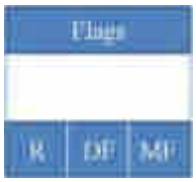
標記設計是使用目前已有的 IP Protocol，不更改它的欄位設定，並且使用 Identification 欄位和 Flags 欄位的保留設定位元。IP Protocol 中，IP 標頭欄位的設定如圖三，前 20 bytes 為每一個封包必定要有的資料，而 Options 是在需要特定的控制時，才會利用到，Padding 則是在 IP Header 不為 32 bits 倍數時，進行補足的位元。



圖三 IP Header

我們的設計是使用 Identification 欄位的 16 bis，做為我們標記資訊放置的位置，且使用 Flags 欄位的保留欄位，有 1bit 來識別是否以經標記過。Flags 總共佔 3bits，如圖四，主要與 IP 封包的切割與重組有關，第一個 bit 為保留用途用，預設值為 0，第二個 bit 為 DF(Don't Fragment)，用來定義 IP 封包是否可以加以切割，第三個 bit 為 MF(May Fragment)，用來定義

此 IP fragment 是否為原始封包的最後一個 IP fragment。我們針對欄位的設定結果如圖五。



圖四 Flags 欄位設計　　圖五 Identification 欄位

各個欄位的說明如下：

(1) Identification: 16bits，只有對未切割過封包，才會將此欄位分成兩欄，分別記錄第一個經過的 MPC 的 IID 值，和目前所處的 MPC IID 值，範圍 0~255。

(2) IID：8bits，標記內容，只有對未切割過封包才會在 Identification field 標記 IID 值，每經過在 gateway 附近的 MPC，就會將 MPC 特定的 IID 值放置在這個欄位，在相同網域內的 MPC 會有相同 IID 值，不同網域間，設定不同的 IID 值，IID 的範圍是從 0~255，然而因為 IID 要有未使用時的數值，因此將 0 保留下來，當作還未填 IID 資料進去，因此真正能夠使用的 IID 就只有 1~255，如圖六。

圖六 標記示意圖

(3) R：有兩個功用，對於未切割過封包是用來判斷是否已經經過第一個 MPC 標記過，對於切割過封包則是用來讓監聽程式判斷是否需要將相關資料寫進資料庫內。

在為每一個封包進行標記時，除了更改 IID 值外，也要針對特定欄位進行修改。Header Checksum 欄位代表標頭檢驗值，當我們修改了 IID 欄位的內容後，檢驗值必需要重新計算，放入一個修改過後的新值，避免在後續的傳送中，封包遭到丟棄。

在不同網域的 MPC，我們會設定不同的 IID，使每個網域都擁有自己的 IID，在 IID 欄位我們使用了 8bits，所以 IID 的範圍是從 0~255，然而因為 IID 要有未使用時的數值，因此將 0 保留下來，當作還未填 IID 資料進去，因此正式能夠使用的 IID 就只有 1~255。

6.3 標記偵測與記錄

封包偵測必須要針對每一個封包的 Flags 的保留欄位，進行解析動作。對於未切割封包，Flags 的保留欄位是用來判斷封包是否已經經過第一個 MPC 標記過，若保留欄位值為 0，表示目前所處的 MPC 是此封包經過的第一台 MPC，會在標記後，再將標記內的所有 IID 及相關資料，記錄到資料庫內；若保留欄位值為 1，會在標記前，將標記內的所有 IID 和本身 IID 及相關資料，記錄到資料庫內。對於切割過封包，Flags 的保留欄位是用來判斷封包是否已經經過出口端 MPC 的監聽程式將所需資料寫進資料庫，若保留欄位值為 0，表示目前所在的 MPC 是出口端，監聽程式需將本身 IID 及相關資料，記錄到資料庫內；若保留欄位值為 1，則監聽程式無須將任何資料寫進資料庫。

在每一台 MPC 上，我們都會架設一個 MySQL 資料庫，並且設計一個 Table，使監聽所偵測到的標記內容寫入到資料庫的 Table 內，而 Table 欄位設定如圖七。

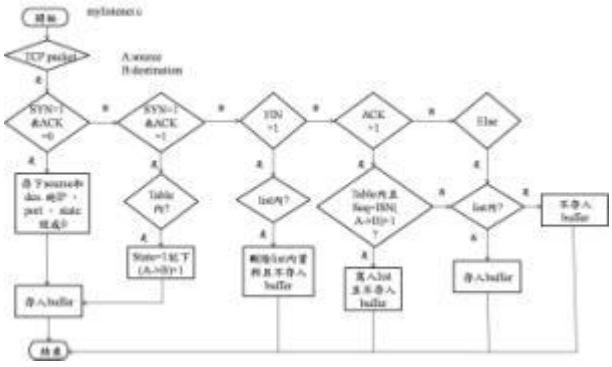
STime 與 ETime 兩個欄位的格式型態是 DATETIME，用來記錄日期以及時間，STime 全名為 Start Time，記錄這個標記內容的起始時間，而 ETime 全名為 End Time，記錄這個標記內容的結束時間，SIP 為來源 IP，DIP 為目的 IP，Protocol 為 IP 通訊協定，DPORT 為目的 Port，IIDNUM 為封包內標記的 IID 數量，IID1、IID2、IID3 代表的是經過的 MPC 的各個 IID。

圖七 MySQL 的 Table 設計

每當一個封包經過 MPC，具有標記的內容如果直接寫入資料庫，會使記錄資料大量產生，短時間內擁有數筆相同的數據資料，為了將這些重覆資料在短時間內能夠集合成同一筆資料，我們在監聽的部份加入檢查 TCP 封包是否完成三向交握程序和 Buffer。藉由檢查 TCP 封包是否完成三向交握程序，再決定是否將其記錄到 Buffer。將沒完成三向交握的連線，記錄為需要寫進 Buffer 的連線封包；若完成三向交握程序的連線，在此連線傳送的封包將不予以記錄到 Buffer，藉此可減少寫入資料庫的資料量，演算法見圖八。當標記資料寫進 Buffer 後，使標記的資料能夠在短時間保留在 Buffer 內，直到 Buffer 滿載或者一段時間後，再寫入資料庫，以減少短時間內相同資料的產生。Buffer 是使用動態產生，並且擁有一個數值限制 Buffer 的最大量，並且可以設定保留在 Buffer 的時間。

當一個標記資料進入 MPC 後，它會先將這筆標記資料記錄放到 Buffer，並且會在 Buffer 內的 STime 與 ETime 寫入現在的時間與其它數值，當下一筆標記資料進入時，在 Buffer 中如果也擁有這筆相同資料時，它會將 ETime 更新為目前時間，直到 STime 超過我們所設定的時間限制，就會寫入資料庫。或者是目前的 Buffer 已達到最大量，我們就將 Buffer 中 STime 最舊的那一個，寫入資料庫。

假如是切割過的封包，即使沒做標記，出口端 MPC 的監聽程式依舊要將其寫入資料庫，作類似 logging 的動作，且 IIDNUM 為 1，IID1 為本身 IID，而 IID2、IID3 皆為 0。



圖八 監聽程式判斷三向交握流程圖

監聽程式判斷三向交握演算法說明如下:

監聽程式會記錄哪些 flow 已經完成三向交握程序，並將此 flow 的 source IP、source port、destination IP 和 destination port 存入陣列，以供監聽程式判斷封包是否需要先記錄到 Buffer 的依據。

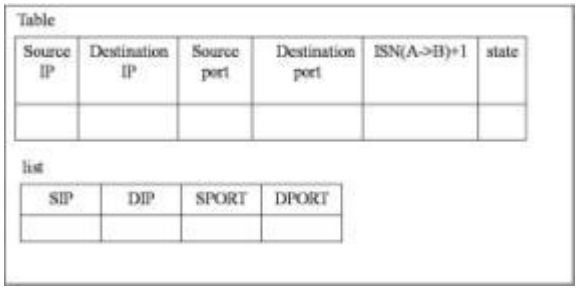
(1) 若監聽到的為 SYN 封包，則將 Source 和 Destination 的 IP 、port 存入 table(由六個一維陣列組成，見圖九)，寫入 Buffer 暫存區。

(2) 若監聽到的為(SYN+ACK)封包，則先檢查在 table 內是否有記錄，若有，則修改相對應的欄位，將 state 設為 1，且記錄下此次的 ACK 值在 ISN(A→B)+1 欄位，寫入 Buffer 暫存區。

(3) 若監聽到的為 FIN 封包，則先檢查在 list 內是否有記錄，若有，則刪除 list 內資料，且不存入 Buffer。

(4) 若監聽到的為其它 ACK 封包，則先檢查在 table 內是否有記錄，若有，再檢查 Seq 是否等於記錄在 ISN(A→B)+1 的值，若相等，則將 source 和 destination 的 IP、 port 記錄在 list 內，且不存入 Buffer，若不相等，則檢查 list 有無記錄，若無，則寫入 Buffer，若有，則不寫入 Buffer。

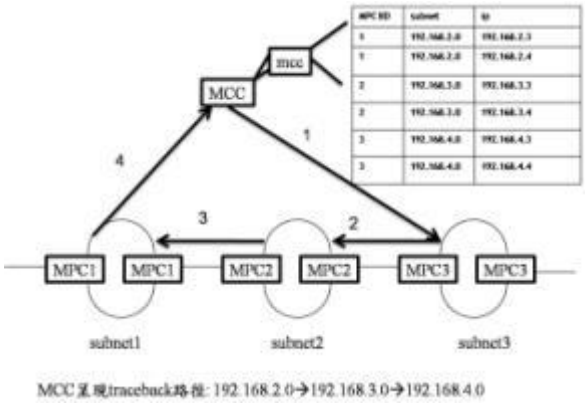
(5) 若監聽到的為其它封包，則檢查 list 有無記錄，
若無，則寫入 Buffer，若有，則不寫入 Buffer。



圖九 Table 和 list

7. 路徑追蹤

路徑追蹤的方式由一台中央控管主機(MCC)發出查詢，在 MCC 會建一個資料庫，有所有 MPC IID 和 subnet、IP 的對照表，由輸入的條件，包括封包的 destination IP，由 destination IP 得知向特定 subnet 裡的所有 MPC 做資料查詢，MPC 依照中央控管主機所設定的查詢條件，由本身的資料庫找到前一個 MPC IID 值，再跟前一個 subnet 裡的所有 MPCs 作查詢，陸續往回查詢相關的 MPC，直到此封包經過的第一台 MPC，即可將資料回傳至 MCC 呈現，路徑的呈現方式是以 subnet 呈現。



圖十 路徑追蹤方式

從圖十中，了解路徑重建的步驟。

Step1:MCC 依據查詢條件 destination IP，知道要向 subnet3 裡的 MPC3 做查詢。

Step2~3:MPC3 根據資料庫內的資料，知道要向 MPC2 做查詢，MPC2 根據資料庫內的資

料，知道要向 MPC1 做查詢，因為 MPC1
是第一個經過的 MPC，所以查詢做到此即
可。

Step4: MPC1 作回傳資料的動作，回傳路徑
MCC。

找出整條路徑的 IID 後，需要將 IID 轉換為 subnet，這時藉由中央控管主機內的 MPC IID 與 subnet 相對應的記錄，將 IID 轉回所處的 subnet。

有關於切割過的封包，則是由MCC向全部MPC查詢是否有此筆記錄，找出出口端subnet，在MCC呈現出入口subnet。

8. 路徑追蹤實作結果

實作的環境設置，使用了兩台 Host 主機，三台有加上自動路由軟體 Quagga 的 MPC，將他們串連成一直線如圖十一，在兩台 Host 主機進行封包傳輸(TCP 連線)，中間的 MPC 進行轉傳的動作，MPC 則是將封包進行標記，在這三台 MPC，第一個 MPC 的 IID 設為 1，第二個 MPC 的 IID 設為 2，第三個 MPC 的 IID 設為 3，MPC 在傳輸過程中所執行的動作是 Router，再加上封包標記的功能，封包從一台 Host 主機開始傳送，經過第一台 MPC3 時，會執行以下動作。

(1) 檢測封包是否經過切割
(2) 若是切割封包，則檢查 Flags 欄位的 R-bit 是否
為 1，若是，則直接離開，若不是，則將 R-bit
設為 1，重新運算 checksum，然後離開。
(3) 若是未切割封包，檢測封包是否已經擁有第一筆
標記記錄，若無，則加入第一筆標記內容，並將
R-bit 設為 1，重新運算 checksum，然後離開，若
有，則在第二個 IID 加入標記內容。

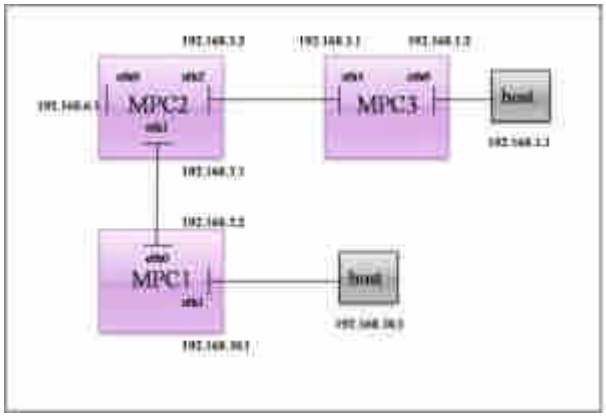
在 MPC3 中，會將 IID 為 3 的標記放入 Identification 欄位內的 IID1 欄位，之後進行轉傳的動作，傳給 MPC

在 MPC2 中，會將 IID 為 2 的標記放入 Identification 欄位內的 IID2 欄位，將 3 取代，之後進行轉傳的動作，傳給 MPC1。

在 MPC1 中，會將 IID 為 1 的標記放入

Identification 欄位內的 IID2 欄位，將 2 取代，之後進行轉傳的動作，傳給另一台 Host。

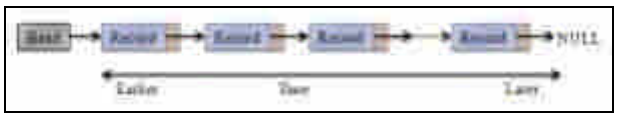
送到另一台 Host 端後，就達成這段的傳輸，而原本的封包內容，只有改變標頭的部份，因此不影響現有的網路傳輸及傳輸的內容。



圖十一 環境設定

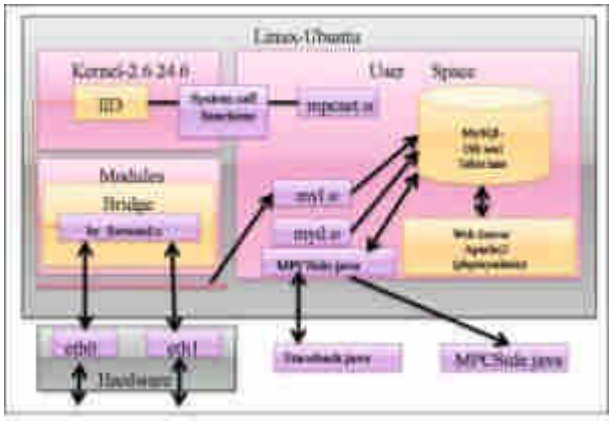
封包在進行傳送的動作，進行記錄的則是做標記的 MPC，封包從 Host 發送經過第一台 MPC 時，會將封包進行標記，標記完後，在轉送的動作發生之後，MPC 會先作三向交握的判斷，再決定是否將這個封包的標記內容先記錄到暫存區內，當傳送的第一筆資料記錄到資料庫之前，會有暫存區暫存現在傳輸的標記內容及相關資訊，這個暫存區會將記錄保留 60 秒，當相同的封包標記及封包內容再次進到暫存區時，會更新這筆原本已在暫存區的資料，而不再額外進行記錄，以減少重覆性的標記記錄。

暫存區的設計是採用 Queue 的方式，如圖十二所示。基本以保留 60 秒為主，但為了防止暫存區資料過多，因此設定了一個最大限，暫存的資料量達到最大限制的量時，它就會將最早的記錄加入資料庫，所以記錄加入資料庫的情況有兩個，第一個是保留 60 秒時間到就將記錄寫入資料庫，第二個情況則是封包傳送量過多，而使標記擁有許多不同的資料，在暫存區資料爆滿時，將記錄寫入資料庫。



圖十二 暫存區的資料結構

在 MPC 上安裝的作業系統是 Linux-Ubuntu，而必須搭配 Bridge module 和 MySQL 軟體，最後加上監聽封包的程式，所有軟體及程式安裝結果的 model 如圖十三。



圖十三 MPC 主要軟體內容

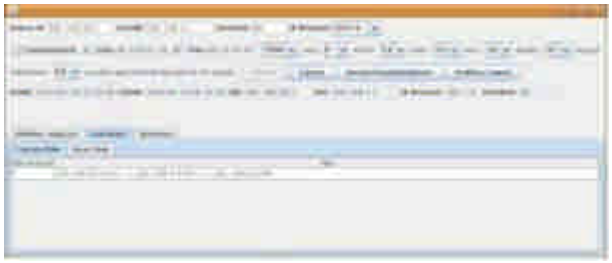
在標記、監聽以及記錄之後，我們使用 phpMyadmin 進行資料庫的管理，我們將 table 建入資料庫，執行結果如圖十四。

圖十四 MySQL 資料庫內的表格格式

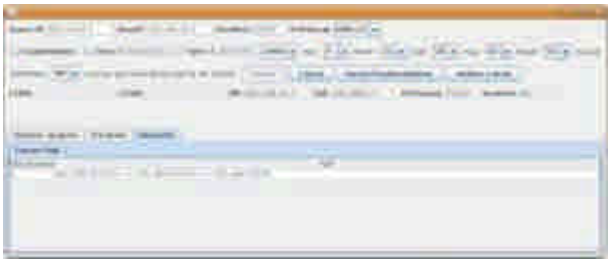
每當記錄要寫入資料庫，就會依照資料庫 table 所設定的欄位一一填入，當沒有使用到的 IID 則用預設數字 0 填入，其資料的記錄結果如圖十五所示。

圖十五 資料庫資料內容

由資料記錄可發現，相同的來源及目的端，以及相同的連結 Port，每隔一分鐘才會有一筆記錄，大幅減少在短時間內的記錄量，而相同的來源及目的端，使用不同的 Port 時，會產生不同的記錄，再加上有判斷是否完成三向交握程序，又再度縮減寫進資料庫的記錄。當想知道某個封包所經過的路徑，可在 MCC 的 GUI 介面進到 Traceback 的畫面，直接手動輸入欲查詢封包的相關資訊，Source IP、Destination IP、Destination Port、Protocol 為必要輸入條件，其它條件為可選取式，如圖十六。亦可從 Honeynet 接收查詢請求，自動呈現封包經過的路徑於介面上，如圖十七。



圖十六 主控端執行畫面



圖十七 主控端執行畫面

9. 結論

在入侵源追蹤技術方面，會有封包標記方法的產生是由於現有提出的一些路徑追蹤方法，是使用額外的封包，根據接收到的封包來源位址，進行回送查詢，但是如果攻擊者是偽造 IP 攻擊，傳送過去的封包有可能與接收到的封包是不同路徑，則無法使用此方法進行路徑追蹤。為了不增加網路流量，我們使用封包表頭的

Identification 欄位和 Flags 欄位的保留設定位元進行封包的標記，且不影響現有的網路通訊協定，不更改現有的網路設備，達成隱密性的效果，並使用 IID 取代原本的 IP address，以減少標記所需要的空間，同時也需要相對應的對換表，找出完整的傳送路徑。

本研究議題所研發的誘捕技術，也可以視同一種「陷阱機制」，其目的是將入侵者導向我方所設置的資源環境之中，並利用該環境對入侵者的行為進行紀錄、分析，甚至加以追蹤。一個成功的陷阱機制常能使入侵者對其侵入過程產生若干迷惑，因而增加其入侵行為的負荷，提高其達成目標的複雜度及不確定性。這種設計的最大效用，在於可藉以瞭解與學習入侵者的思維邏輯、使用的工具和入侵目的。透過擷取其入侵手法，更能有效檢視我方網路的安全性及所面臨的威脅，從而制訂更有效的防護機制。

10. 誌謝

感謝國家科學委員會及國防部軍備局中山科學研究院提供本計畫之補助及支援，及所有協助 98 年度國防科技學術合作計畫成果發表會的相關人員提供學術交流平台。

11. 參考文獻

[1] G Yang, CM Rong and L Peng, "A Novel Approach for Redirecting Module in Honeypot Systems", J. China Univ. of Posts and Telecommunications, V12, N3, Sep., 2005.

[2] L Spitzner, "Honeypots: Tracking Hackers", ISBN:0-321-10895-7, Addison Wesley, 2002.

[3] R Grimes, "Honeypots for Windows", ISBN:1-590-59335-9, Apress, 2005.

[4] M Meijerink, J Spellen, "Intrusion Detection System honeypots", University of Amsterdam, Feb. 13, 2006.

[5] C Döring, "Improving network security with Honeypots", Master's thesis, University of Applied Sciences Darmstadt, 2005.

[6] C Viecco, "Improving Honeynet Data Analysis", Proc. 2002 IEEE Workshop on Information Assurance and Security, US Military Academy, West Point, NY, 17 - 19 June 2002.

[7] AE Avila, "Analyzing Intrusions of a Hybrid Vitural Honeyney", Master Thesis, UT El Paso, 2005.

[8] T. Baba and S. Matsuda, “Tracing network attacks to their sources,” IEEE Internet Computing, vol. 6, pp. 20-26, Apr. 2002.

[9] S. Savage, D. Wetheral, A. Karlin, and T. Anderson, “Practical network support for IP traceback,” in Proceedings of ACM SIGCOMM’00, vol. 30, pp. 295-306, Oct. 2000.

[10] A. Belenky and N. Ansari, “IP traceback with deterministic packet marking,” IEEE Communication Letters, vol. 7, pp. 162-164, Apr. 2003.

[11] A. Belenky and N. Ansari, “Tracing multiple attackers with deterministic packet marking (DPM),” in Proceedings of IEEE Pacific Rim Con. Communications, Computers and Signal Processing, vol. 1, pp. 49-52, Aug. 2003.

以履歷為基礎之網路行為輔助監控系統

彭文志[1]、黃俊龍[1]、廖忠訓[2]、李政輝[2]、張翔任[2]、楊濬仲[2]、蔡金亮[2]

[1]交通大學資訊工程學系教授

[2]交通大學資訊工程學系研究生

摘要

網路的發展使生活便利卻也造成網路犯罪崛起。一個自動化網路威脅分析與預警系統能幫助資安專家有效分析網路威脅。在真實世界中每台主機會有不同的連線情況，因此我們提出一套以網路連結履歷為核心的預警系統，該系統包含資料倉儲、資安法則定義與編輯介面、預警機制以及風險量化分析模組。我們利用防火牆日誌來建立主機的履歷，藉以監控主機行為，並利用資安法則驗證介面將系統偵測到的異常行為以圖形化的方式呈現給系統管理者，並讓系統管理者透過資安法則定義與編輯介面操作本系統，本系統透過半自動方式減輕系統管理者負擔，並有效提升資訊安全。

關鍵字：網路安全，資料探勘，履歷，風險分析

A profile-based network security remote monitoring system

Wen-Chih Peng, Jiun-Long Huang, Zhung-Xun Liao, Zheng-Hui Lee, Hsiang-Jen Chang, Chun-Chung Yang and Chin-Liang Tsai

Department of Computer Science,

National Chiao Tung University

Abstract

The development of Internet causes our life more and more continent but also leads to more Internet crimes. To help network security experts for analysis the menace from Internet, we need an automatic analyzer and pre-warning system. In the real world life, different hosts would have different connection behavior, thus we proposed a pre-warning system which is based on network usage profiles. This system consists of data warehouse, rule editor, pre-warning, and risk analysis modules. We adopt the firewall log to construct the profiles of hosts which is used to monitor the behavior of hosts. In addition, the rule editor could help experts to customize new rules, and the abnormal hosts will be displayed on the graphical manner. We design and implement a semi-automatic system which abates the burden of system managers and enhances the safety of networks.

Keywords: security, data mining, profile, risk analysis

1. 前言

過去對於網路攻擊及異常行為之偵測大多需要事先找到攻擊或異常的樣式(pattern)或特徵(signature)，對於已知的攻擊行為可有效的隔絕，但對於新型態的攻擊手法，因為事先並無相對應的pattern及signature，便無法在第一時間進行偵測；另一方面，大多數的IDS系統皆是被動的偵測，而無法預先得知可能的攻擊行為而進行主動預警。因此，本系統將利用主機連線行為的規律性(regularity)，來進行偵測，當主機之行為異於平常之規律性，便有可能是遭受攻擊或行為異常。圖一為網路上一伺服器之連線數量統計圖，由圖中可以看出主機在連線上具有一定的規律性。而對於預警方面，我們則將分別利用失誤樹及分群法來計算主機或群組的風險程度，對可能遭受攻擊的主機進行預警。

2. 系統架構及元件介紹

2.1 系統簡介

本計畫開發網路威脅風險分析與威脅預警之核心技術與軟體單元，實作2-tier之系統架構，設計一套以履歷為基礎之輔助系統，藉由防火牆的連線紀錄，可以不需要事先得知攻擊pattern及signature便能偵測出異常行為，同時藉由風險分析，我們可以在主機遭受攻擊或入侵前，事先加以預警。同時，我們也將加強法則編輯，讓使用者可以自行定義法則，透過人性化的使用者介面，輸入自定規則，並且在法則瀏覽器中顯示法則及異常連線記錄。

2.2 2-tier系統架構

本系統之目標為製作一套輔助管理者決策暨制定資安規則之系統，並且利用視覺化的顯示介面來達成網路威脅分析及風險量化，將防火牆日誌分解成包括Source、Destination、Port、Time等欄位，系統根據給定之規則防護並監測主機的安全。使用者可利用查詢工具和網路連結履歷(Profile)將詳細連線資料取出分析，配合時序資料之輔助來找出難以發現的攻擊，判斷是否有異常行為發生。

圖二為本系統之硬體架構，其中SIM_1~SIM_4分散於各子網路中，負責偵測網域內的異常主機，並將偵測之結果送至Master SIM上，當使用者由MCC登入後，便能直接由Master SIM取得異常資料。

圖三為本系統的軟體架構圖，其中包含：資料倉儲及資料探勘、法則編輯器、預警機制及風險量化分析。將於後面的章節分項逐一說明介紹。

2.3 主機履歷分析

我們所建立的履歷為一階層式的結構，初始的網路連接履歷為空，接著從防火牆日誌(firewall log)中取得每個主機的IP位置和網際網路服務後，會先建立出如圖四之樣示，並在others紀錄此IP所有連線數目和連線型態(如Http、Ftp)。

然而當某項服務的連線數值(count)大於使用者所定義的門檻時(Count Threshold)，會在網路連結履歷(profile)新增一項個別的節點(node)如圖五所示，將FTP服務從others分離出來。經過一段使用者所定義的時間(Time window size)後，整體的網路連結履歷(profile)會變成如圖六，其中因為排版空間的限制，我們只顯示In link的分支，另外在Out link的分支其結構和In link是相同的，而每一層的定義將一一說明如下：

第一層：針對該主機所有的連結建立履歷。

格式: (IP, Pattern)

範例: (140.113.6.2, Pattern)

第二層：針對該主機的連出連結及連入連結分別建立履歷。

格式: (IP, Direction, Pattern)

範例: (140.113.6.2, IN, Pattern)

第三層：除了連結方向外，還會根據服務類別建立各自的履歷。

格式: (IP, Direction, Service, Pattern)

範例: (140.113.6.2, IN, http, Pattern)

第四層：除了連結方向及服務類別外，則會根據對方的IP位址建立個別履歷。

格式: (IP, Direction, Service, Domain, IP, Pattern)

範例: (140.113.6.2, IN, http, 140.112.*.*, 140.112.172.46, Pattern)

在每一層的履歷中，我們都會利用資料倉儲的技術來建立時間序列式的樣式(time series pattern)，如圖七。

3. 資料倉儲

3.1 介紹

資料倉儲為資料探勘中的一項技術，利用事前的運算來加速查詢的速度，雖然目前有許多包含資料倉儲的資料庫系統，但基於價格及和Java程式的連接性上之考量，我們選用MySQL做為後端資料庫，而以Mondrian做為資料倉儲的介面。為使Modrian能順利運作，我們必需對資料庫進行修改，加上時間標記表格，並且透過XML檔案來描述資料倉儲cube的各dimensions。

由於Log DB僅存放各別的連線記錄，當我們要查詢某一主機在某段時間內的連線總數時，資料庫需要針對不同的需求來進行查詢，對於程式的效率性較差，因此，我們利用資料倉儲來預先將可能會用到的數值計算出來，我們將計算各別維度，如:IP、Port及時間在不同解析程度，如：日、週、季、月、年，的連線總數，將來需要用到這些資料時，便能即時取用，而不需要再額外進行計算。

4. 法則編輯器

使用者可利用法則編輯器將自定的資安規則輸入，未來系統將會每日回報違反這些規則的主機，並且做為預警之用。資安法則定義及編輯器提供一圖形化的介面讓使用者可以輸入及設定自行定義之資安規則。我們將規則拆解成各種子規則的組合，如圖八為一利用兩項子法則(S_1及S_2)組合而成的法則(R_1)，S_1及S_2利用Union的方式組合。而相對的，圖九為多層次的子法則組合，其中，S_2和S_3先以Sub的方式組合成R_4，接著R_4再以Union的方式和S_1組合成R_3，最後R_3再和R_3以intersect的方式組合成R_2，而R_2即為我們最終的法則。利用這樣的拆解及組合的方式，我們可以讓使用者自行設計規則，而非只受限於已事先定義的法則項目。為了達成自定規則的功能，我們利用巢狀表格(nested table)的方式，在資料庫中記錄子法則(subrule)，再利用集合運算，聯集、交集及差集，來對子法組進行組合。如表一及表二分別為法則表格及子法則表格。其中每個internal node皆存在法則表格，而leaf node則存在子法則表格。再透過法則表格中的Operator欄位來記錄組合的方式。

雖然利用子法則拆解方式可以有效率的讓使用者自定法則，但由於一條規則被拆成好幾個部份，因此，在每次讀取規則時，皆需要進行一次組合，traverse整個rule tree才能將整個規則組合起來，對於較長或結構較複雜的法則來說，需要花費的時間也會較多。因此，我們利用XML文件同屬樹狀結構的特性，將每一條規則以XML的方式來描述，之後在每次讀取法則時，只需要讀取XML文件，而不需要組合整個rule tree。圖十為一以XML文件描述更新頻率異常法則的範例。

5. 預警機制

預警機制可以有主機尚未發生異常行為時，先行預測其將來有可能出現異常。預警機制利用資料探勘中的分群法(clustering)，將使用網路連線行為較類似的主機進行分群，當群中主機發生異常的比例過高時，則群中未發生異常之主機將被視為有可能發生異常行為，亦即為預警之對像。我們將取各主機當日之履歷，做為資料物件(data object)，進行分群。主要可以分為兩部份，其一為特徵選擇(feature section)，其二為相似度比對(similarity measurement)。在特徵選擇方面，初步我們將利用後序排列(post order)的方式將樹狀結構的profile轉成序列(sequence)，而此post order sequence即為該profile的特徵；而在相似度比對方面，我們將利用editor distance的方式來計算兩個sequence的相似度，editor distance為一利用dynamic programming的概念設計的pseudo-polynomial algorithm，在效率上有較好的表現。

當為各主機進行分群後，如圖十一所示，當群組內的主機發生異常，我們便能計算其異常的比例，做為群組的風險程度，當愈多的主機發生異常，其風險程度也愈高。

6. 風險量化分析

在風險量化分析這個部份，我們利用失誤樹分析(fault tree analysis)來進行，我們將每天利用失誤樹計算一次風險值，並將結果上傳至master SIM，以供未來MCC查詢，因此，每台主機每天會有不同的失誤樹的風險值。由於對於使用者新增的規則，其相對應的失誤樹並無法完全由資料倉儲中取得所需的資料，亦即，無法完全得知每個節點的機率值，因為，我們僅針對五項已知的資安規則進行計算，將這五棵失誤樹內進在程式中。

系統可以分為兩部份：(1)失誤樹編輯器及(2)失誤樹分析。以下分別介紹：

6.1 失誤樹編輯器

利用修改FaultCat來達到編輯失誤樹的功能，使用者可以輕易得利用滑鼠點選編輯失誤樹。圖十二為編輯器之介面。當使用者編輯完成後，系統將以XML格式儲存，並且做為接下來分析之用。

6.2 失誤樹分析

失誤樹分析模組主要針對使用者編輯完成之失誤樹XML文件進行分析，將XML文件還原成樹狀結構，並且利用Mondrian連結資料倉儲取得各失誤樹中節點之數值，由下往上算出風險值。

7. 網路威脅分析

在資安法則上，我們將針對五項資安法則進行格式轉換，將舊的法則格式轉換為XML型式，同時我們也將針對三種攻擊行為，分別為：port scan、warm及trojan，進行偵測，以下分別介紹這幾種異常及攻擊行為：

7.1 預先定義之資安規則

7.1.1 未更新之主機

如圖十三所示，為避免已知的漏洞被有心人士利用，每部電腦必需至update server下載並安裝patch，若發現有某些電腦長時間未連線至update server，則視為違反資安規則，亦即這些電腦極有可能存在資安漏洞。

7.1.2 未回報之主機

如圖十四所示，為避免稽核程式(agent)回報資訊，有些電腦或營區會安裝防火牆(firewall)阻檔回報的封包。因此，若發現有某些電腦或營區長時間未進行回報，則視為規避稽核，可能內部正進行違法的行為。

7.1.3 具有不安全服務(unsafe service)組合之主機

如圖十五所示，由資訊安全專家定義，哪些服務(service)不能同時存在一台伺服器(server)上。

7.1.4 未列管伺服器

如圖十六所示，營區內的所有伺服器都必需登記列管，不能私自架設伺服器。

7.1.5 使用未申請的Port

如圖十七所示，當使用者連往遠方不知名的Port時，有可能是遭到攻擊而將資料外洩。

7.2 惡意攻擊行為

7.2.1 掃描通訊埠攻擊

掃描通訊埠攻擊透過對目標主機傳送封包藉以探查目標主機可能存在的漏洞。這種攻擊通常是入侵的第一步，如果偵測到掃描攻擊將對維護資訊安全將有很大的助益。

遭受掃描攻擊時，主機會收到針對各種不同服務類型的連線，這些連線所要求的服務與履歷中所記錄的服務種類往往會有很大差異。我們的系統藉由比對履歷記載的服務與被要求的服務，便可發現掃描通訊埠攻擊。另外來自掃埠工具的攻擊會產生大量的連線，當超過InLink的安全限度時我們就將他視為遭受攻擊。

來自駭客的掃埠攻擊相較之下複雜許多。他們會有計劃地針對某些重點埠進行掃描並分散掃描時間降低被發現的可能，此種攻擊單是觀察連線數目難以發現，必須利用連線所要求的埠種類來輔助偵測。若是針對單一主機多個埠的攻擊，可以利用InLink所記錄的連線種類來判斷，當掃埠攻擊發生時，觀察時間內所累積的連線種類會比平常多，且不屬於此主機服務的要求也會上升。若是針對多主機單一埠的掃描，我們可觀察全域的被要求服務統計，若是某一個服務或是某一個埠被大量的要求就有可能是此類攻擊。

7.2.2 電腦蠕蟲

電腦「蠕蟲」跟病毒一樣，可將自身從一台電腦複製到另一台，但蠕蟲會藉由控制電腦上可以傳送檔案或資訊的功能自動複製。一旦您的系統中有蠕蟲存在，它就會自動蔓延。在蠕蟲蔓延的過程當中，它會嘗試進行大量的連結；例如利用網路上的芳鄰進行傳染的蠕蟲，它會自動偵測區域網路上的其他主機，並嘗試透過網路分享的資料夾進行傳染。感染蠕蟲的主機之連結數量會遠大於履歷中所記錄的行為，故我們的系統可以發現感染蠕蟲的主機，並將其歸類為異常行為。

7.2.3 木馬程式

一個完整的特洛伊木馬套裝程式含了兩部分：服務端（伺服器部分）和用戶端（控制器部分）。植入對方電腦的是服務端，而駭客利用用戶端進入運行了服務端的電腦。運行了木馬程式的服務

端以後，會產生一個有著容易迷惑用戶的名稱的進程，暗中打開埠，向指定地點發送資料，透過木馬程式，惡意攻擊者可以掌握受害者的電腦。

如果多台電腦皆感染同一隻木馬程式，則當惡意攻擊者要操控這些電腦時，會有許多主機同時收到不在履歷中的連線，因為這與履歷中所記錄的有很大差異，因此我們的系統可以發現並將其歸類為異常行為。

8. 結論

當網路流量愈來愈大，使用率也愈來愈普及，由目前資料探勘應用在掃描偵測、網頁日誌分析、防火牆規則分析及異常偵測的研究中，可以發現資料探勘已經逐漸在IDS上扮演重要的角色，透過資料探勘豐富的資源和理論基礎，無論是利用目前已有的方法或是開發新的探勘演算法，都很適合用來改善目前在攻擊偵測上的不足。

本系統利用防火牆日誌配合資料探勘的技術建立一套有效的入侵偵測系統，當內部主機有異常連線發生時，系統將此情形夠過GUI介面傳送給資安專家，並根據資安專家之判斷能使系統擁有最新的資料來防護主機之安全。我們設計了一套便於網路管理者使用的人性化操作介面，easy to use的設計理念讓即使非電腦或資訊安全專家也能輕鬆觀看內部網路連線狀況及各主機之網路連結履歷，並可運用介面提供的功能輕鬆編輯各種主機群組及資安規則。系統彈性亦為本計畫核心之一，除了預先定義之群組與規則類別，管理者亦可自訂新類別來設計特殊的資安規則，並由規則資料檢視介面瀏覽對應結果。

9. 誌謝

感謝中山科學研究院及其相關人員在計劃過程中的指導與協助。

10. 參考文獻

1. Jaeyeon Jung, Vern Paxson, Arthur W. Berger, and Hari Balakrishnan., “Fast portscan detection using sequential hypothesis testing“, IEEE Symposium on Securicy and Privacy, 2004.

2. C. Kruegel and G. Vigna, “Anomaly Detection of Web-based Attacks,”in ACM conference on Computer and Communications Security (CCS), 2003.

3. X. Wang, J. Zhou, S. Yu and L. Cai, “Data Mining Methods for Anomaly Detection of HTTP Request Exploitations,”in Fussy System and Knowledge discovery (FSKD), 2005.

4. K. Golnabi, R.K. Min, L. Khan and E. Al-Shaer,“Analysis of Firewall Policy Rules Using Data Mining Techniques,” in IEEE/IFIP Network Operations and Management Symposium, 2006.

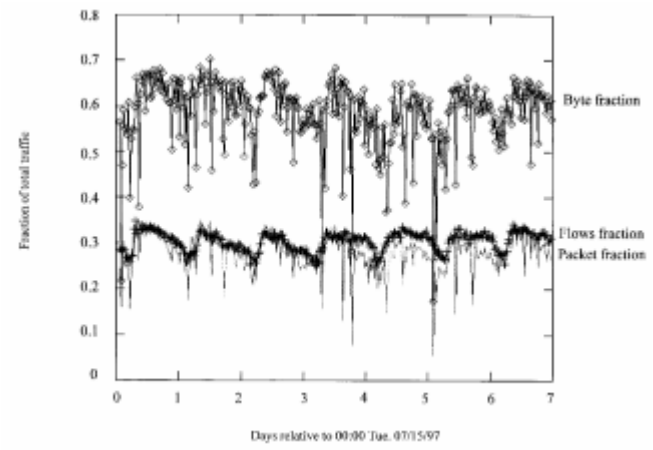
5. Aleksandar Lazarevic, Levent Ertoz, Vipin Kumar, Aysel Ozgur and Jaideep Srivastava, “A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection,” in SDM Conference, 2003.

6. T. Lane, C. E. Brodley, “Sequence Matching and Learning in Anomaly Detection for Computer Security,”in AAAI Workshop: AI Approaches to Fraud Detection and Risk Management, 1997.

7. K. Sequeira, M. Zaki, “ADMIT: Anomaly-base Data Mining for Intrusions,” in ACM SIGKDD

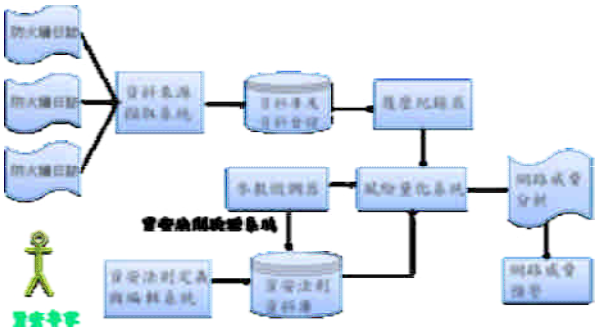
Conference, Edmonton, 2002.

11. 圖表彙整

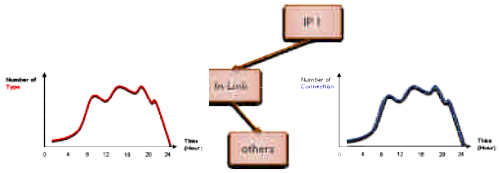


圖一、主機連線數量統計圖

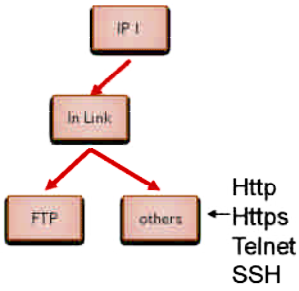
圖二、2-tier之系統架構



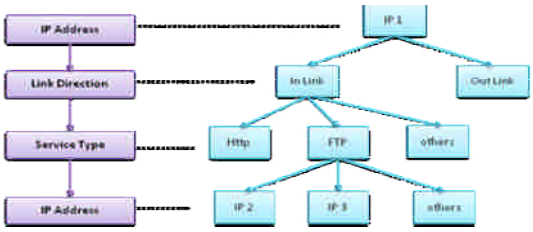
圖三、系統架構圖



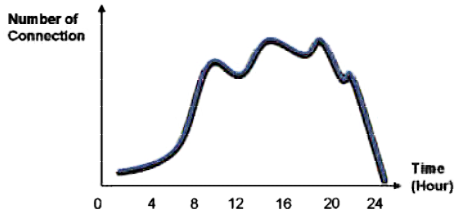
圖四、網路連結履歷初始圖



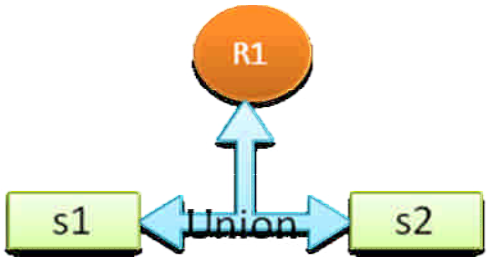
圖五、新增個別服務項目



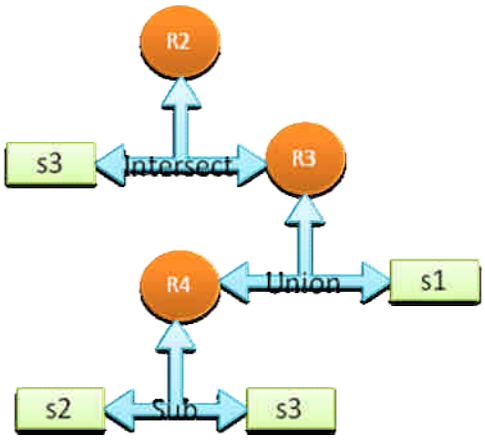
圖六、主機履歷格式



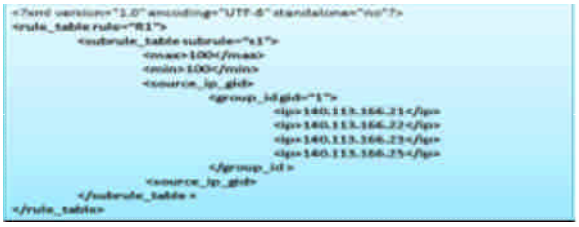
圖七、時間序列樣式



圖八、子法則拆解範例



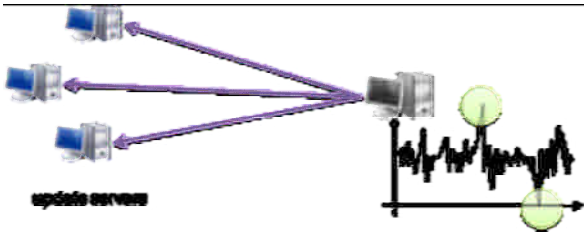
圖九、多重子法則拆解範例



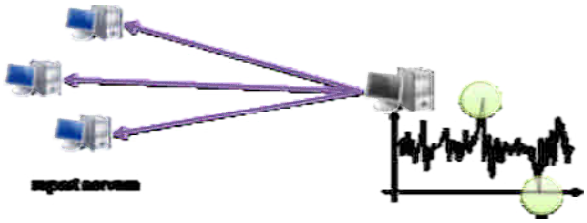
圖十、更新頻率異常法則之XML文件

圖十一、主機分群圖

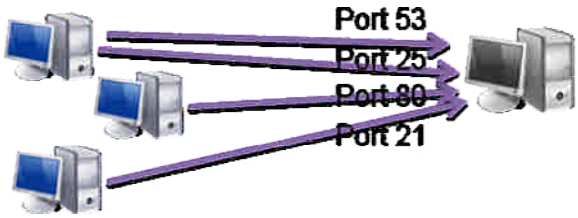
圖十二、失誤樹編輯器



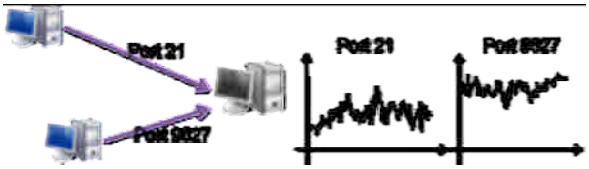
圖十三、違反未更新之主機法則



圖十四、違反未回報主機法則



圖十五、違反具有不安全服務組合法則



圖十六、違反未列管伺服器法則

圖十七、違反使用未申請的Port法則

表四十五、法則表格

Rule	Root	Left	Right	Operator	Comment

R1	Yes	S1	Null	Null	列管主機規則
R2	Yes	S2	Null	Null	提供軟體更新的伺服器規則
R3	Yes	S3	null	null	申請使用特定 port
R4	Yes	S4	null	null	使用不安全組合服務之主機

表二、子法則表格

Sub Rule	Source ip	Source port	Dest ip	max	min	Unsafe port	Src_ip_gid	dst_ip_gid
s1	Null	Null	Null	100	2	Null	1	Null
s2	Null	Null	Null	1000	1	Null	null	5
s3	Null	80	Null	null	null	null	null	Null
s4	Null	Null	Null	Null	Null	100	null	null