殭屍網路活動的偵測與阻絕工具研發

丁光立
高雄師範大學
資訊教育研究所研究生
tiserle@gmail.com

楊中皇
高雄師範大學
資訊教育研究所教授
chyang@nknucc.edu.tw

摘要

網路犯罪行為日益增加，如釣魚網站、錢騾、竊取個資及販賣、DDoS (分散式阻斷服務)等案例時有所聞。在諸多類別中，DDoS 屬於主動攻擊的型態，為攻擊者在有其目的下進行。DDoS 往往利用殭屍網路(Botnet)做為攻擊的來源，並使用木馬、蠕蟲等作為散佈及傳播的工具，以增加所控制之殭屍電腦的數量，在成功植入控制程式後，控制者(botmaster)再透過命令及控制伺服器(C&C Server)加以控制被感染的電腦(bots)，從而進行大規模的攻擊行動。殭屍網路的控制方式有許多變化，並且控制時可能進行加密，增加偵測上的困難。本文回歸網路最基本的觀念，先分析殭屍網路的行為及活動方式，再使用開放原始碼軟體的網路監控工具：ntop，對網路進行監控，最後開發系統，對可疑的網路行為進行攔阻，防範於未然，避免產生大規模的殭屍網路活動。

關鍵詞：殭屍網路、控制者(botmaster)、命令及控制伺服器(C&C Server)、被感染的電腦(bots)、ntop

Abstract

With internet crime growing, such as phishing, money mules, personal data stealing and trafficking, DDos, and other cases often heard by people from time to time. DDos is a deliberate attack, and the attackers always have a motive. DDos mostly uses Botnet as source of attack, and distributes trojans and worms to infect hosts. Infected hosts become bots, and could be controlled by the the botmaster. Because command and control servers use dynamic types and encryption methods to communicate with bots, it's difficult to detect bots. We try to detect bot-like traffic, and develop a system to deny traffic of who looks like bots.

Keywords: Botnet, botmaster, C&C Server , bots, ntop

1. 前言

在網路頻寬提昇及電腦運算能力日益增加的情形下，分散式運算也開始被廣泛的應用。但原先正面的工具或是想法，往往被有心人士加以利用，並且用於負面的用途。分散式運算的概念同樣的被駭客加以利用，使用分散式架構的攻擊行為與日俱增，以下為近年來較重大的攻擊案例：

2003 年美國奧勒岡州駭客控制 2 萬台殭屍網路主機，對 eBay 網站發動阻斷式服務攻擊[2]。

2004 年 10 月中國大陸河北駭客操控 6 萬台殭屍網路主機連續 3 個月同時發動攻擊北京某音樂網站，造成該網站癱瘓[2]。

2005 年新型殭屍網路病毒 Zotob 發動阻斷式服務攻擊多家美國知名公司網站[2]。

2006 年 2 月，刑事局與微軟清查後發現台灣遭受殭屍網路感染主機約為 5 萬 7783 台，至 9 月 17 日更高達 8 萬 8136 台，僅半年期間受感染數快速增加 3 萬台主機[1]。

據美國聯邦調查局(FBI)的統計發現，超過 75%網絡詐騙案件，是透過垃圾電子郵件進行，且單是 2007 年的詐騙金額就超過 2.39 億美元[4]。

根據 Gais 實驗室統計分析，95%垃圾電郵正是由殭屍網路所發出[4]。目前 DDoS 主要利用殭屍網路來增加其攻擊來源，此外殭屍網路也已成為垃圾郵件的最大來源，不僅對網路安全造成極大的威脅，並造成網路資源的浪費。殭屍網路有著不易偵測、不易防禦的特性，殭屍網路的控制者在控制殭屍網路時，會使用許多不同的方式，例如使用 IRC(Internet Relay Chat)、IM(Instant Messaging)、P2P(Peer to Peer)等通訊協定進行通訊，並且在通訊過程中進行加密，增加了偵測上的困難。本文使用開放原始碼的 ntop 程式，經過改寫後，對網路活動進行偵測。為簡化偵測方式，我們不針對 ISO(International Standard Organization)所提出之 OSI(Open System Interface)網路七層中的第

五、六及七層作監控，原因為這些較高的層別在監控上難度較高，並且需要更多的資源及時間才能得到讓人滿意的結果，況且若通訊過程中有加密時，則會增加大量的時間成本。因此我們針對較易於辨別的網路層和傳輸層作監控，利用殭屍網路活動的特性來偵測殭屍網路存在與否。開發系統將可疑的網路活動加以阻絕，以避免災害擴大及減少受害者。並計畫結合開放原始碼軟體中的遠端弱點偵測掃瞄軟體 Nessus 對這些可疑的受害者進行主動偵測，確認是否已受到感染或者有需要修補的漏洞，再透過管理者告知受害者及進行後續處理。最後計畫將此系統在學校等相關單位部署，以印證其成效。

2. 文獻探討

2.1 殭屍網路的來源

在早期 IRC 蓬勃發展的情形下，管理者需要一些重複性的功能來協助管理，例如聊天室的管理、記錄事件和對話及防止頻道被濫用等，因此有人設計了一些良性的管理用程式來協助管理。

1993 年 12 月，Robey Pointer 使用 TCL 語言設計了 Eggdrop 的程式，供管理者使用。但駭客亦使用相同的思維模式，設計了供私人目的使用的殭屍程式[5]。

1999 年，Mobman 所撰寫的名為 SubSeven (Backdoor-G)的後門程式，在 V2.1 後開始使用 IRC 協定來構建攻擊者對殭屍主機的控制通道[3][15]，這是第一個具代表性意義的殭屍程式。之後 GTBot、Sdbot 等程式廣為流傳[13]，此時使用 IRC 協定的程式開始成為主流。

在 2003 年之後，蠕蟲技術日漸成熟，殭屍程式也搭上這個順風車，使用主動傳播技術大量而快速的傳播，比較出名的有 Deloder[13]。

2004 年後，出現了 P2P 殭屍網路，殭屍程式本身包含了 P2P 的 client，連入採用 Gnutella 技術（開放原始碼的共享技術）的 server，並利用 WASTE[10](採用 RSA 加密的 P2P 文件共享協定)進行通訊。使得每一台殭屍主機可以很方便地找到其他的殭屍主機並進行通信[17]。

殭屍網路依其特性可以分類如下：隨機型、集中型和 P2P 型[6]，因此對應的偵測及攔阻方式也會有算不同。

2.2 對殭屍網路偵測的相關研究

殭屍網路造成的危害及威脅相當的巨大，針對殭屍網路的特性及活動方式，有許多專家學者提出偵測殭屍網路活動及散布的方法。

Chi, Z.及 Zhao, Z.[6]提出利用 Router 的 ID 作識別並結合 IDS 系統來判別該路徑是否有殭屍網路的攻擊行為發生，以便管理者進行後續處理。

Zou, C.C.及 Cunningham, R.[18]發現使用 honeypot 偵測殭屍網路時駭客容易察覺，因此他們提出使用雙重 honeypot 的方式，讓第一個被植入殭屍程式的 honeypot 感染第二台 honeypot 並回報給駭客，讓駭客無從得知所成功植入殭屍程式的是 honeypot，並且使用一種蠕蟲來建置 P2P 殭屍網路，以證明他們的方法可以降低 Hacker 發現 honeypot 的機率。

Villamarin-Salomon, R.及 Brustoloni, J.C. [16]因被感染的殭屍電腦會高速而且集中的對 DDNS Server 作存取，並且查詢記錄中會有 NXDOMAIN(該 Domain 不存在)的情形，藉此查獲殭屍網路的存在。

Choi, H., Lee, H.及 Kim, H.等三位學者[7]發現傳統的透過監視 DNS Traffic 以偵測殭屍網路的方法容易被破解，因此提出監視殭屍網路對於 DNS Traffic 的群組活動，並且在校園網路中作測試。

Kugisaki, Y., Kasahara, Y., Hori, Y.及 Sakurai, K.等四位學者[12]則從 IRC 殭屍網路的 traffic 著手，觀察使用 IRC 的固定 Port 與 Server 連結的 client，查驗其 traffic，並檢查是否為殭屍網路的成員。

Gu, G., Zhang, J.及 Lee, W.等三位學者 [11]建置 BotSniffer 的系統，藉由殭屍網路行為的一致性作判別，並在真實世界使用。

Strayer, W.T., Walsh, R., Livadas, C.及 Lapsley, D.[14]使用過濾的方法逐步將 traffic 做檢查，並架設無害的殭屍網路驗證所開發的管線化系統是否能有效將殭屍網路的流量找出。

表 1 相關研究對照表

學者	使用方法
Chi, Z. Zhao, Z.	利用 Router 的 ID 作識別並結合 IDS 系統來判別該路徑是否有殭屍網路的攻擊行為。

Zou, C.C. Cunningham, R.	使用雙重 honeypot，減少被駭客發現的機率。
Villamarin-Salomon, R. Brustoloni, J.C.	bot 會高速而且集中的對 DDNS Server 作存取，並且查詢記錄會有 NXDOMAIN 的情形，藉此查獲 Botnet 的存在。
Choi, H. Lee, H. Kim, H.	監視殭屍網路對於 DNS Traffic 的群組活動。
Kugisaki, Y. Kasahara, Y. Hori, Y. Sakurai, K.	觀察使用 IRC 的固定 Port 與 Server 連結的 client，查驗其 traffic，並檢查是否為 bots。
Gu, G. Zhang, J. Lee, W.	建置 BotSniffer 的系統，藉由 bots 行為的一致性作判別。
Strayer W.T. Walsh R. Livadas C. Lapsley D.	架設無害的 Botnet 以驗證所開發的管線化系統是否能有效將 bot traffic 找出。

綜觀以上多位學者的研究不難看出，針對殭屍網路的各種特性，已有多種偵測方法被提出，並且許多的研究方法對於偵知殭屍網路的活動均有其成效。但事實上，我們在日常生活中，還是經常會聽到殭屍網路造成企業界或者政府機關損失的新聞及消息。原因為上述這些方法都有其成效，但卻無法輕易的部署在各級單位。有鑑於此，本文試圖去找出一個更輕易偵知殭屍網路的方法，降低其偵察的複雜度，並且降低此系統的硬體及軟體需求。以便在全國中小學部署此系統，讓這些殭屍網路從最底層就受到攔阻，避免造成後續更大的傷害。

2.3 ntop

ntop 是一個開放原始碼的網路流量監控程式。它使用 C 語言開發，可在任何 Unix like 的平台上使用。由於它是開放原始碼的軟體，因此可以用來加以改寫，以滿足我們的需求。ntop 可分為三大部份：

(1) Packet Capture，它利用 libpcap 函式，對網路上的封包進行擷取，再交由 Packet Analyzer 元件進行處理。
(2) Packet Analyzer，其對於常見的通訊協定具備辨識能力，並在網路中對於傳送及接收的封包，依據其通訊協定的類別及進行傳送接收的主機，記錄其類別及封包數。
(3) Report Engine，對於所記錄的數據，經分類排序後轉換成網頁型態，交由內建的網頁伺服器處理，並利用 gd 及 libpng 函式將數據轉換為餅狀或柱狀圖，以利客戶瀏覽。管理者亦可透過瀏覽器對 ntop 進行遠端管理，避免因需另行設定網頁伺服器而增加了系統管理的負擔及並且造成安全性的問題。

2.4 Perl 及 shell script

Perl 是非常易用的程式語言，對於文字的處理相當的便利，本系統主要使用 Perl 的模組開發，在 CPAN (Comprehensive Perl Archive Network)的網頁中[9]，提供了許多便利的模組供使用，本文使用 LWP::UserAgent 模組，此模組的功能為讀取網頁資料。我們使用此模組擷取本系統改寫之 ntop 程式所列出 Active TCP/UDP Sessions 的原始資料，並結合自行開發的 shell script，以提供管理者所需之攔阻及告警功能。

3. 系統設計與實作

3.1 ntop 程式的改寫

ntop 發展已歷時十年，是一個相當穩定，並且功能強大的程式。但若使用 ntop 作為偵測殭屍網路的工具時，所需用到的功能未加以整合，無法讓管理者很快的找到所需使用的功能。為提升其易用性，本研究在主選單中新增 Botnet Detecton 的選單，並且將與殭屍網路相關的功能收集在此選單中，以便管理者查看及管理。所收集的功能如下：

Local to Local 的流量統計：管理者可藉此檢查是否在區域網路中有大規模廣播或者群播的行為。若有大量的主機收到同樣大小的封包，則可能是在區域網路中有試探或惡意攻擊的行為。

Local to Remote 的流量統計：管理者可藉此檢查是否在區網中使用者，是否對遠端網路上的主機 IP 有大規模廣播或者群播的行為。若有大量的遠端主機同時收到同樣大小的封包，則可能是在區域網路的使用者對遠端網路有試探或惡意攻擊的行為。

Remote to Local 的流量統計：查看遠端主機是否有連至本地的主機，並同時傳輸相同大小的資料。若有此行為，則可能是殭屍網路的控制者在控制被植入程式的主機，或者被植入

程式的主機在更新其攻擊程式或者進行 DNS 的查詢動作。

IP Traffic：針對殭屍網路常用到的各種協定，例如 DNS、HTTP、IRC、各種 P2P 協定、MSN 及 AOL 等 IM 協定進行監控。若被感染的電腦有一致性的行為，例如同時查詢某筆 DNS 記錄，造成同時多個 IP 有同樣的流量時，可能就是殭屍網路活動的徵兆。

Host Activity：在一天 24 個小時中，以小時為單位，用顏色區別出電腦的活動頻率。顏色越深代表在該時段電腦活動的頻率越高，0%為無色，0%～25%為淺藍色，25%～75%為淺綠色，75%～100%為紅色。。若有一些主機總是在夜半三更或是特定時段有大量的對外連線，該主機可能就已被植入殭屍程式。

Active TCP/UDP Sessions：針對每一個連線作記錄，其記錄包含來源 IP、來源 Port、目的 IP、目的 Port、收送的資料量，連線起始、終止時間，連線歷時及 TCP 的封包資訊等，以便管理者查看。

本研究將 Active TCP/UDP Sessions 的項次作改寫，並新增為另一個項次，新項次中將不需要的資訊均剔除，僅保留來源 IP、來源 Port、目的 IP、目的 Port，並利用自行撰寫的程式進行後續監控及處理。本系統的架構如圖 1 所示，使用 Cisco Switch，並利用其 SPAN (Catalyst Switched Port Analyzer) 功能[8]，即俗稱的 mirror port，對流經 Switch 的流量進行側錄，而不直接攔截封包。在此架構中，ntop 伺服器僅負責封包分析的功能，不需擔任路由器、橋接器或 NAT 的角色，將其功能單純化，以降低管理者維護上的困難度，並且避免因伺服器異常造成網路中斷或癱瘓，造成更大的災害。

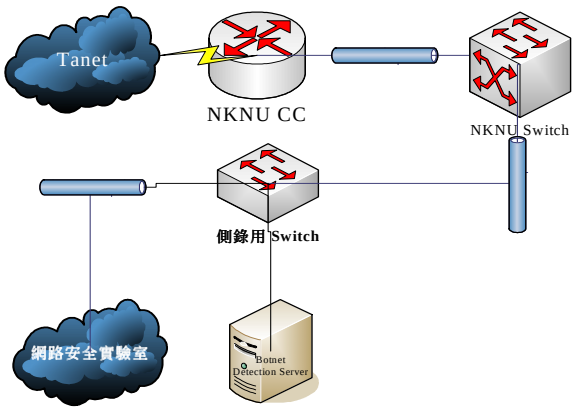

圖 1 系統架構圖

3.2 本研究所發展的監控系統

殭屍網路的活動情形，有以下數種型態：
1、botmaster 透過 C&C Server 下 command 給 bots，網路活動特性為同一來源 IP、不同 Port 連線至不同目的 IP、同一 Port。
2、bots 連線至 C&C Server 更新攻擊碼或下載攻擊程式，網路活動特性為不同來源 IP、不同 Port 連線至同一目的 IP、同一 Port。
3、bots 連線至 DDNS 或 DNS Server 查詢攻擊標的 IP，網路活動特性為不同來源 IP、不同 Port 連線至同一目的 IP、同一 Port。
4、bots 進行攻擊，網路活動特性為不同來源 IP、不同 Port 連線至同一目的 IP、同 Port。
因此本系統針對此四類網路活動型態進行偵測。由於 ntop 本身未提供主動告警功能，因此本研究中，以 Perl 撰寫系統，使用 Perl Module LWP::UserAgent 將擷取改寫過後的 ntop 的 Active TCP/UDP Sessions 資訊，並結合 shell script，以提供主動告警的功能，以增加管理者的便利性。系統主要流程如下：

首先，針對 session 的數量管理，系統原先就預設其上限值，管理者亦可自行設定上限值，若有主機的 session 數超出上限，系統會自動將該主機進行攔阻，並且告知管理者有異常狀況發生，需進行相關處理。

其次，提供黑白名單的功能，避免某些正常行為，或者是重要對外服務主機的連線被本系統進行攔阻。黑名單可由管理者自行設定，也可由系統在運作時自行增加。白名單由系統管理者自行設定。

第三，提供 E-mail 或簡訊方式，主動提醒管理者，網路中有異常的大量 session 出現，並且將超出 session 上限的使用者 IP 告知使用者，提供相關的 Host Activity 及 Active TCP/UDP Sessions 資訊的連結，以便讓管理者能快速的掌握該電腦最近的網路使用情形，以便做出對應的處理。

系統流程圖如圖 2 所示，ntop 程式在系統中一開始就執行，之後執行本系統開發的監控程式，依序讀入黑名單(要進行攔阻的 IP 清單)，白名單(不進行任何攔阻的 IP 清單)。以白名單為優先，監控程式不會將已在白名單中的 IP 加入黑名單。若監控程式在運作過程中，發現有異常的大量 session 時，通知管理者，並自動將該來源 IP 加入黑名單中，再進行流量攔阻的動作，本程式預計也將提供管理者自訂選項的功能，讓管理者決定針對異常的大量 session 是由程式逕行攔阻抑或通知管理者，由管理者

自行決定。

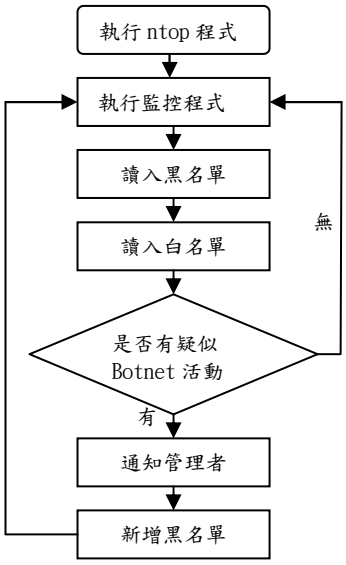

圖 2 系統流程圖

4. 結論

本系統功能已大致開發完成，目前在高雄師範大學資訊教育所之網路安全實驗室進行相關測試。主要目的為測試其穩定性，並且驗證功能是否能達到當初所設定的需求。確認本系統可穩定運作後，將架設至高雄縣市部份中小學及各級院校實際上線運作，以驗證其成效。但單純偵測及攔阻異常的大量 session 並無法印證出本系統是否真正有其效用。殭屍網路活動的時間和地點是無法預測的，因此無法針對架設本系統之前與架設之後的狀況作一比較。故本研究後續計畫結合另一套開放原始碼的遠端弱點偵測掃瞄軟體－Nessus，針對被管理者列為黑名單，及被本系統列為黑名單的用戶，進行弱點偵測及掃描的動作，若掃描結果能為讓主機存在嚴重的系統漏洞或有潛藏的資安問題，則可代表本系統確實能防範在網路中潛藏的危機，讓管理者事先將這些危機解除。另外亦計畫針對在 Host Activity 中在特定時段有大量網路 session 的主機，使用 Nessus 進行掃描及偵測，以達到預警的功能，進而防範於未然，將可能的災害先行消除或減到最小。最後計畫將本系統應用在國中小學的校園網路中，讓管理者可以減輕管理的負擔，並且在殭屍網路活動的初期就加以攔阻，減少後續人力物力的浪費，並減少相關的網路犯罪案件。

參考文獻

[1] 今日新聞，http://www.nownews.com/2006/09/28/339-1996812.htm。

[2] 內政部警政署刑事警察局，http://www.cib.gov.tw/news/news02_2.aspx?no=261。

[3] 諸葛建偉、韓心慧、葉志遠、鄒維，“殭屍網絡的發現與跟踪”，中國網絡與信息安全技術研討會論文集，pp. 183−189，2005。

[4] 聯合新聞網，http://mag.udn.com/mag/digital/storypage.jsp?f_ART_ID=1270342002。

[5] Canvan, J., ”The evolution of malicious IRC bots,” ***Virus Bulletin Conference***, pp. 104-114, 2005.

[6] Chi, Z. and Zhao, Z., “Detecting and Blocking Malicious Traffic Caused by IRC Protocol Based botnets,” ***Network and Parallel Computing Workshops,*** pp. 485-489, 2007.

[7] Choi, H., Lee, H. and Kim, H., “Botnet Detection by Monitoring Group Activities in DNS Traffic,“ ***Computer and Information Technology,*** pp. 715-720, 2007.

[8] Catalyst Switched Port Analyzer (SPAN) Configuration Example, http://www.cisco.com/en/US/products/hw/switches/ps708/products_tech_note09186a008015c612.shtml.

[9] CPAN, http://www.cpan.org/

[10] Dittrich, D. and Dietrich, S., “New Directions in Peer-to-Peer Malware,“ ***Sarnoff Symposium***, pp. 1-5, 2008.

[11] Gu, G., Zhang, J. and Lee, W., “Detecting botnet command and control channels in network traffic,“ Annual ***Network and Distributed System Security Symposium,*** 2008.

[12] Kugisaki, Y., Kasahara, Y., Hori, Y. and Sakurai, K., “Bot Detection Based on Traffic Analysis,“ ***Intelligent Pervasive Computing***, pp. 303-306

[13] Pouget, F., Dacier, M. and Pham, V., “Understanding threats: a prerequisite to enhance survivability of computing systems,” ***International Journal of Critical Infrastructures***, Volume 4, Numbers 1-2, pp. 153-171, 2008

[14] Strayer, W.T., Walsh, R., Livadas, C. and Lapsley, D., “Detecting Botnets with Tight command and Control,“ ***Local Computer Networks*** , pp. 195-202, 2006.

[15] SubSeven, http://dark-e.com/archive/trojans/subseven/21/index.shtml.

[16] Villamarin-Salomon, R. and Brustoloni, J.C., "Identifying Botnets Using Anomaly Detection Techniques Applied to DNS Traffic," ***Consumer Communications and Networking Conference***, pp. 476-481, 2008.

[17] Wang, P., Sparks, S. and Zou, C.C., "An advanced hybrid peer-to-peer botnet," ***Hot Topics in Understanding botnets***, 2007.

[18] Zou, C.C. and Cunningham, R., "Honeypot-Aware Advanced Botnet Construction and Maintenance," ***International Conference on Dependable Systems and Networks,*** pp. 199-208, 2006.