國立臺東大學資訊管理學系
環境經濟資訊管理碩士專班
碩士論文

指導教授：謝昆霖　博士

建置 P2P Botnet 網路流量偵測系統

研 究 生： 李明鴻　撰

中 華 民 國 九 十 九 年 七 月

國立臺東大學資訊管理學系
環境經濟資訊管理碩士專班
碩士論文

建置 P2P Botnet 網路流量偵測系統

研 究 生： 李明鴻　撰
指導教授： 謝昆霖　博士
中 華 民 國 九 十 九 年 七 月

國立臺東大學
學位論文考試委員審定書

系所別：

本班 李明鴻 君

所提之論文 建置 P2P BOTNAT 網路流量偵測系統

業經本委員會通過合於 碩士學位論文 條件

論文學位考試委員會：

陳彥仁

（學位考試委員會主席）

施能木

謝品勇

（指導教授）

論文學位考試日期：99 年 07 月 03 日

國立臺東大學

附註：1. 本表一式二份經學位考試委員會簽後，正本送交系所辦公室及註冊組或進修部存查。

2. 本表為日夜學制通用，請依個人學制分送教務處或進修部辦理。

博碩士論文授權書

本授權書所授權之論文為本人在 國立臺東大學 資訊管理學 系(所)
環境經濟資訊管理碩士專組（班）九十八學年度第 二 學期取得 碩 士學位之論文。

論文名稱： 建置 P2P BOTNET 網路流量偵測系統

本人具有著作財產權之論文全文資料，授權予下列單位：

同意	不同意	單　位
■	□	國家圖書館
■	□	本人畢業學校圖書館
□	■	與本人畢業學校圖書館簽訂合作協議之資料庫業者

得不限地域、時間與次數以微縮、光碟或其他各種數位化方式重製後散布發行或上載網站，藉由網路傳輸，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

□同意 ■不同意 本人畢業學校圖書館基於學術傳播之目的，在上述範圍內得再授權第三人進行資料重製。

本論文為本人向經濟部智慧財產局申請專利(未申請者本條款請不予理會)的附件之一，申請文號為：_____，請將全文資料延後半年再公開。

公開時程

立即公開	一年後公開	二年後公開	三年後公開
			✓

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。上述同意與不同意之欄位若未勾選，本人同意視同授權。

指導教授姓名： 謝孫源 (親筆簽名)

研究生簽名： 李明鴻 (親筆正楷)

學　　號： 4397014 (務必填寫)

日　　期：中華民國 99 年 7 月 29 日

1.本授權書 (得自 http: //www.lib.nttu.edu.tw/theses/ 下載) 請以黑筆撰寫並影印裝訂於書名頁之次頁。

2.依據 91 學年度第一學期一次教務會議決議:研究生畢業論文「至少需授權學校圖書館數位化，並至遲於三年後上載網路供各界使用及校內瀏覽。」

授權書版本:2008/05/29

謝誌

這本論文的完成，要感謝的人好多。

感謝謝昆霖老師的指導。老師靈活的思想給了我這個題目，在鑽研文獻的同時，不時也回想起真是好多條理需要釐清，每當有所困惑，請教謝老師之後總是可以有瞬間的豁然開朗。每當進度落後時，謝老師的臨門一腳提點，總是給了極大的動力往前繼續狂奔！謝老師不僅是在我學業上的老師及支撐，也曾經是在工作上的指導員，回想那段在學校的工作，總感覺慶幸老師給了我比別人更多的工作歷練，這些經驗不知不覺已經累積在工作職涯當中，老師真的是給了我好大的寶藏。此外感謝施能木老師及陳彥匡老師也藉著論文，提點我一個研究該有的嚴謹度，讓我知道做研究該有的務實態度。

感謝電算中心的舊長官們，郭俊賢先生、林美秀小姐、方中秋先生及劉世泓先生。感謝老大總是扮演人生導師，當人生有所困惑，總是可以從老大的言行中獲得一些啓示；感謝中秋哥，在業界的經驗及教導的能力，總是令我受益匪淺。感謝美秀姐及世泓哥，每當心情鬱悶時，總是能從你們那邊獲的一定程度的抒懷。

感謝碩專班兩年的同學，俊隆兄及玉珍姐，從兩位身上我看到了求學問的態度是如此的努力不懈。

最後感謝我親愛的家人，老爸、老媽、大姐及二姐，總是能包容著我。有您們真好！

最後，感謝台東大學，提供了教育資源、工作環境也培養了我一身歷練，讓我覺得我的人生不凡！

謝謝大家。

Lee MingHung @NTTU

建置 P2P BOTNET 網路流量偵測系統

作者：李明鴻

國立台東大學 資訊管理環資碩專班

摘要

殭屍網路（Botnet）是一種可以由遠端多階層式架構針對各系統發動攻擊的資安威脅，且由於其隱匿的特性，所發動攻擊也不容易完全防範。而對網路管理人員若是可以即早監控該類網路流量，應可使遭受網路攻擊的災害損失降至最小。然而事實上殭屍網路仍然具有某些特定的網路行爲，如經由特定埠號進行網路資料傳遞，且針對發動攻擊時，也是針對特定的服務。針對網頁伺服器發動分散式阻斷服務攻擊，散發垃圾信件。

故本研究旨在針對區域網路環境進行殭屍網路流量偵測，期望由該類病毒常用的傳輸資料特徵及網路服務的流量來進行分類，進而找出淺在的中毒電腦。經本研究的流量偵測系統評估後，系統可以找出已知中毒電腦並偵測潛在可疑電腦。

關鍵詞：殭屍網路、DDOS、流量偵測

Contruct P2P Botnet network traffic detecting system

Lee Ming-Hung

Abstract

Botnet is a remote and multi-hierarchical network system to attack the internet information security. Because of its hidden features, it is not easy to monitor its work and completely prevent its attacks. However, if internet managers can detect and monitor the network traffic as soon as possible, it would be possible for them to minimize disasters of Botnet attacks. In fact, Botnet does contain specific network behaviors, such as network communicating through specific ports. In addition, its attack often targets specific services such as interrupting web server functioning or distributing spam emails.

This study was to detect Botnet network traffic in the LAN network environment, in which the Botnet virus could be specified based on its network communicating characteristics and network service traffic. Therefore, the infected computers could be furthermore identified. After the Botnet network traffic system was evaluated, the research results demonstrated that the system could identify the infected computers and detect suspicious Botnet computers.

Keywords：Botnet、DDOS、Network traffic detect

目　次

圖目次

表目次

第一章 緒論

本章依次說明本研究的動機、研究目的、研究流程及範圍限制和名詞釋義等四小節。

第一節 研究動機

近年來 DDOS 攻擊可由新聞中報導已發展成爲流行的網路攻擊手法之一，而導致此種攻擊的成因已經有一定數量從傳統的病毒感染轉變成殭屍網路的攻擊模式。根據報導統計全球電腦網路近來出現 BotNet 網路攻擊犯罪也正悄悄襲擊台灣網路，甚至台北市所感染的殭屍電腦高達三十四萬台。而政府單位也於相關網站發布公告勿輕忽 BotNet 威力，以免網路攻擊事件持續擴大危害，如此才能確保資訊系統安全與防止重要資料外洩。現今在網路如此發達的世界中，個人資料保護的議題越來越受尊重，然而病毒自我演化卻也無時無刻的進行著，如同上述描述所提及的殭屍網路，根據資安業者的統計台灣受到殭屍病毒控制的數量已位居全球前十大。

而一般常見的殭屍網路可依其網路發展拓樸大約分爲三類：集中式、分散式及混雜模式。其中集中式最早從 IRC 通訊協定中來作發展，而成員組成可分爲：控制者、被控端及 IRC 伺服器，一般被控端中毒之後便會開啓 IRC 通訊，而當控制者需要發動攻擊時便透過 IRC Server 發布攻擊指令，進而達成駭客所希望達成的目的。而針對此種攻擊，目前最有效方式爲在攻擊發動前，截獲控制者經由 IRC Server 所下達之指令或將被控端切離 IRC Server 網路；然而分散式殭屍網路(P2P Botnet)也便由此而生，由於 P2P 模式中同儕網路中的電腦，都可以演化成 IRC server 的角色，而原本的封包擷取，也因爲 P2P 加密的傳輸，而更難偵測其特殊發動攻擊的字串。因此大幅的提升 Botnet 殭屍網路隱藏的給人們帶來資訊安全的威脅。

第二節 研究目的

目前 BotNet 的種類已經由傳統的集中式通訊(例如：IRC based、HTTP Based)轉變爲分散式的 P2P Based，此一轉變最主要是爲了避免傳統 BotNet 的 C&C Server 被偵測或者攔截後，無法將攻擊者的命令正確發送出去，甚者 P2P Based 若要更新攻擊程式亦可從同儕清單中快速更新。而針對傳統使用 IRC 或 HTTP 的 BotNet 以往的偵測模式大都是，建立 Honeypot 分析電腦中毒後的特性，進而分析攻擊特徵碼，並在網路流量中擷取分析封包以比對相關攻擊特徵碼，再加以攔截攻擊；但是 P2P BotNet 已經可以實現將攻擊特徵進行加密，並透過同儕清單中的電腦加速傳播攻擊程式及感染新電腦；雖然依舊可以側錄封包內容，但撰寫攻擊特徵也就更爲麻煩。想當然，未來攻擊病毒的種類勢必以 P2P 作爲其技術，而發動攻擊時也就成爲最有攻擊性的模式。

本研究的目的希望可以在區域網路的架構底下，嘗試利用網路流量分類紀錄及網路流量統計的判定方式找尋出已知或潛在的 P2P BotNet 網路並提早產生告警提供給網路管理者及使用者，以便做好漏洞修補及資訊安全的防護工作。

第三節 研究流程及範圍限制

本研究流程及範圍限制如下：主要研究流程如下圖

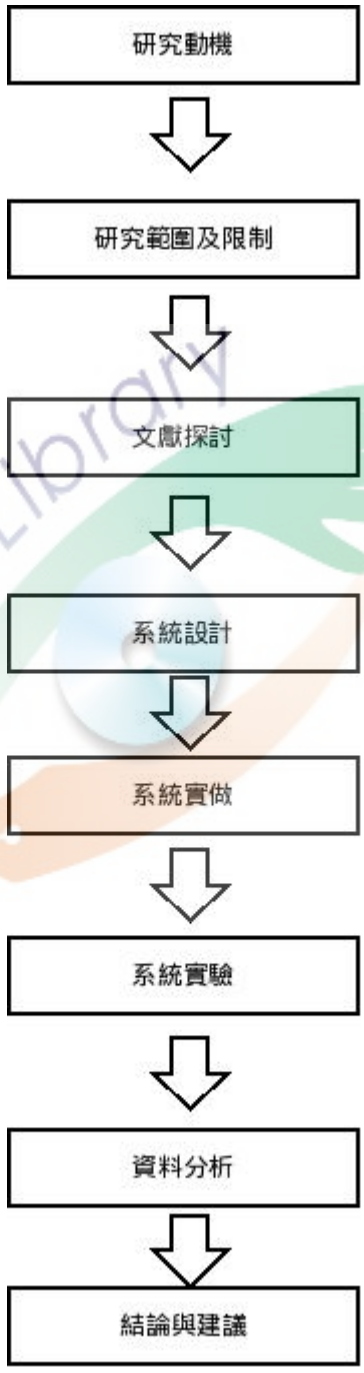


圖 1 研究流程圖

研究範圍則是限制以 virtual Box 虛擬機器架構電腦環境，分別模擬三台各安裝不統種類的殭屍病毒電腦，網路環境則是限制在以寬頻網路橋接器當作 NAT 的網路環境架構中。研究限制病毒樣本非爲最新樣本，雖然仍有殭屍病毒的流量特徵，不完全是目前正在傳播中的病毒。

第四節 名詞釋義

一、殭屍網路

殭屍網路（Botnet）是指一種受遠端控制的殭屍電腦所組成的網路。而此種殭屍電腦至少隱藏在數以萬計的電腦系統中默默執行殭屍病毒。

二、分散式阻斷服務攻擊

分散式阻斷服務攻擊 (distributed denial of service attacks，簡稱 DDOS 攻擊)，利用分散於不同地方的多台主機，發送大量僞造來源位置的封包，癱瘓受害者所在的電腦網路，造成該種網路系統無法提供正常的網路服務。

三、P2P 疆屍網路

在 P2P 屍網路中，殭屍病毒採用 P2P 的方式來進行資料的傳輸與加密，使得比一般殭屍病毒有更有的隱匿性，更加難以偵測。

第二章 文獻探討

本研究將針對殭屍網路的流量特徵來進行偵測；所以本章分別介紹 ddos 攻擊，Botnet 及 P2P Botnet 特性，和其他相關研究等四部份做一文獻探討。

第一節 DDOS 攻擊

此節將介紹 DDOS 攻擊的相關研究文獻，主要內容爲分散式阻斷攻擊特性、攻擊模式、防禦方式及其 Botnet 相關聯結程度。

一、 分散式阻斷攻擊(DDOS)特性及攻擊模式

DDoS 也算是 DoS 的一種。在 A taxonomy of DDoS attack and DDoS defense mechanisms 文章中(Mirkovic, 2004)，原本 DOS 攻擊爲 DoS（Denial of Service）阻斷服務之簡稱，攻擊者藉由異常的連線方式占用系統的服務資源，以達成干擾正常系統提供的服務，而他與一般網路攻擊之不同點爲此種攻擊需要取得系統最高權限。而 DOS 攻擊演化成 DDOS 在於它的攻擊模式並非一對一，而是以分散式多對一的方式同時對一個目標發動攻擊，而這些發動攻擊的點，通常是已遭受入侵而不自知的殭屍電腦。由於這種攻擊大部分以遠端遙控的方式，利用殭屍腦行兇，因此不僅難以防範，追查攻擊端來源更是不易。

而 DDOS 攻擊模式大約可分爲以下幾個階段，底下將介紹一下，並請參考圖 2：

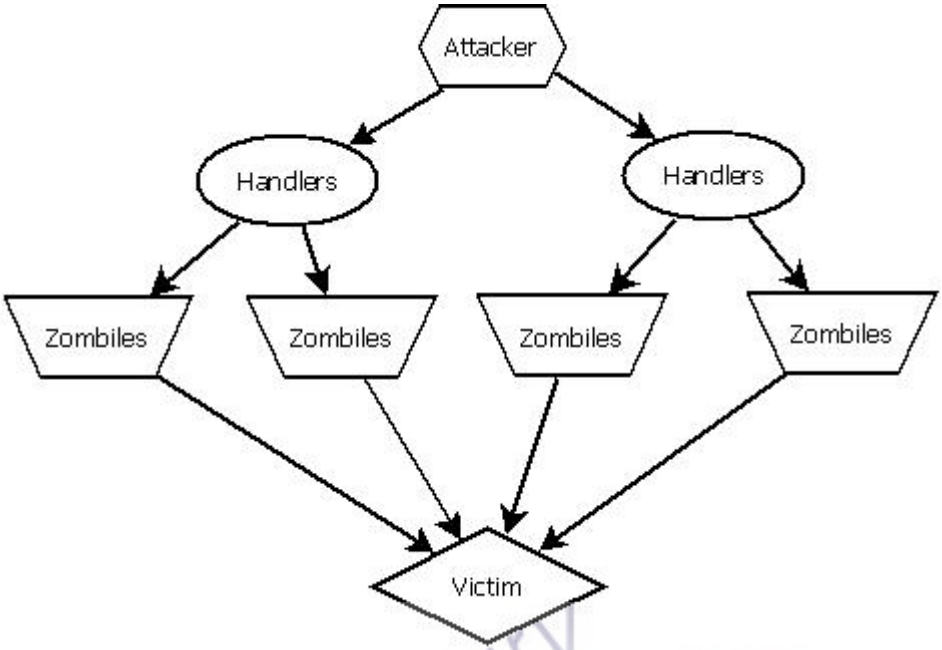


圖 2 DDOS 攻擊示意圖

(一)攻擊者(Attacker)先對疏於管理之系統主機(Handler)進行滲透，並將這些主機列爲往後 DDOS 攻擊之發動中繼站。

(二) 攻擊者將再度利用相同的攻擊手法藉由 Handler 主機中，大量探測更多的漏洞系統，並將其欲發動攻擊的程式指令碼植入受害端(Zombiles)並等待發動下一次攻擊。

(三) 接下來等待時機，將對目標(Victim)執行攻擊，以致該系統無法提供正常服務。

而從 DDOS 攻擊演變成 Botnet 攻擊時候，步驟會有如下演變過程，請參考圖 3：

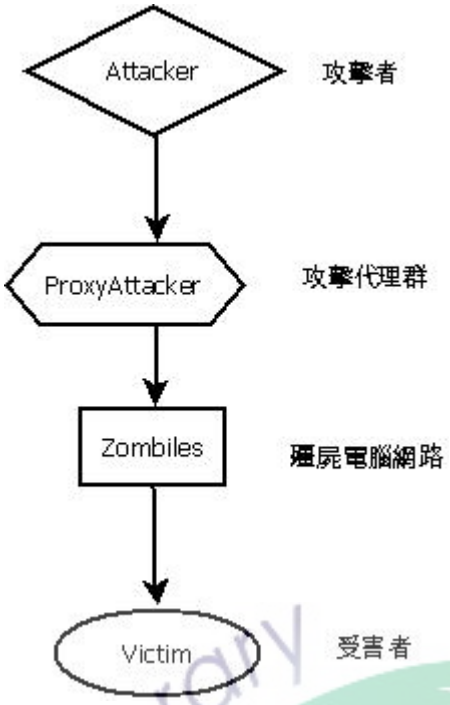


圖 3 Botnet 發動攻擊示意圖

攻擊者先利用工具掃描網路中有安全漏洞之伺服器，利用弱點攻擊取得系統權限之後，接著安裝攻擊代理及攻擊殭屍代理傀儡(Botnet)。攻擊代理扮演著轉發攻擊指令的角色並將指令轉發到攻擊代理傀儡機器上頭。而攻擊殭屍代理傀儡(Botnet)則是成爲攻擊的發動者。所以一旦發生攻擊行爲時，最原始的攻擊指令者的來源追查將有相當程度的困難性。攻擊者通過攻擊代理向攻擊僵屍傀儡機發出攻擊指令，使所有的僵屍傀儡機可以立刻或在預定的時刻同時向受害機器發動攻擊，自己則立刻離線以逃避追蹤。大量的攻擊回報一方面會造成目標伺服器系統及其網路資源被消耗殆盡，同時也會阻塞受害目標所在網路中的網路設備，從而達到使目標伺服器被阻斷服務之目的。

二、 DDOS 攻擊手法

攻擊者發動攻擊的對象可能視目標主機的網路頻寬、系統資源(CPU 使用、記憶體、磁碟空間、應用程式所提供的服務)等進行不同的攻擊。攻擊者的手法可以分爲許多種，在(Naoumov & Ross, 2006)的文章中，介紹幾個常見的攻擊手法：

(一) ICMP Flooding Attack

一般是 ICMP 封包被用來回報網路設備的狀態，在系統中常用的指令爲 ping。正常情形下，當系統發出 Echo request 時，對方系統會回應 Echo reply，來回應本身機器狀態。而 ICMP 洪水型攻擊(Labib & Vemuri, 2006)在此一情形之下，攻擊者可以假造大量來源不明 IP 位置對被攻擊者發送 ICMP 大封包，此時被攻擊者會回應等量的大封包向不同的 IP 作爲回應，但此類大量封包將把被攻擊者的網路頻寬耗盡，導致其他使用者無法使用網路資源。

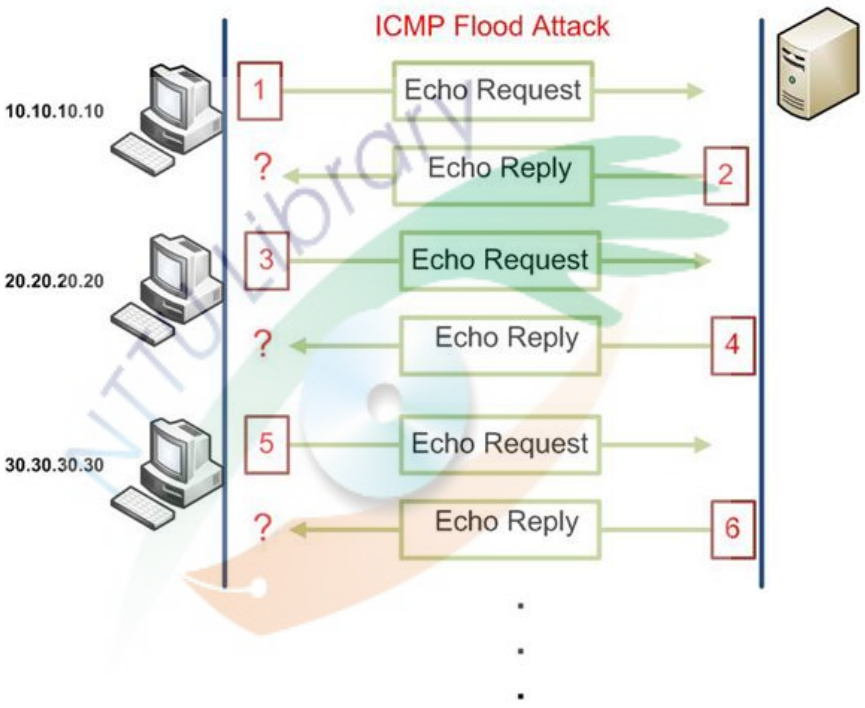


圖 4　ICMP Flooding attack

(二) TCP SYN Flooding Attack

TCP SYN Flooding Attack(Naoumov & Ross, 2006)，如下圖：一般 TCP 於建立連線時，會經過三方握手 (Three-Way Handshake) 的步驟，而攻擊者從 TCP 協定找出一項可達到攻擊的手法，即是攻擊者對伺服器連續發出假造 SYN 封包來要求連線，此時 Server 亦會對應發出 Sync+Ack 封包作爲回應，但由於這些攻擊者是經由假造的 IP 發送，所以伺服器的回應封包，會發生不知送往何處的等待情形，並把保留系統資源直到連線時

間 time out 爲止。所以當攻擊者發送大量的假造封包，系統資源來不及釋放提供新服務，就會造成被攻擊者服務資源被耗盡的情形，而到至系統發稱異常。

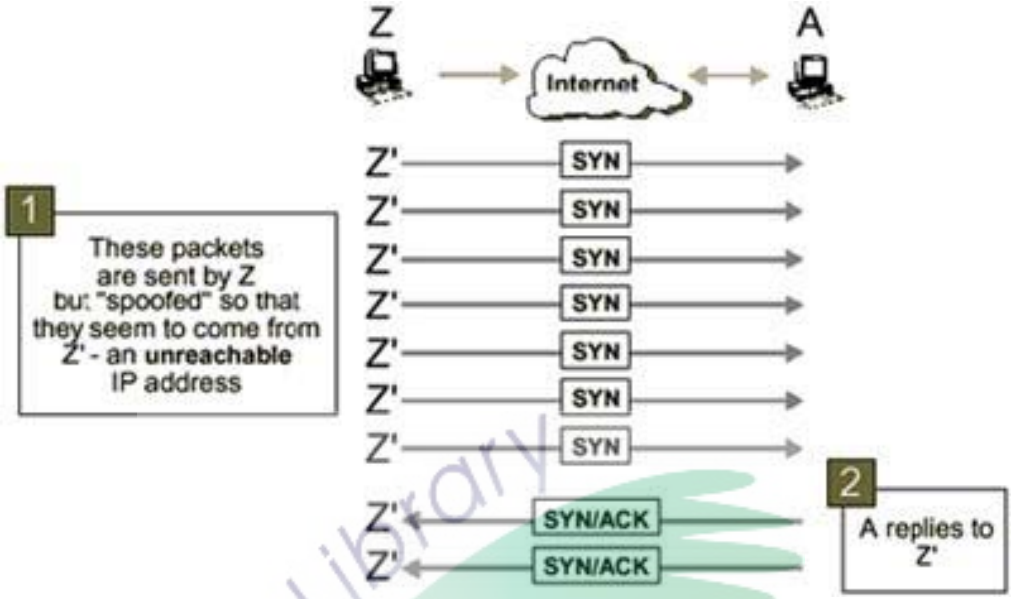


圖 5　TCP Syn Flooding

(三) UDP Flooding 攻擊

此類攻擊手法由於 UDP 封包本身爲不可靠連線的特性，所以攻擊者只要向被攻擊端主機發送假造的來源 IP，被攻擊主機便會對假造 IP 來一直發送回應封包，一則可以消耗網路頻寬，再則可以系統主機資源。

(四) Smurf 攻擊

Smurf 攻擊(Hussein & Zulkernine, 2006)則是類似 ICMP Flooding Attack 攻擊的一種，其最主手法爲像被攻擊者的網段發送大封包至 Broadcast address，致使其他相同網段的電腦也都會產生回應，進而塞爆被攻擊者的網段網路頻寬，造成該網域癱瘓。

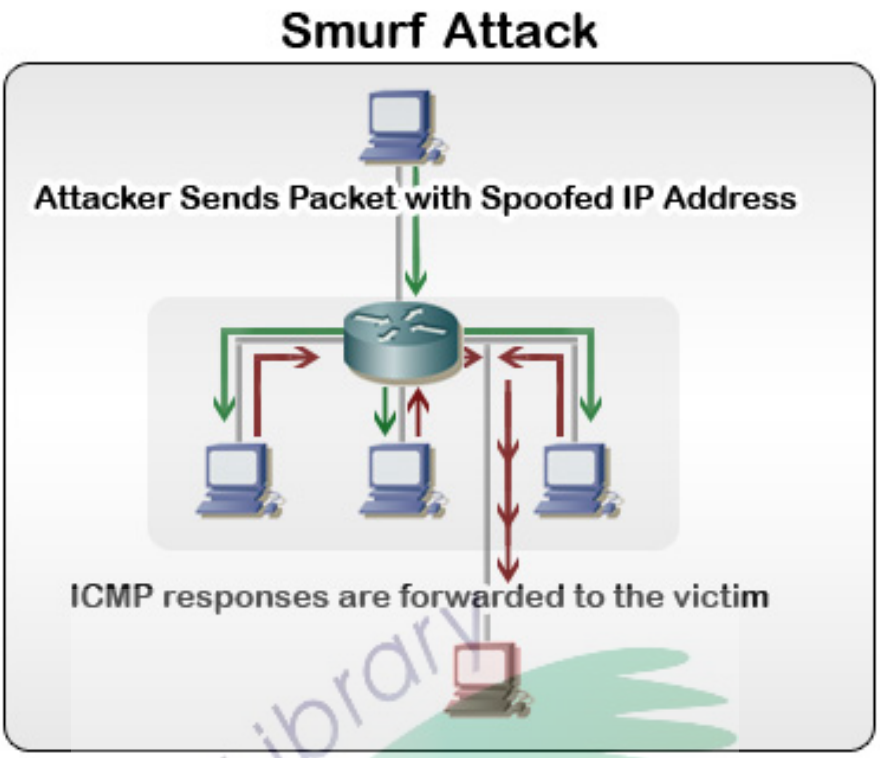


圖 6　Smuf Attack

(五) 混合型攻擊

混合型攻擊 (Mixed attack)：攻擊者分別結合先前提到的 UDP Flood 攻擊、TCP SYN Flood 攻擊和 ICMP Flood 攻擊等攻擊手法，企圖繞過路由器 (Router) 或是入侵偵測系統 (IDS) 的偵測。

(六) Teardrop 攻擊

Teardrop 攻擊 (Santhanam, Nandiraju, Nandiraju, & Agrawal, 2007)(Teardrop Attack) 如圖 7：利用 IP 封包的分割和結合的漏洞來進行攻擊的目的，利用封包傳送過程需要將封包重新組裝的特性，在傳送封包的過程修該 TCP 封包的組裝位移資料，接著等被攻擊主機需要將封包組裝回去時，封包組裝後出現資料錯誤，進而使主機系統產生當機現象，此攻擊手法也會影響到嵌入式系統 (Embedded System)。

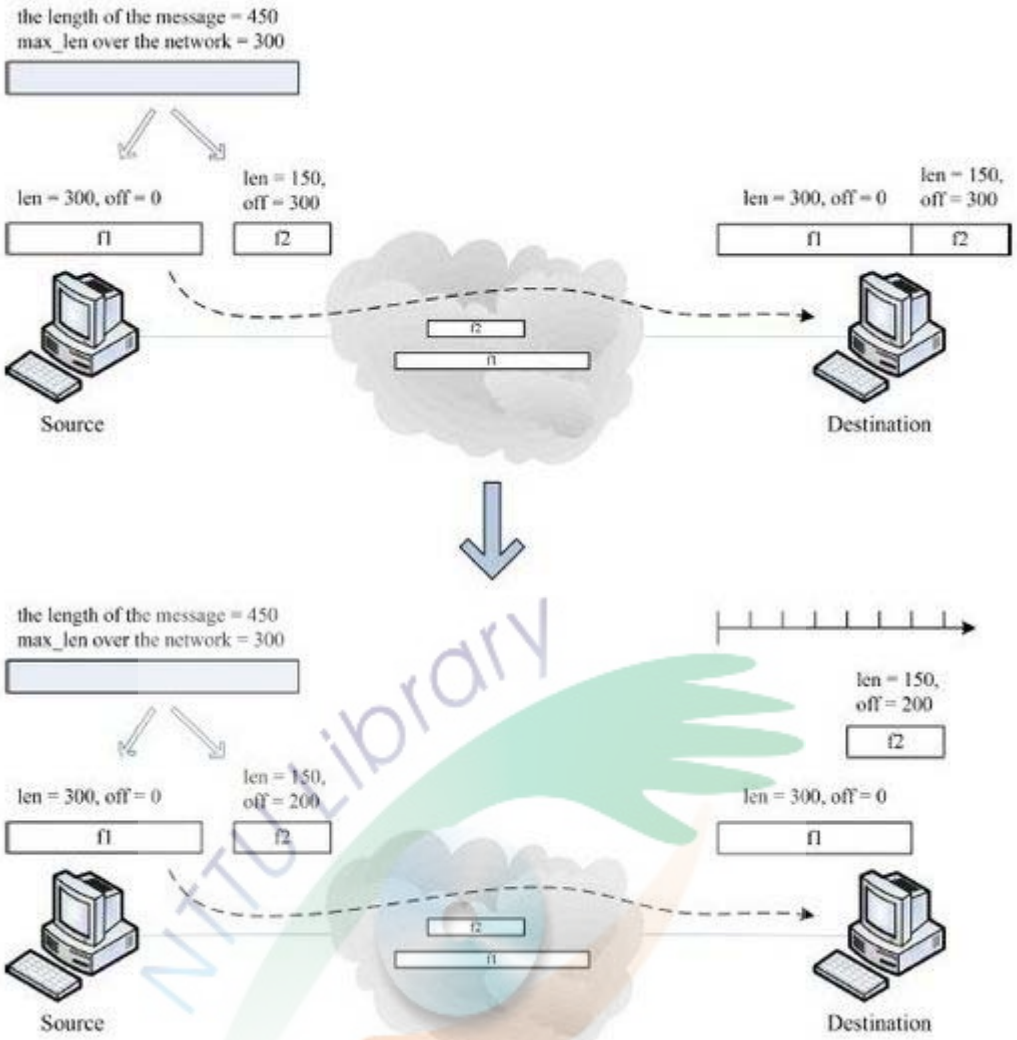


圖 7　Tear Drop Attack

三、 DDOS 攻擊防禦模式

在 A Statistics-based Fuzzy Flow Control Scheme for DDoS Defense(C. Chang, 2007)文章中提及，一般來說，DDoS 的防禦研究主要可以分爲三大類：入侵預防、入侵偵測與入侵回應。入侵預防的目的就是爲了事先要在攻擊封包到達攻擊目標前可提前阻止。入侵偵測就是平常網路流量中用來偵測各種攻擊事件發生的方法。入侵回應則是當攻擊行爲被發現時，用來控制攻擊行爲的各種方法。以下簡單敘述各種方法之行爲：

(一) 入侵防禦(Intrusion Prevention)

最好的策略來防備任何攻擊就是當攻擊還沒發生的時候就預防攻擊的發生，一般而言阻擋外部異常的 IP 爲主，主要是判斷網段外部 IP 的正

確性，如：外部封包來源 IP 不會是網段內部 IP 網段資料，若是則有可能爲 IP spoof；另外，路由器部分也可增加擋不屬於內部的 NAT IP 網段。

(二) 入侵偵測(Intrusion Detection)

在資訊安全的領域中，被動的防守已經不足以因應日益增長的攻擊，所以我們需要主動的偵測出攻擊行爲來採取對應措施。

(三) 入侵回應(Intrusion Response)

入侵回應機制是一旦攻擊被入侵偵測所識別出來時，可以即時採用的回應策略，一般來說是將攻擊的惡意網路流量阻絕。大部分方式是採用網管以手動的方式來來設定阻絕規則給上一層的路由器或限制特定類型的封包通過，或是採用解析封包標頭，再歸納攻擊封包給丟棄，但目前的方式皆無法完全解決 DDoS 攻擊。

第二節 Botnet 及 P2P Botnet 介紹

以下將介紹概述 Botnet 背景，與 P2P BotNet 網路之特性。

一、 Botnet

Bot 是取自機器人(robot) 的簡寫，而 botnet 則是由自動在背景執行代理軟體的惡意程式被控端電腦所組成的。而最早的 botnet(S. Chang, Zhang, Guan, & Daniels, 2009; Dagon, Gu, Lee, & Lee, 2007)是由IRC bots所組成，在(陳怡綾, 2008; 曾瑞瑜, 2008)文章中也分別提到這些 IRC botnet 是透過病毒感染而導致，這些被控端的電腦，主要是依靠 IRC 通訊來與通主控端作溝通，主控端會透過事先定義的 IRC channel 新來散佈攻擊資訊或是惡意軟更新，被控端接受指令之後在開始對特定目標發動網路攻擊。

而主控端主要依靠惡意程序來感染這些被控端，一般常見的惡意程序會藏匿在正常網路服務當中，如網頁服務，信件服務，但是當電腦系統沒有即時做好安全性更新就容易被植入惡意程式，而當一個初始的惡意程式被執行時，便會回到駭客端進行最新的病毒程式更新，並且引發更多的淺在系統安全攻擊，而最後被控端電腦就成爲殭屍網路的一員。

而常見的殭屍網路所發起的初始攻擊大部分是要擴展更多的殭屍網路同儕，所以會藉著廣告軟體，木馬軟體或是垃圾信件來散撥病毒。而真正大規模攻擊則是發動 DDOS 攻擊，將對方網路資源耗盡以無法提供正常網路服務。爲了避免被依靠 IP 位置而被找出真正發動攻擊處，最近殭屍網路又發展出 Fast flux，將散播的中繼站透過快速的 DNS 查詢位置變動，來避免 IP 位置的曝露。

二、 P2P Based BotNet

以 P2P (Peer-to-Peer)爲散佈架構的殭屍網路，透過 P2P 技術的特性讓每個在殭屍網路上的電腦都是客戶端也是伺服端，透過預設的同儕清單連

線上殭屍網路，彼此間形成密集的網路拓撲結構。由於不需要經由特定的伺服器，殭屍電腦可以快速經由同儕端電腦下載程式或接收指令，攻擊則可從任何一台 P2P 殭屍網路上的殭屍電腦來進行控制整個 P2P 殭屍網路。

三、 P2P botnet 與一般 P2P 軟體網路行爲模式比較

此次研究主要是針對 P2P Botnet 病毒(Noh, Oh, Lee, Noh, & Jeong, 2009)，而爲了要分辨其中與正常 P2P 軟體的網路行爲模式，故以下敘述正常 P2P 軟體與 P2P BotNet 不同之處：

(一) 正常 P2P 軟體行爲

正常 P2P 軟體會事先安裝在電腦中，開啓連線時會開啓通訊埠，並開始更新同儕清單(peer list)同儕清單上之 IP 爲以存在網路上的電腦，而新加入的電腦必須先連線到這些清單上的 IP 才能開始加入網路；當開始加入之後，藉由連上的電腦再繼續更新同儕清單，再繼續相互連接。由於同儕清單的電腦不一定都是連線狀態，所以發出的連線封包會有不少是沒有回應的。

所以，正常的 P2P 軟體在連接網路前，可能會有開啓通訊埠及短時間內連接不同 IP 的連線數量。當連線之後則開始進行資料之傳送，如檔案下載。

(二) P2P BotNet 行爲

如最近被大量研究的 P2P BotNet 程式爲 Peacomm(Grizzard, Sharma, Nunnery, Kang, & Dagon, 2007)，最主要是因爲它是採用 DHT 的 kademlia P2P 網路，感染方式爲使用者點選電子郵件的附加檔案而被感染；其他 P2P BotNet 也各自擁有自己的行爲模式，如 Sinit(Wang & Laih, 2008)病毒則是經由 IE 漏洞，攻擊者在網頁上植入惡意程式之後，使用者如不小心點選該網頁，則同時也執行了該病毒，接著受害電腦便開啓 TCP、UDP53 port 進行更新惡意程式。還有 Nugache(Fischer, 2007)的病毒則是透過 MSN、電子郵件附件及微軟系統漏洞的散播，接著便會自動連上其他同儕電腦，並開啓 TCP 8 port 等待接受及提供新受害者更新惡意程式。

最後，SpamThru(Stewart, 2006)則是手動被感染，如點選惡意程式之後，透過 P2P 連接 BotNet 之 Server 下載惡意程式，並開始廣發垃圾信件。

綜上所述，各類 P2P 連接到控制端及下載惡意程式的方式都不太一樣，但是其實網路攻擊模式或運行週期大都可分爲以下階段：感染-連接-下載-攻擊等四階段，如下圖：

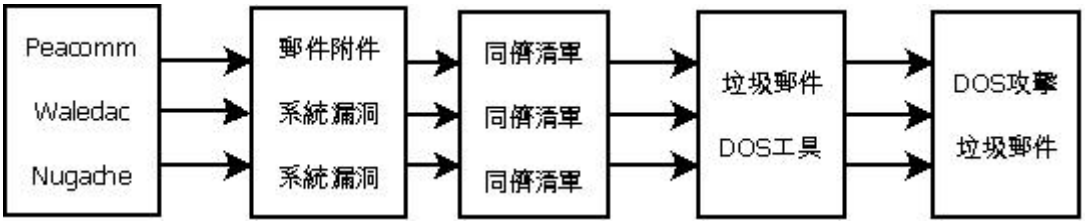


圖 8　Botnet 生命週期

第三節 現有偵測技術

一、 本機端

主機型偵測方法是先定義殭屍程式會有的動作(如:新增特定名稱的檔案檔案，修改 windows regist)，並將所使用到的系統呼叫爲有問題的系統呼叫，則系統上執行中的程序若有使用到這些系統呼叫並且所使用的參數是從網路收到的情況即是可疑的程序，將可疑的程序作記號並追蹤即可偵測出殭屍程式

二、 網路端

主要介紹利用網路流量及 DNS 紀錄分析爲主的文獻。網路流量偵測(Gu, Zhang, & Lee, 2008)在文章中提及，利用分離網路中的 HTTP 及 IRC 流量來進行監測。該研究中架構如圖 9 所示，當一般網路流量進入時，先隔離常用網域流量如 Google 及 yahoo 等網站流量。接著將網路流量來監測其中 Botnet 中的 IRC 及 HTTP 流量並將其中流出流入訊息分別記錄、惡意程式流量(如垃圾信件、惡意程式下載)，最後送進其關聯引擎作 Botnet 流量分類並寄送報表。

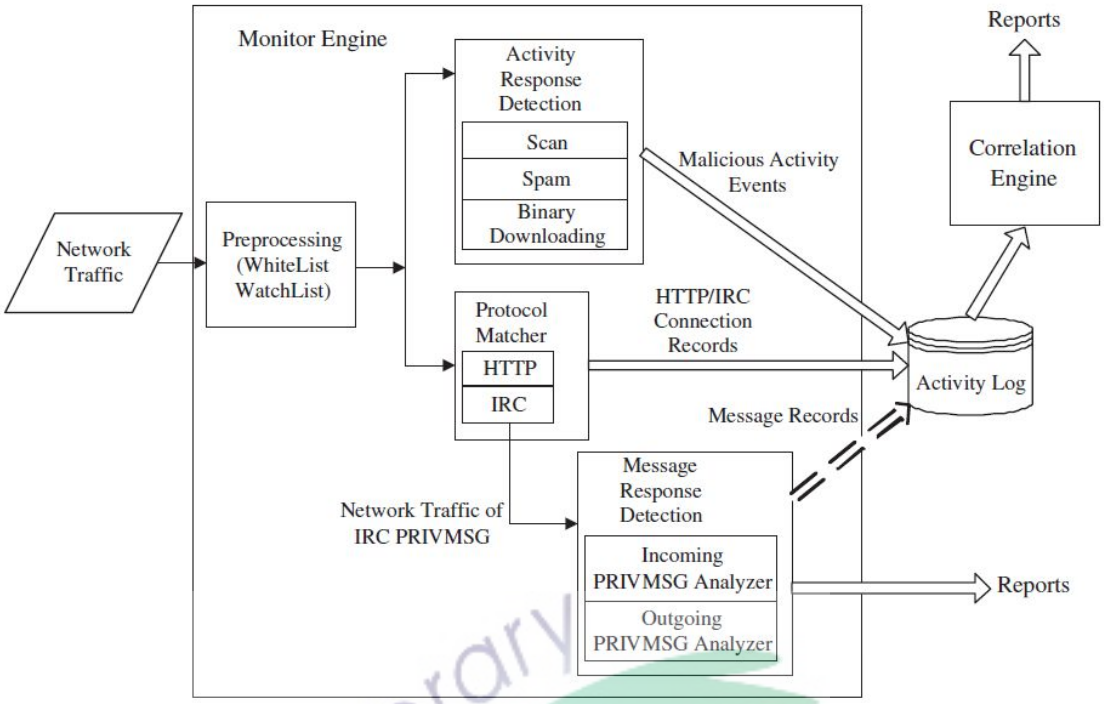


圖 9 BotSniffer(Gu, et al., 2008) 系統偵測流程示意圖

其他在(Choi, Lee, & Kim, 2007)利用監測 DNS 封包的 Group Activities 特性來偵測僵屍網路：原理是利用 DNS 查詢的封包來作分析，將其分爲合法的要求跟有風險的要求，透過此方式來判定是否有 botnet 組成的風險存在，進而阻止 botnet 的形成。該研究設計一個演算法來判斷 DNS 的 traffic 是否合法，而此演算法的精髓就是 group activity。原理如下：

(一) botnet 網路起始聯絡階段：有漏洞的電腦被入侵之後，當被安裝了 bot code，會想辦法與 bot master 做聯繫，此時會透過 bot code 中預設的 C&C server 來聯繫，而 C&C server 的位置會用動態網域名的方式儲存，以增加 bot 組成的可更動性與成功率。在此就會有 DNS query 的發生。

(二) 備用 C&C 聯絡階段：如果 bot code 中預設的 C&C server 連不上，那麼所有中此毒的電腦就會連上候補的 C&C server，當然這裡也會有 DNS query。

(三) 當 bot master 命令 bots 們要轉移到 C&C server 或是只是他們向 server 下載程式時，也會有 dns 的 query 發生。針對要蒐集的 DNS qurty，對 query 依照時間還有 IP 來源分群；將部分設定爲合法的 IP 來源從資料庫中剔

除；接著將 S(similarity)給計算出來，如果 S 越接近 0，表示他是合法的可能性越高，算出的 S 若越接近 1，那是 botnet 的 traffic 可能性越高，如果是-1，則列入持續觀察的清單。

下式中的 A、B 代表的是 IP list 的 size，而 C 代表著 A 與 B 兩個 IP list 中相同 IP 的數量。透過此方式，可以計算出 S，當 S 大於門檻值時就可針對有問題的 IP 做處理。簡單的說，C 越大代表著 A 與 B 兩個 Table 中重複的 IP 數越多，也就是連動性越大。又 C 與 S 成正比，所以當 S 值越接近 1 時，這些 IPAddresses 爲 bot 的機率就越高

$$S = \frac{1}{2} \bullet \left(\frac{C}{A} + \frac{C}{B} \right) (A \neq 0, \ B \neq 0)$$

而在其他文獻提及基於 DNS 通信數據挖掘的 Botnet 檢測方法研究(Ishibashi, et al., 2005)顯示，DNS 封包中的資訊有限，不容易用來分辨一個 DNS 查詢是由正常的使用者或惡意的行爲所引起，爲此作者使用“RIPPER(Repeated Incremental Pruning to Produce Error Reduction)”演算法針對 DNS 封包資料進行挖掘。該作者透過從網路流量中過濾出 DNS 封包的方式來蒐集資料，雖然會比直接向 DNS 伺服器的管理者拿去資料來的不容易，不過相對的也才能蒐集到不是使用本地 DNS 伺服器的數據。而蒐集到的數據當中發現約有 10%的 DNS 數據不是使用內部的 DNS 伺服器，而查詢類型除了常見的 type A 與 PTR 外還有大量的 MX 與 AXFR/IXFR 類型。而針對蒐集來的資料是透過 Ripper 數據挖掘演算法，此演算法的特性在於他有很好的歸類準確性和簡單的規則。系統在真實的校園環境中測試的結果也證實了此方法是有效的。

第四節 其他相關文獻

在 Revealing Botnet Membership Using DNSBL Counter-Intelligence(Ramachandran, Feamster, & Dagon, 2006)文中，提及 botmaster 會利用已經定期偵測哪些 DNS 是被列爲黑名單，而藉此進行反偵查來找出 botmaster 的位置。

在 Characterizing Botnets from Email Spam Records(Zhuang, et al., 2008)文中，則是提及利用偵測 spam mail 位置來找出可能的 Botnet 控制端。

在 An Investigation and Implementation of Botnet Detection Schemes(Wang & Laih, 2008)文中，則是提及利用 emulab 來偵測 IRC Based Botnet，而不需爲了其組成環境而困擾。

要預防 BotNet 主要可藉由兩種形式：一種是本機端上安裝防毒軟體及病毒碼需要定期更新，但是此中作法需要每位使用者有高度的配合，否則一但有漏網之魚，也是一樣會遭受 BotNet 的植入；另一種則是，從網路管理中隔開各種 BotNet C&C 連接途徑，可以利用網路流量、DNS 紀錄或者封包擷取方式加以得到控制，相較於主機端的控管方式，網路偵測模式有相對較大的變化機制可供應用。

第三章 系統設計

本章根據研究目的及有關文獻探討的結果，提出本研究的研究架構、系統架構、研究工具、研究程序以及資料分析等小節，以說明本研究之實施方法與步驟。

第一節 研究架構

本文主要研究架構如下圖。

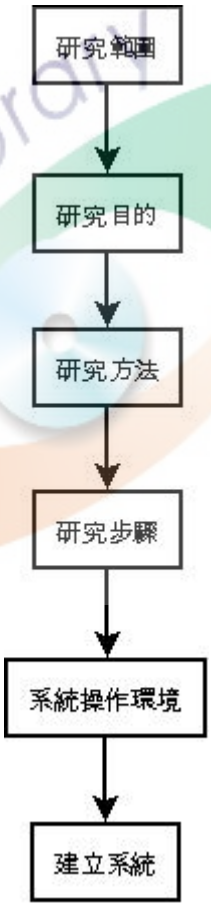


圖 10　研究架構圖

本研究旨在小型區域網路環境中利用虛擬機器模擬感染 botnet 病毒的電腦，並在其網路架構中擺放網路流量偵測系統，從旁側錄網路流量，並針對已知的

botnet 網路流量特徵加以記錄，再產生相關流量報表以網頁形式呈現，期望能在流量報表頁面以快速的方式找出潛在中毒電腦。

第二節 系統架構

實驗環境中，將會以寬頻網路的 ATU-R 模擬成 nat 主機，由一台電腦 A（IP：192.168.1.101）安裝 VirtualBox 虛擬機器，再從此台機器中模擬中毒 pc 共三台如 B（IP：192.168.1.102）,C（IP：192.168.1.103）,D（IP：192.168.1.104），其中每台網路架構是採用 bridge mode，藉此可以使得 abcd 四台電腦皆視爲同一網段，以模仿學生宿舍電腦爲同一樓層，並在 A 電腦上架設此套網路流量分析系統，來進行含有各種殭屍網路特徵網路流量的統計，網路架構圖如下。

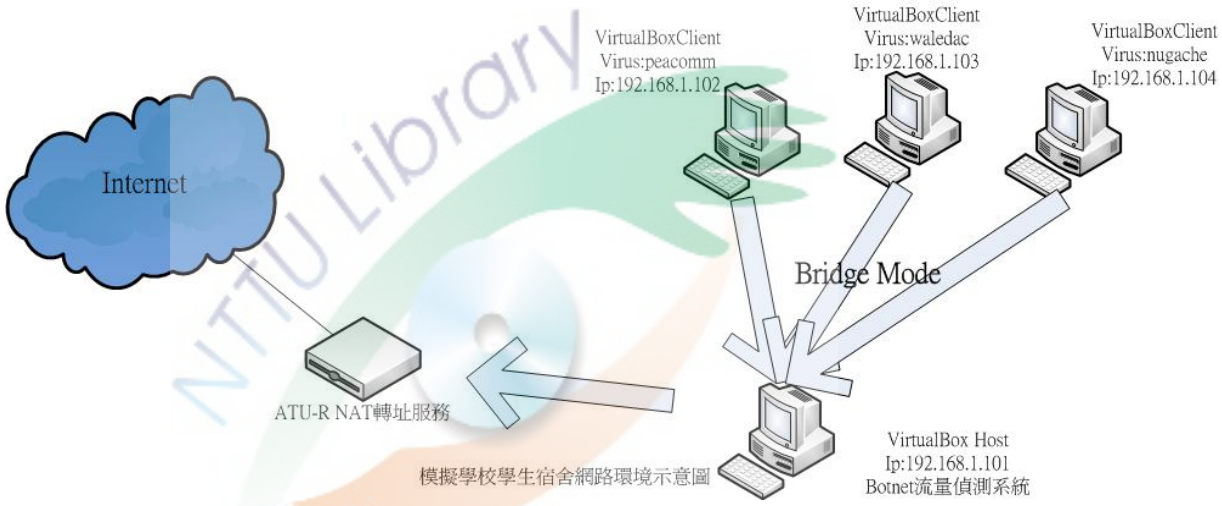


圖 11　網路架構圖

各虛擬電腦的中毒資料及 IP 如表 1：

表 1 虛擬電腦模擬中毒一覽表

電腦代號	電腦 IP	病毒種類	備註
A	192.168.1.102	Peacomm	VirtualBox 虛擬機器
B	192.168.1.103	Waledac	VirtualBox 虛擬機器
C	192.168.1.104	Nugache	VirtualBox 虛擬機器
D	192.168.1.01	無	實體電腦 Bridge 端

系統主要接收網路流量來自 ATU-R 底下的電腦，只要在該環境中使得 a 電腦加裝 libpacp 程式即可透過設定監聽網卡，而開始抓取流量資料。故，若是要採取一般有 switch 功能交換器之網路，因爲 switch 交換器本身有紀錄 port 對 mac 之對應，所以無法監聽非本機端流量的狀況，但若是有支援網路管理的交換器，則可透過設定 port mirror 方式，將特定網路在進行複製而達到進行監聽得目的。

而本實驗系統 client 電腦所採用 NAT 模式，因爲 client pc 上網須透過 ATU-R 作轉址，且利用 a 電腦和其他三台中毒電腦是藉由 bridge mode 來作橋接方式，實際上三台中毒電腦實際流量仍需透過 a 電腦網卡作轉送至 nat 之後才能送往 internet，所以一定可以偵測到中毒 client 之網路流量。

系統一但接收到網路流量之後，會針對其封包特性，如 OSI 七層之 Transport 層的 TCP，UDP 及 ICMP，Neteork 層的 IP 資料加以分類並收集統計並繪製相關網路圖形。其收集網路封包架構如圖 12：

圖 12 網路封包收集架構圖

此次系統運作環境主要來自筆者自行模擬之網路架構，NAT 轉址服務採取 ATU-R，並於 a 電腦加裝 apache 網頁系統，VirtualBox OSE(open source edition)並安裝相關函式庫，對不同網路階層的封包進行擷取、紀錄並畫圖。而系統運作流程如下圖：

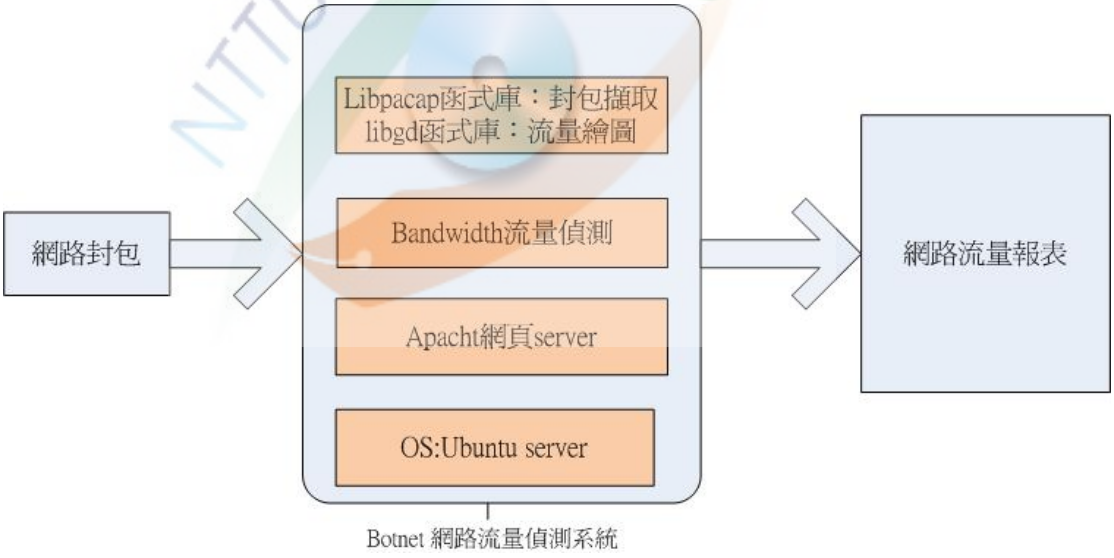


圖 13 系統運作流程圖

第三節 研究工具

網路流量偵測的封包抓取主要依靠 libpcap 函式庫，該函式庫可抓取從 datalink-layer 以上標頭之資料，如採用 ethernet 封裝；Network layer 是採用 IP 協定；Transport layer 是採用 TCP、UDP 或是 ICMP。而本系統主要是應用於模擬學生宿舍，所以是在 ethernet 去讀取 IP 表頭、TCP port number、UDP port number 及 ICMP 通訊協定等相關資料。

封包篩選流程則是利用上述已知 P2P Botnet 網路病毒傳輸的特性。資料傳輸 port 加以規劃。如早期 Botnet 爲執行 IRC 通訊協定來溝通並發動攻擊，而 IRC 通訊協定及爲 TCP port 667；HTTP Botnet 則是經過網頁(TCP port 80)瀏覽來下載、及更新病毒及攻擊指令;而本研究所分析之 P2P Bonet 爲利用 TCP port 8、或是 UDP port 18334 或 12814，來進行資料通訊。

而當開始發動傳輸資料或攻擊時這些受感染之 PC 大部分是針對網頁進行 DDOS，則是利用 TCP port 80；發送 spam mail 則是利用 TCP port 25 來進行，而伴隨攻擊發生也會產生 DNS query 之流量，利用 UDP port 53。且爲了知道被攻擊端是否存在也會發送 ICMP 來進行確認

故本系統特別針對 TCP port 8,25,80、UDP port 53,18214,18334 及 ICMP 來進行收集及篩選，如圖 14。

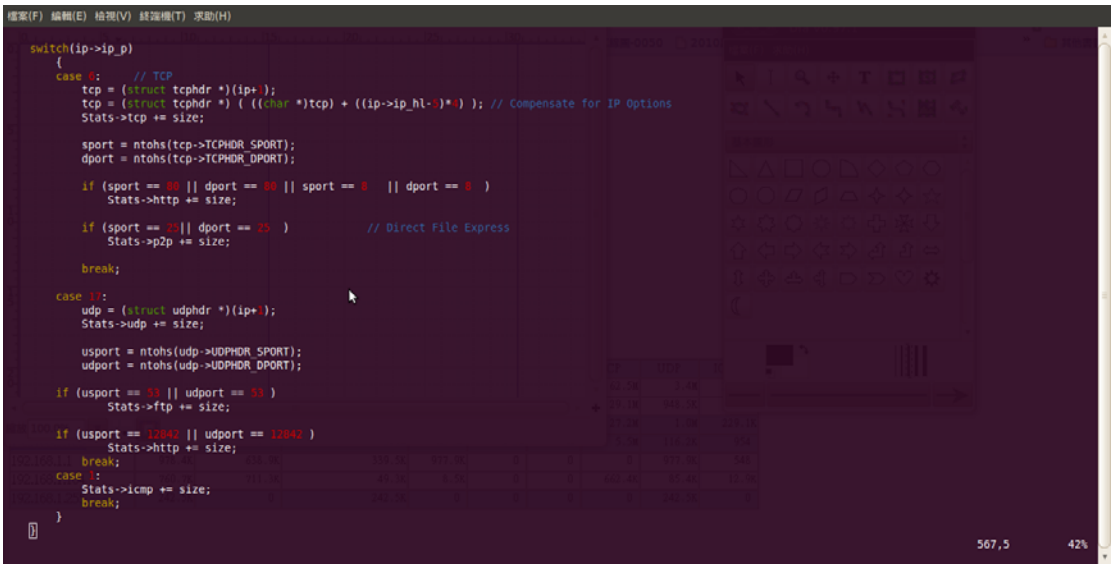


圖 14 Botnet 偵測定義程式碼

第四章 資料分析與研究發現

本章共分二節，旨在根據報表所得之資料進行分析，以瞭解系統是否完全偵測到殭屍網路的網路流量。

第一節 資料分析

本節將依次分析報表頁面以得研究發現與結果。本研究用來偵測殭屍網路的依據主要是針對各種已知的殭屍網路病毒若琪網路傳訊過程中有利用固定通信埠號者則記入在 Botnet 欄位內，這種電腦即是有受到感染的狀況；若是 Botnet 欄位沒有流量紀錄但是電腦有發送 ICMP 的流量，此種則是可能爲異常，可以提供網路管理者一個判斷流量的依據。系統產生報表頁面主要爲：各 IP 所產生的網路流量大小紀錄，如下圖：

Top 20 IPs by Traffic - Daily

Ip and Name	Total	Total Sent	Total Received	DNS	Botnet	MAIL	TCP	UDP	ICMP
Total	9.32M	4.57M	4.98M	2.38M	6.44M	0	7.8M	2.32M	0.22K
192.168.1.102	1.93M	1.56M	603.47K	866.4K	966.5K	0	0.7M	1.0M	229.1K
192.168.1.103	5.27M	1.69M	3.59M	30.4K	5.5M	0	5.5M	116.2K	954
192.168.1.1	978.4K	638.9K	339.5K	977.9K	0	0	0	977.9K	548
192.168.1.104	945.26K	711.3K	241.21K	559K	0	0	905.13K	10.46K	30.64K
192.168.1.255	242.5K	0	242.5K	0	0	0	0	242.5K	0
192.168.1.101	0	0	0	0	0	0	0	0	0

圖 15 系統報表

而藉由 wireshark 側錄虛擬電腦各自產生之網路流量如表 2：

表 2 Botnet 一般網路流量紀錄比較表

電腦 IP	Wireshark -TCP	Wireshark -UDP	Wireshark -ICMP	Botnet 系統準確率		
				TCP	UDP	ICMP
192.168.1.102	736K	1064K	227K	99%	99%	99%
192.168.1.103	5529K	113K	930 Bytes	99%	99%	99%
192.168.1.104	915K	11K	30K	99%	99%	99%

而藉由 wireshark 側錄虛擬電腦病毒網路流量如下表

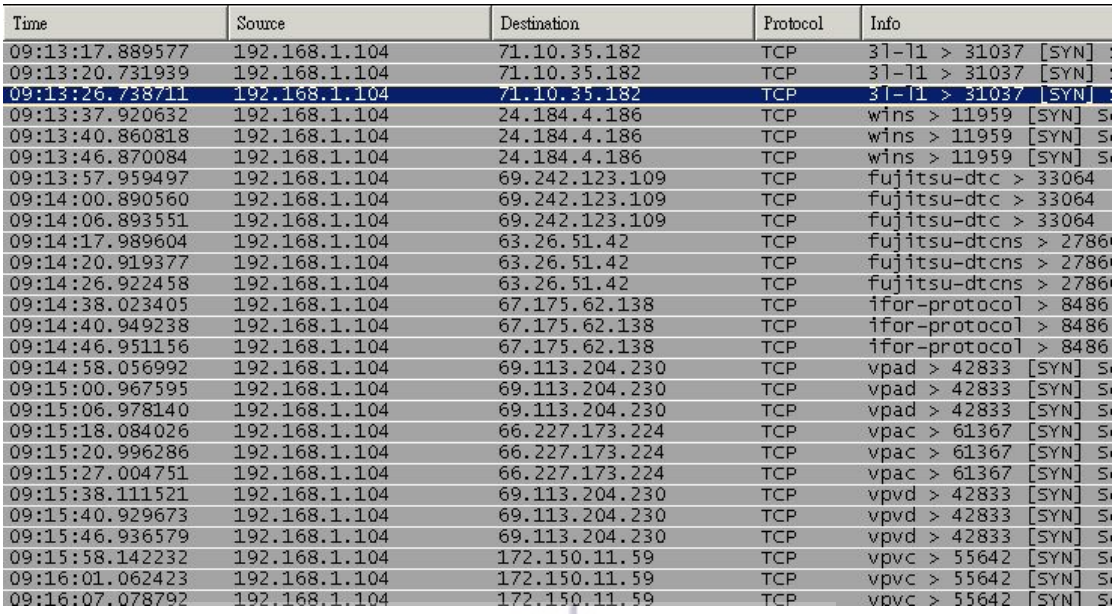
表 3 Botnet 病毒流量紀錄比較表

電腦 IP	系統 Botnet 欄位	Wireshark 病毒流量	準確率
192.168.1.102(Peacomm)	966.5k	970K	99%
192.168.1.103(waledac)	5.5M	5.6M	98%
192.168.1.104(Nugache)	0	914K	0%

由上面兩個表可以得知，本研究所開發的系統針對網路流量可以有極高的準確率，然而對於各 IP 所產生的病毒流量(Botnet 欄位流量紀錄)，以 Peacomm 及 Waledac 兩隻病毒也可以正常偵測，但 Nugache 病毒在系統中則是無法測得。

換去話說，針對 Peacomm 及 Waledac 兩隻病毒，其網路傳輸通訊埠皆是有固定(分別爲 TCP port 80、UDP port 18334)的狀態，可以直接定義在系統的網路流量偵測條件中，故可以達到極高的準確率。

而針對 Nugache 病毒系統卻無法偵測，詳細探究其原因，發現這隻病毒的網路特性是發送固定大小的封包(62 Bytes)資料來傳遞訊息，而非經由固定的通訊埠，所以系統無法偵測其流量，但藉由其封包傳輸過程來觀察會發現這隻病毒的網路行爲特徵是對每個同儕 IP 發送 3 個資料大小爲 62 Bytes 的封包，若是同儕無回應，則繼續下個 IP。如圖 16：



Time	Source	Destination	Protocol	Info
09:13:17.889577	192.168.1.104	71.10.35.182	TCP	3l-l1 > 31037 [SYN]
09:13:20.731939	192.168.1.104	71.10.35.182	TCP	3l-l1 > 31037 [SYN]
09:13:26.738711	192.168.1.104	71.10.35.182	TCP	3l-l1 > 31037 [SYN]
09:13:37.920632	192.168.1.104	24.184.4.186	TCP	wins > 11959 [SYN] S
09:13:40.860818	192.168.1.104	24.184.4.186	TCP	wins > 11959 [SYN] S
09:13:46.870084	192.168.1.104	24.184.4.186	TCP	wins > 11959 [SYN] S
09:13:57.959497	192.168.1.104	69.242.123.109	TCP	fujitsu-dtc > 33064
09:14:00.890560	192.168.1.104	69.242.123.109	TCP	fujitsu-dtc > 33064
09:14:06.893551	192.168.1.104	69.242.123.109	TCP	fujitsu-dtc > 33064
09:14:17.989604	192.168.1.104	63.26.51.42	TCP	fujitsu-dtcns > 2786
09:14:20.919377	192.168.1.104	63.26.51.42	TCP	fujitsu-dtcns > 2786
09:14:26.922458	192.168.1.104	63.26.51.42	TCP	fujitsu-dtcns > 2786
09:14:38.023405	192.168.1.104	67.175.62.138	TCP	ifor-protocol > 8486
09:14:40.949238	192.168.1.104	67.175.62.138	TCP	ifor-protocol > 8486
09:14:46.951156	192.168.1.104	67.175.62.138	TCP	ifor-protocol > 8486
09:14:58.056992	192.168.1.104	69.113.204.230	TCP	vpad > 42833 [SYN] S
09:15:00.967595	192.168.1.104	69.113.204.230	TCP	vpad > 42833 [SYN] S
09:15:06.978140	192.168.1.104	69.113.204.230	TCP	vpad > 42833 [SYN] S
09:15:18.084026	192.168.1.104	66.227.173.224	TCP	vpac > 61367 [SYN] S
09:15:20.996286	192.168.1.104	66.227.173.224	TCP	vpac > 61367 [SYN] S
09:15:27.004751	192.168.1.104	66.227.173.224	TCP	vpac > 61367 [SYN] S
09:15:38.111521	192.168.1.104	69.113.204.230	TCP	vpvd > 42833 [SYN] S
09:15:40.929673	192.168.1.104	69.113.204.230	TCP	vpvd > 42833 [SYN] S
09:15:46.936579	192.168.1.104	69.113.204.230	TCP	vpvd > 42833 [SYN] S
09:15:58.142232	192.168.1.104	172.150.11.59	TCP	vpvc > 55642 [SYN] S
09:16:01.062423	192.168.1.104	172.150.11.59	TCP	vpvc > 55642 [SYN] S
09:16:07.078792	192.168.1.104	172.150.11.59	TCP	vpvc > 55642 [SYN] S

圖 16 Nugache 病毒流量特徵圖

所以若是可針對 Packet 數量來做進一步的偵測規劃，即可偵測到這隻病毒的網路行爲。

第二節 研究發現及實驗結果

一、 系統報表針對特定 port 傳輸資料可產生監控紀錄

本研究中所觀察的病毒裡面，以 peacomm 和 waledac 等兩隻病毒的網路流量是可以被系統所偵測到的；但是若以 waledac 病毒來看，其傳輸的特徵是以 TCP port 80 爲主，與一般瀏覽網頁行爲極爲相似，但若加以考慮 waledac 的 ICMP 流量狀況，便可以得知這是與一般網頁不同之處。而 nugache 這隻病毒沒有在 botnet 流量偵測裡頭，是就可以完全排除其中毒的可能？相同的這隻病毒所產生的 ICMP 流量也是較一般電腦爲高，所以儘管無法列入即時監控名單，但是網管人員亦可多加留意；而系統監控的 MAIL 流量，則都是 0。主要是因爲這些病毒都尚未開始發生生命週期的發動攻擊階段，所以偵測 mail 的流量紀錄都是空白。

二、 本系統與其他相關研究有以下幾點優勢

(一) 偵測系統建置方面

由於軟體環境皆是在 open source 的資源，在 OS 平台方面利用 Ubuntu linux 平台，其核心套件是修改於 Debian linux 平台，這些平台本身擁有高安全性及高穩定度，連硬體需求也不高，且只要是 Unix 平台都是可以免費取得作業系統，故成本上耗費較低，此外研究中所使用的套件 bandwidthd 由 C 語言進行開發，具備有跨平台的特性，只要具備相關函式庫即可編譯完成。

(二) 網路架構方面

本次實驗中是採用 NAT 架構與其他殭屍病毒電腦爲 Bridge mode，所以可以讓網路卡以 promisc 方式進行封包擷取。但若非採用 NAT 模式，是否本系統就無法進行偵測？答案是否，因爲只要交換器有 port miorr 功能則本系統亦可採用。若是採用 routing 方式，則可在硬體環境使用兩片網卡，一片當作收集 NAT 轉送對內的流量，而另一片則是負責對外的路

由動作，則本系統一樣可以監測內部 IP 的網路流量情形。故系統的網路環境適應性極高。

(三) 可外加採用即時防禦動作方面

在本系統中由於作業系統本身就具備有防火牆的功能，所以當偵測到有中毒電腦時其可馬上採取防護或阻絕動作，如採用 Iptables 套件。或若是網路設備具備有網路管理的功能，如 Cisco ip access list，則可撰寫網管設備防護的指令，透過平台自動至網路設備中執行指令並進行阻擋動作。

第五章 結論與建議

實做紀錄 P2P botnet 的流量偵測系統過程中，本研究以網路閘道端的觀點，藉由分析 nat 上的網路流量將殭屍病毒常用的網路特徵作一的收集，並將這些特徵納入本研究的流量系統進行網路封包的收集、分類及統計，藉由此種方式提供給網路管理人員一種快速辨別殭屍電腦的方法。

由於本研究之系統在虛擬環境下確實可以將已知殭屍網路病毒的電腦作一記錄然而，殭屍病毒的種類及演化卻是越來越快速，本研究所採取之特徵對於未知病毒的判別及紀錄，勢必力有未逮；故筆者在研究中發現有兩個面向是可以繼續努力的：

一、 殭屍網路之傳播期：增加紀錄封包數量的統計

中毒初期的電腦由於要試著聯絡 peer 清單上的中毒夥伴但是卻不一定可以連上，所以會發出持續性的少量封包，但是累積流量卻未必也大，所以若是在可以針對特定 tcp 或 udp port 去作 packet 數量統計，則可得連續時間內一直傳送 packet 的圖形，此種圖形跟一般傳輸大檔在短時間內發出數量多的情形大有不相同之處。

二、 殭屍網路之攻擊期：即時監控未知特定傳輸服務

當發動攻擊時，將會出現 spam 或者 DDOS 的攻擊型態，但是由於限今收信的行爲大部份是藉由 webmail 去作收送信的的習慣，剛好與本機以 MDA 方式以 tcp port 25 送信不同；所以若是發生 spam 的現象，則本系統可以監測對外 mail 的流量狀況，加上現今學生使用 MDA 來收發信件的機率很低，所以若是直接由 PC 端所發出的 tcp port 25 SMTP 的網路流量流量，該電腦其實是大有可疑的攻擊行爲狀況產生；然而若非已知的服務，那就是更要努力的方向。

綜上所述，假使能在本系統發生監測異常事件發生時，開發讓系統自動產生阻絕活動的相關防禦行爲，即時來作阻擋，則更可加深資訊安全的縱深防禦程度，以避免對校外電腦產生攻擊及資安事件。

參考文獻

一、 中文部分

曾瑞瑜（2008）。*以活動關連爲基礎的 irc 殭屍網路偵測*。未出版之碩士論文，國立成功大學資訊工程學系碩博士班。

陳怡綾（2008）。*在 irc 伺服器偵測以 irc 爲主的殭屍網路*。未出版之碩士論文，國立中山大學資訊管理學系研究所。

二、 英文部分

Chang, C. (2007). A Statistics-based Fuzzy Flow Control Scheme for DDoS Defense.

Chang, S., Zhang, L., Guan, Y., & Daniels, T. (2009). A framework for p2p botnets.

Choi, H., Lee, H., & Kim, H. (2007). Botnet detection by monitoring group activities in DNS traffic.

Dagon, D., Gu, G., Lee, C., & Lee, W. (2007). A taxonomy of botnet structures.

Fischer, D. (2007). Storm, nugache lead dangerous new botnet barrage. SearchSecurity. com, December.

Grizzard, J., Sharma, V., Nunnery, C., Kang, B., & Dagon, D. (2007). Peer-to-peer botnets: Overview and case study.

Gu, G., Zhang, J., & Lee, W. (2008). BotSniffer: Detecting botnet command and control channels in network traffic.

Hussein, M., & Zulkernine, M. (2006). UMLintr: a UML profile for specifying intrusions.

Ishibashi, K., Toyono, T., Toyama, K., Ishino, M., Ohshima, H., & Mizukoshi, I. (2005). Detecting mass-mailing worm infected hosts by mining DNS traffic data.

Labib, K., & Vemuri, V. (2006). An application of principal component analysis to the detection and visualization of computer network attacks.

Mirkovic, J. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. ACM SIGCOMM Computer Communication Review, 34(2), 39-53.

Naoumov, N., & Ross, K. (2006). Exploiting P2P systems for DDoS attacks. Paper presented at the Proceedings of the 1st international conference on Scalable information systems.

Noh, S., Oh, J., Lee, J., Noh, B., & Jeong, H. (2009). Detecting P2P botnets using a multi-phased flow model.

Ramachandran, A., Feamster, N., & Dagon, D. (2006). Revealing botnet membership using DNSBL counter-intelligence.

Santhanam, L., Nandiraju, D., Nandiraju, N., & Agrawal, D. (2007). Active cache based defense against dos attacks in wireless mesh network.

Stewart, J. (2006). SpamThru trojan analysis. Secureworks. com, 18.

Wang, S., & Laih, A. (2008). An Investigation and Implementation of Botnet Detection Schemes.

Zhuang, L., Dunagan, J., Simon, D., Wang, H., Osipkov, I., Hulten, G., et al. (2008). Characterizing botnets from email spam records.