以開放原始碼為基礎的蜜罐系統設計與實現

葉昭熙　　楊中皇
國立高雄師範大學資訊教育研究所
Stanley5402@gmail.com
chyang@computer.org

摘要

蜜罐(Honeypot)是一種用來蒐集攻擊者資訊的資訊系統資源(Information system resource)，部署在組織防火牆之外可當作是早期警戒系統，部署在防火牆之內則可作為縱深防禦(Defense in depth)的一環，可用來偵測繞過防火牆、入侵偵測系統的入侵者或者是來自於組織內部的威脅。Honeyd 是一個開放原始碼的蜜罐系統，但由於其使用命令列式的介面而且設定繁瑣，往往會讓初學者充滿挫折，本研究即在使用開放原始碼的開發工具 Eclipse 以及 Java 語言來替 Honeyd 建構一個友善的使用者圖形介面，同時將建構完成的系統安裝於可攜性高的 Live USB 中，以方便使用者的攜帶與部署。藉由圖形化的管理介面，使用者可使用選單快速的建立 Honeyd 的範本檔或是驅動網路掃描工具(Nmap)對目標電腦進行掃描，並將得到的結果加以分析後轉化為 Honeyd 的範本檔，此外系統並提供日誌檢視介面與即時的警訊傳送功能。

關鍵字：蜜罐(Honeypot)、蜜網(Honeynet)、縱深防禦(Defense in depth)、Honeyd。

1. 緒論

隨著網際網路的盛行以及頻寬的不斷增加，使用者電腦面臨著日益嚴重的駭客攻擊與惡意軟體的威脅，由於傳統的資料安全技術[1]，如防火牆、入侵偵測系統(IDS)、入侵防禦系統(IPS)及防毒軟體等，大多數是屬於被動式的安全防禦機制，必須透過病毒定義碼或特徵定義檔來分析網路攻擊行為是否存在，雖然此方法擁有很高的精準度[2]，但是當面對零時差攻擊（Zero-day Exploits）和不斷變種的病毒時，就會因為比對不到資料庫中的相關特徵定義而變得無計可施，使用者只能依失誤來動態調整防禦策略，而蜜罐系統的應用則可以改變這種局面；蜜罐是一種用來蒐集攻擊者資訊的資訊系統資源，部署在組織防火牆之外可當作是早期警戒系統，部署在防火牆之內則可作為縱深防禦的一環，可用來偵測繞過防火牆、入侵偵測系統的入侵者，或者是其他來自於組織內部的威脅。

Honeyd是一種開放原始碼的蜜罐系統，但由於其使用命令列式的介面，而且設定繁瑣，往往會讓初學者感到挫折，本研究旨在藉開放原始碼的開發工具Eclipse以及Java語言來替Honeyd建構一個友善的圖形使用介面，並將建構完成的系統安裝於一可攜性高的Live USB中，讓使用者能夠輕易攜帶並使用。利用本研究所開發出來的圖形化管理介面，使用者可使用選單快速的建立Honeyd的範本檔或是驅動網路掃描工具(Nmap)對目標電腦進行掃描，並將得到的結果加以分析後轉化為Honeyd的範本檔，此外系統並提供日誌檢視介面與即時的警訊傳送功能。

2. 文獻探討

2.1 蜜罐的歷史起源

在 1980 年代[16]，於美國加州 LBL 國家實驗室(Lawrence Berkeley Laboratory)工作的 Clifford Stoll 扮演著抓住德國駭客 Markus Hess 的角色。整個故事的起源是某一天 Clifford 的老闆要求他解決一個記帳系統上的 75 美分誤差，他追蹤到這個錯誤是來自於主機內一個未被授權的用戶，此用戶利用 Gnu Emacs 的 sendmail 功能弱點來獲得系統的 root 使用權；與其將入侵者隔絕開來，Clifford 決定讓入侵者繼續存取系統，並在存取的同時將其所有的活動列印出來，於是在接下來的 10 個月，Clifford 花了很多時間和努力去追蹤駭客的來源，這幾乎是歷史上第一個被文件所記載的駭客案件。為了誘使駭客在線上停留足夠長的時間以利回溯追蹤，Clifford 設立了一個精心製作的陷阱，他根據想像中的戰略防禦契約(Strategic Defense Contract)在 LBL 內創立了一個新的虛擬部門，此外，他也在電腦系統內的"SDInet"目錄中放滿了使用官僚術語及會令人印象深刻的大檔案，經過長達一年的追蹤之後，發現入侵者是透過通訊衛星或海底電纜來自於德國的 DATEX-P 網路，隨後德國警方就在漢諾威的駭客家中找到了入侵者 Markus 入侵 LBL 及美國軍方電腦的證據。

2.2 蜜罐的簡介

網路管理人員通常會使用防火牆和入侵偵測系統來保護他們的網路[11]，防火牆可根據要求的服務、使用者、封包來源及目的地來控制往來的交通；入侵偵測系統則可以安置在區域網路和網際網路之間，或區域網路上的其它重要地點偵測可疑的封包，但是就像人們有時出門也會忘記將窗戶鎖上一樣[9]，有時可能也會忘記更新防火牆的規則，而採用異常偵測(Anomaly Detection)技術的入侵偵測系統，長久以來也有讓人詬病的高誤報(False Positive)率問題，若使用蜜罐則可以彌補防火牆和入侵偵測系統先天上的缺點，甚至把它當作是引導電腦安全研究及教育的平台。研究蜜罐可以幫我們從技術及人種誌(ethnological)上的兩個不同觀點來增進對於駭客團體的認識[15]，就技術上而言，它提供一種新的方法幫助發現 Rootkit、特洛伊木馬程式(Trojan horse)及潛在的零時差攻擊；就人種誌而言，它可以讓我們更深入了解隱藏在駭客團體間的連結或其中感興趣的部份。

蜜罐是一種獨特的安全性資源，它可以用來模擬作業系統及其弱點，其主要目的是用來欺騙入侵者或當作是被攻擊的警報器，蜜罐的定義來自於一個擁有超過5000位安全技術專家的網路公共論壇 Honeypot maillist[17]：蜜罐是一種資訊系統資源，它的最主要價值在於被非授權者或非法者所使用(*A Honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource*)。這裡的資訊系統資源所指的是諸如工作站、檔案伺服器、郵件伺服器、印表機或甚至是一整個網路，而且此系統資源必須沒有任何的生產價值可言，因此在理論上任何企圖跟它建立的連線都有可能是探測、攻擊或者是入侵活動。由於蜜罐可以提供其它工具所無法獲得的獨特攻擊資訊[5]，所以其所搜集到的資訊都具有很高的價值，假如說保護生產線網路就好像是保護城堡一般，那麼蜜罐就好像是一條深入到敵後的間諜網路。

蜜網(Honeynet)是由數個蜜罐所組成[10]，它是一個用來搜集駭客相關資料的網路陷阱，搜集諸如：他們是誰?使用什麼軟體?利用什麼樣的弱點?蜜網使用常見的作業系統如Windows XP或Linux的預設安裝[8]，根據The Honeynet Project的研究：一個使用預設安裝的Red Hat 6.2伺服器在連上網際網路不到72小時內就會被攻破，此外蜜網也可以說是一種架構[18]，它就好像是一個魚缸，你可以先在魚缸中擺放任何你喜歡的東西如礁石、枯木、水草，然後再從透明的玻璃魚缸外觀察其中魚和水草的互動，而在蜜網中除了蜜罐之外，你還需要一些跟網路軟硬體相關的裝置如防火牆、路由器、交換器以及日誌記錄工具與封包分析器等，第一代蜜網的架構大致如圖1所示。

在蜜網中最重要的元件就是數據控制(Data Control)和數據捕捉(Data Capture)，數據控制定義了怎樣將攻擊者的活動限制在蜜網中而不被攻擊者察覺，數據捕捉則是指在攻擊者不知情的情況下，如何掌握其所有的攻擊活動，而兩者之中又以數據控制最為重要。

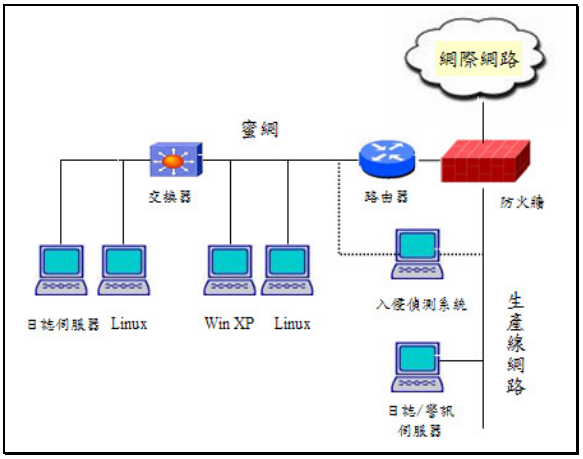

圖 1 第一代蜜網的組成元件

2.3 蜜罐的分類

Hassan Artail和其他學者共同提出了如表1所示的蜜罐分類方式[4]：

表 1 蜜罐的分類方式

	生產型	研究型
低互動	虛擬	
高互動	實體	

一、低互動的系統

低互動的蜜罐系統使用軟體去模擬作業系統的特性與網路服務來吸引入侵者，由於低互動蜜罐只能提供有限的互動，在理想上，入侵者的活動會被局限在此模擬環境中而不會危及宿主系統(Host System)。低互動的蜜罐系統部署在組織防火牆之內，可以當成一個很容易安裝及維護的入侵偵測感知器(Instruction Detection Sensor)[13]，用來偵測來自於其它內部機器的可疑活動，或當作是外部安全機制是否正常運作的指標；部署在組織防火牆之外，則可用來減緩網際網路上蠕蟲的掃描速度(又稱為 Tarpit)或是當作早期預警系統，用來搜集統計性的資料如：攻擊頻率、數量、型態以及來源地，讓組織在再次遭受更嚴重的攻擊之前，能先採取防禦措施。

二、高互動的系統

高互動的蜜罐系統使用真正的作業系統與應用程式，組織可以藉由提供攻擊者真實的系統來與其進行互動，以便學習更多關於攻擊者行為的知識，但部署高互動的蜜罐較為複雜，建置這類蜜罐需要額外的技術，以避免攻擊者利用它來攻擊其它的系統。

高互動的系統在應用上可用來發現新的Rootkit、木馬後門程式、潛在的零時差攻擊[12]，或藉由主動加入疆屍網路(Botnets)觀察蠕蟲是如何透過IRC (Internet Relay Chat)來發起分散式阻斷攻擊(DDoS)，並研究其解決之道。此外由於入侵者會在高互動的系統上留下更多的足跡，如電子郵件、IRC 交談紀錄、Rootkit、使用的攻擊工具等，因此它還可以用來協助電腦鑑識中的證據搜集工作。

三、研究型的系統

像 The Honeyney Project 所開發出來的蜜網就是一個研究型的系統[20]，The Honeyney Project 是一個非營利、自發性的安全研究組織，其成員包含了來自 Sun Microsystems、Cisco Systems、Guardent、Foundstone 等公司以及澳洲、加拿大、荷蘭、以色列等不同國家的志願者。當蜜網捕捉到訊息之後，研究人員就會利用它來分析入侵者所使用的工具、策略及動機，並用以預測下一次攻擊的時間及型態。部署研究型的系統並不會幫組織增加任何安全性，但其所搜集到的攻擊者資訊可以用來作為早期預警或預測，並間接幫助組織對於即將面臨的威脅提供更好的防護。

四、生產型的系統

所謂生產型的系統是指用來減少組織內部電腦被攻擊機率的蜜罐，藉由捕捉繞過防火牆、入侵偵測系統的入侵者，或是來自於組織內部的威脅(如契約商、心存報復的員工、生意上的夥伴與系統、網路、資料庫管理員)[19]，用以輔助網路型與主機型的入侵防護機制並成為組織內縱深防禦的一環。生產型的蜜罐系統在部署時應模擬現有環境的作業系統及網路服務，若所有的系統都有安裝補丁程式(Patch)，那麼蜜罐也應該要安裝補丁程式。生產型的系統可以使用高互動或低互動的蜜罐，若要使用高互動的蜜罐則要做好資料控制，以避免入侵者將之當成跳板並用來攻擊其它電腦系統，其缺點是只能捕捉有限的資訊，因為當駭客手法被學起來之後，漏洞就應該補起來。

五、 真實的系統

真實的系統是一種很好的高互動系統，不需要再做任何的模擬，而一台未安裝補丁程式的電腦作業系統可說是一個最簡單型態的蜜罐，例如你安裝了 Windows XP 卻故意不安裝 SP1 及 SP2。其缺點是每個系統都需要額外的硬體與軟體版權成本，在部署及管理上很麻煩，而資料控制也很困難並且需要很多額外的工作。

六、虛擬的系統

虛擬的系統可以模擬真實的作業系統及其上所運行的服務，按照其使用的技術大概可分為虛擬機器與使用仿真的服務：

1. 虛擬機器(Virtual Machine)：和真實的系統很像，允許很快的重新部署，但虛擬機器有可能被入侵者識別出來並被當作跳板[7]，如 VMware ，Microsoft Virtual PC 及 Virtual Box 都是屬於此類的軟體。虛擬機器的缺點和真實的系統很像，其每個系統都需要額外的記憶體及軟體版權成本，同時在管理上也很麻煩。

2. 仿真的服務：使用仿真的服務直接運行在真實的作業系統上，是屬於一種低互動、低成本、易安裝的蜜罐(如 Honeyd)，其缺點是無法用來長期捕捉駭客的活動，且功能也很有限，其困難點在於不可能做到百分之百完全模擬作業系統的特性及其所有的服務。

2.4 Honeyd

Honeyd 是由 Dr. Niels Provos 於 2002 年 4 月所開發出來的一個開放原始碼與低互動的蜜罐系統[6]，它原先只能執行在 Unix-like 的環境下，後來 Michael Davis 將其移植 Windows 平台上。Honeyd 擅長根據 Nmap 及 Xprobe 的特徵檔(Fingerprint)來模擬作業系統的 IP stacks，並利用外部程式來模擬其上所運行的服務，如 FTP、TELNET、SMTP、HTTP 等，它可以在單一台電腦上建立數個虛擬的蜜罐或甚至是一整串的網路，Honeyd 允許在單一台電腦上最多可模擬 65536 個不同的 IP 位址，其詳細的運作方式如圖 2 所示[14]：

一. 當 Honeyd Daemon 接收到封包時會將之交給封包分配器(Packet Dispatcher)，封包分配器會檢查封包的長度及 checksum。由於分配器只認識 ICMP、TCP 及 UDP 三種通訊協定，其它的協定會被捨棄。

二. 封包分配器會根據目的地 IP 查詢配置資料庫(Configuration Database)中相對應的模擬範本，若是找不到相對應的 IP 則會使用預設的範本(Default Template)。

三. 對於 TCP 與 UDP 協定，Honeyd Daemon 可以透過外部程式如 Perl、Shell Script、Python 建立模擬服務，外部程式使用標準輸入接受資料並透過標準輸出將結果回傳給 Honeyd。

四. 最後個性化引擎(Personality Engine)會修改每一個往外送的封包，使其模擬原先所設定的行為。

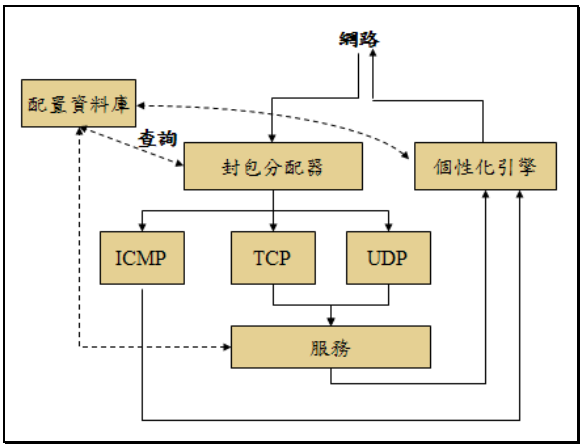

圖 2　Honeyd 的運作方式

3.　系統實作

系統的實作是採用 Eclipse 及其輔助畫面設計的外掛程式-- Jigloo，Eclipse 是開放原始碼的專案，可以到 http://www.eclipse.org/downloads 去尋找最新版本下載。Eclipse 本身是用 Java 語言撰寫，但下載的壓縮檔中並不包含 Java 執行環境，需要使用者另行安裝 JRE 或 JDK，Eclipse 的安裝步驟非常簡單，只需要將下載的壓縮檔直接解壓縮即可。

3.1　Live USB 製作

本研究製作 Live USB 所選用的作業系統是 2007 年 4 月 19 日 ubuntu 所釋出代號為 Feisty Fawn 的最新版本--7.04 版。ubuntu 的硬體支援性佳、容易安裝，而且使用其提供的套件管理工具可透過視窗操作介面直接線上安裝套件，更重要的是 ubuntu 提供了更嚴密的安全控管服務。將 ubuntu 7.04 安裝在 4Gb USB 隨身碟上的步驟如下[3]：

一、 使用 ubuntu 7.04 Desktop 版的 Live-CD 開機。

二、 將 USB 隨身碟插入電腦並檢查其代號，如：(hd1，0)。

三、 按 ubuntu 桌面上的 Install 圖示，開始安裝步驟。

四、 將隨身碟分割為 ext3 及 swap 兩種型態的 partition，並按照畫面指示繼續下一步驟。

五、 看到"Ready to install"畫面出現時，按"Advanced"圖示，填入正確的隨身碟代號如：(hd1，0)，並將 grub 開機程式安裝在 USB 隨身碟上。(注意：若此步驟出錯會導致電腦不能開機，若 grub 開機程式不小心安裝到 windows 系統上，可用 fixmbr.exe 修復)

六、 開始安裝(約 60 分鐘)。

七、 安裝完成後，編輯"\boot\grub\menu.lst"，將其中的(hd1，0)或(hd2，0)改為(hd0，0)。

八、 為了避免顯示卡不相容的問題並讓 USB 隨身碟能在任何電腦都能開機，將"\etc\x11\xorg.conf"這個檔案更改檔名。

九、 將電腦的 Bios 設定改為從 USB 隨身碟開機，儲存並離開後即完成。

3.2　系統架構

本系統的架構如圖 3 所示，主要模組分為 Honeyd 管理介面、範本檔產生模組、IP 位址連結模組與警訊傳送模組等，分別說明如下：

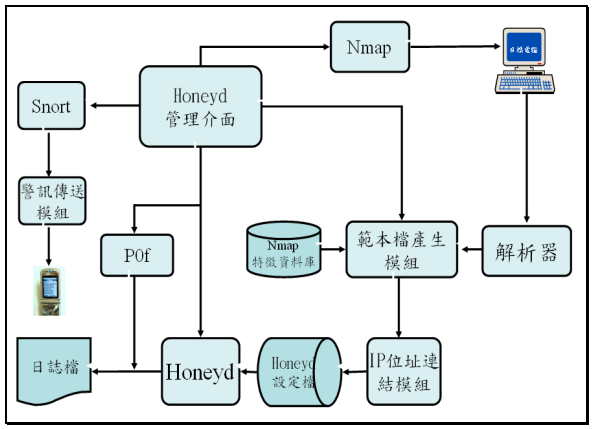

圖 3　系統架構圖

- Honeyd 管理介面：可以讓使用者設定各種路徑、參數(中英文操作介面)或直接管理其它程式的啟動(如 Snort、P0f)。此外，使用者尚可透過圖形介面驅動 Nmap 對目標電腦進

行掃描，掃描後解析器(Parser)會將回傳結果中所猜測到的可能作業系統型態及開放的 TCP 埠萃取出來，透過和 Nmap 之特徵資料庫(fingerprint)的比對分析，讓使用者選擇最類似的形態，再自動建立可供 Honeyd 使用的範本檔。

- 範本檔產生模組：此模組可讓使用者自行從 Nmap 的特徵資料庫中挑選可以進行模擬的作業系統型態，再針對 Honeyd 可模擬的三種通訊協定：TCP、UDP、ICMP 分別設定其狀態與要模擬的服務，其中 TCP 及 UDP 的允許狀態有三種：OPEN、BLOCK、RESET；而 ICMP 的允許狀態只有 OPEN 與 BLOCK 二種。針對所要模擬的服務，使用者可以設定執行服務的埠及相對應的外部程式；針對系統特性，還可以設定 Unix Like 作業系統上所特有的 UID 及 GID 及系統的待機時間(Uptime)；針對網路傳輸還可以設定其封包遺失率，以模擬真實世界的網路傳輸狀態。
- IP 位址連結模組：用來連結 Honeyd 範本檔與 IP 位址，在一個設定檔(Config File)內可以包含多個已經和 IP 位址連結的範本檔。
- 警訊傳送模組：每當系統啟動時，就會同時啟動一個程序來監視 Snort 的 Alert 檔案；當發現檔案有異動時，就會自動經由警訊傳送模組傳送一組訊息至系統管理員的手機中，此外為了抑制過多的訊息傳送，必須等系統管理員確認過事件後並按下重置鍵才能再次傳送。

3.3 系統操作

一、 使用圖形介面選單產生範本檔：Honeyd 管理介面啟動後，可選擇"範本建立"進入如圖 4 的範本檔產生介面，使用者透過選單選擇要模擬的行為及服務後，按"建立"就會在右下方視窗看到輸出結果，確認無誤後按"儲存"會看到如圖 5 的輸出結果產生的畫面。

圖 4 範本檔產生介面

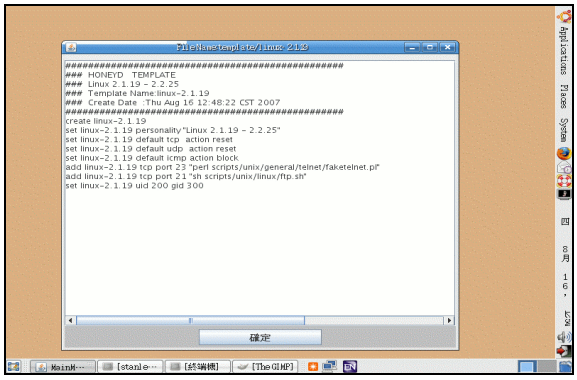

圖 5 輸出結果產生的畫面

二、 藉由 Nmap 對目標電腦掃瞄產生範本檔：從管理介面選擇"Nmap"進入如圖 6 的 Nmap 管理介面，在完成對目標電腦的掃描後，選擇"分析"會進入如圖 7 的 Nmap 日誌分析畫面，選單中會自動列出經過關鍵字比對後的相似選項供使用者挑選，輸入範本名稱後按"建立"與"儲存"即可完成產生與存檔的工作。

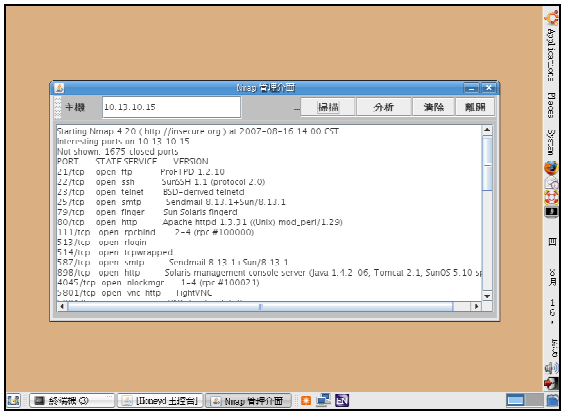

圖 6 Nmap 管理介面

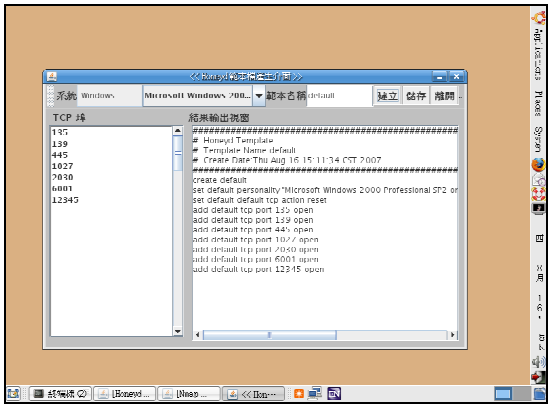

圖 7　Nmap 日誌分析畫面

三、 產生 Honeyd 的設定檔：從管理介面選擇”IP 位址連結”進入如圖 8 的 Honeyd 設定檔產生介面，輸入要和範本檔作連結的 IP 位址後按”加入”，則在上側的視窗中可以看到輸出結果，若要模擬的作業系統為路由器則要加入預設閘道的資料，確認無誤後按”儲存”即可存檔完成。

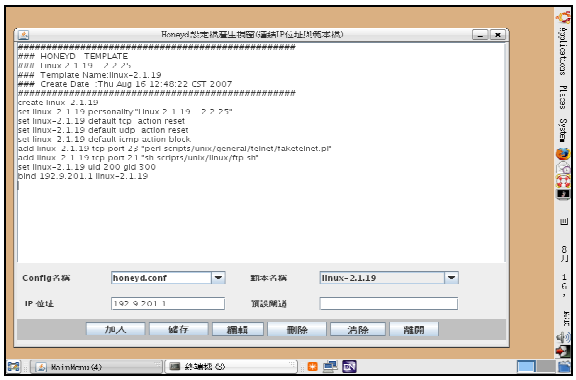

圖 8　Honeyd 設定檔產生介面

4. 系統部署

為了驗證系統的有效性及搜集攻擊的型態與來源，因此將之部署於學校的網路環境中。

4.1 環境建立

本實驗使用 Honeyd 模擬了兩台虛擬蜜罐系統，一台為 Linux 2.x 系統，另一台為 Solaris 2.x 系統。Linux 系統採用圖形介面選單產生，並且執行利用外部程式所模擬的 Telnet 與 Ftp 服務；Solaris 2.x 系統使用掃描真實系統所建立的範本來產生。

4.2 資料分析

經過72小時的資料搜集後，Linux及Solaris系統各自遭受到1097及995次的TCP連線攻擊(平均每小時13~15次)，針對所遭受攻擊的埠，整理出如表2及表3的前10名排行榜。

Windows 系統的 TCP 埠 139 (NetBIOS)及 445 (CIFS)是被設計用來存取具敏感性或未加密的資料，因此長期以來就是入侵者或蠕蟲進行弱點攻擊前建立連線的目標，但由於這兩種協定本來就是被設計用於內部網路而非網際網路，因此防火牆或路由器都應該將它過濾掉[21]。

表 2 Linux 系統

排行	埠	次數
1	445	781
2	139	75
3	2967	59
4	1433	24
5	5900	24
6	8080	22
7	2968	18
8	22	13
9	54363	9
10	80	7

表 3 Solaris 系統

排行	埠	次數
1	445	676
2	139	81
3	2967	41
4	8080	32
5	22	32
6	5900	31
7	1433	22
8	80	10
9	25	9
10	4899	7

再來，若將攻擊來源根據校內與校外區分，其遭受 TCP 連線攻擊的次數分別為 1722 及 370 次，表 4 及表 5 分列出其遭受攻擊頻率最高的前 10 個埠，根據表格內的資料來看，從內部及外部所發起的攻擊型態、數量、頻率都有很明顯的不同。最後，針對外部攻擊來源國家整理出表 6。

表 4 來自於校內的攻擊

排行	埠	次數
1	445	1457
2	139	156
3	2967	81
4	2968	18
5	33099	4
6	33098	4
7	22	2

表 5 來自於校外的攻擊

排行	埠	次數
1	5900	55
2	8080	54
3	1433	46
4	22	43
5	2967	19
6	80	17
7	4899	14
8	25	10
9	54363	9
10	5168	9

表 6 外部攻擊來源分析

編號	國家	次數
1	中國大陸	150
2	台灣	77
3	美國	66
4	巴西	16
5	南韓	11
6	日本	11
7	義大利	10
8	捷克	8
9	孟加拉	5
10	印度	3

編號	國家	次數
11	德國	3
12	哥倫比亞	2
13	俄羅斯	2
14	愛沙尼亞	2
15	法國	1
16	馬來西亞	1
17	斯洛伐克	1
18	薩爾瓦多	1

5. 結論

本研究所開發的圖形介面系統，結合了網路掃描工具 Nmap，利用其強大的作業系統辨識能力及對 TCP/IP 的 TCP 埠掃描功能，快速擷取另一台電腦的特性，並將擷取到的資料轉換成部署蜜罐所需的範本檔。此外，使用者也可根據自己的需求使用內建的範本與服務，或者使用選單自行建立，最後將建構完成的系統安裝於可攜性高的 Live USB 隨身碟中，使用者使用本系統只需要具備一些基礎的相關知識，即可快速的進行蜜罐的部署。

由於 Honeyd 可以模擬很複雜的網路結構，因此在未來可進行的研究方面，可考慮用更具使用者親和性的視覺化及拖曳(Drag and Drop)方式來描繪出複雜的網路結構，或者是針對所使用掃描工具的效率或架構再加以加強，以更增加使用者運用蜜罐進行動態部署的能力。

參考文獻

[1] 林弘憲，“結合網路誘捕技術在網路蠕蟲攻擊分析之應用研究”，樹德科技大學資訊管理研究所碩士論文，2005。

[2] 劉醇瑞，“基於行為探測法之惡意流量隔離機制”，世新大學資訊管理學研究所碩士論文，2005。

[3] Billypan 的部落格，2007，http://www. wretch. cc/ blog/billypan101.

[4] Hassan Artail , Haidar Safab , Malek Sraj , Iyad Kuwatly and Zaid Al-Masri, “A hybrid Honeypot framework for improving intrusion detection systems in protecting organizational networks,” *Computers & Security,* Volume 25, Issue 4, June 2006, pp. 274-288.

[5] Anton Chuvakin, “Honeynets: High Value

Security Data:Analysis of real attacks launched at a honeypot," *Network Security* , Volume 2003, Issue 8, 2003, pp. 11-15.

[6] Honeyd, http://www.Honeyd.org/

[7] Thorsten Holz and Frederic Raynal, "Detecting Honeypots and other suspicious environments," in *Proceedings of the IEEE Workshop on Information Assurance and Security United States Military Academy,* West Point, NY, 2005.

[8] Gerald Johns, "Watching Hackers in the Honeynet, " *Network Security*, Volume 2001, Issue 8, August 2001, pp.6-6(1).

[9] R. A. Kemmerer and G Vigna, "Intrusion Detection: A Brief History and Overview," *IEEE Computer Society*, Volume 35, Issue 4, 2002, pp. 27-30.

[10] Andrea Kirkby, "Honeynet Phase Two: Knowing Your Enemy More," Computer Fraud & Security, Volume 2001, Issue 12, 2001, pp. 8-9.

[11] John G. Levine, Julian B. Grizzard and Henry L. Owen, "Using honeynets to protect large enterprise networks," *IEEE Security & Privacy*, Volume 2, Issue 6, 2004, pp.73 -75.

[12] Bill Maccarty, "Botnets: Big and Bigger," *IEEE Security & Privacy*, Volume 1, Issue 4, 2003, pp. 87-90.

[13] Robert McGrew, "Experiences with Honeypot Systems: Development, Deployment, and Analysis," in *Proceedings of the 39th Annual International Conference on System Sciences*, Hawaii, 2006.

[14] Niels Provos , "Honeyd: A Virtual Honeypot Daemon," in *13th USENIX Security Symposium*, San Diego, CA, 2004.

[15] Frederic Raynal, Yann Berthier, Philippe Biondi and Danielle Kaminsky, "Honeypot Forensics Part I: Analyzing the Network," *IEEE Security and Privacy*, Volume 2, Issue 4, 2004, pp.72-78.

[16] Clifford Stoll, "Stalking the wily hacker, "*Communications of the ACM*, Volume 31, Issue 5, 1988, pp. 484-497.

[17] Lance Spitzner, "Honeytokens: The Other Honeypot," 2003, http://www.securityfocus. com/infocus/1713.

[18] Lance Spitzner, "The Honeynet Project: Trapping the Hackers," *IEEE Security & Privacy*, Volume 1, Issue 2, 2003, pp. 15-23.

[19] Lance Spitzner, "Honeypots: Catching the Insider Threat," in *Proceedings of the 19th Annual Computer Security Applications Conference,* Las Vegas, NV, USA, 2003.

[20] The Honeynet Project, 2007, http://www. honeynet.org.

[21] Thomas Kristensen, "The Big Picture on Big Flaws: RPC DCOM Vulnerability — What went wrong?," *Network Security,* Volume 2003, Issue 9, September, 2003, pp. 19-20.