主動式傳播殭屍網路防禦機制
A Defense Mechanism for the Active Spread of Botnet

葉奇鑫　　曾黎明　　彭博涵
國立中央大學
candle576@dslab.csie.ncu.edu.tw
tsenglm@cc.ncu.edu.tw
natata@dslab.csie.ncu.edu.tw

摘要

殭屍網路是目前最嚴重的資訊安全威脅之一，原因在於殭屍網路是結合多種惡意程式特色的攻擊手法，攻擊者具有高度隱密性並能同時一對多來操控整個殭屍網路進行惡意行為。

主動式入侵方式是指攻擊者去利用漏洞入侵到受害者電腦，並取得權限去安裝惡意程式。我們將針對主動式入侵攻擊的方式提出防禦機制，利用動態延伸蜜罐誘捕系統(Dynamic Extensible Two-way Honeypot)機制提供連上特殊的偽造 C&C 伺服器來作為解毒的方法以防止主動式殭屍網路的傳播。

關鍵詞：殭屍網路、主動式入侵、誘捕系統、解毒、殭屍網路傳播

Abstract

Botnet is a combination of a variety of unique malware attack techniques. Victims install and execute the botnet malware by themselves, because they have a bad habit of network usage. Even if their computers have been cleared the malware, they are likely to be a bot again in a short time. It explains the passive decoy botnet is difficult to prevent. We will focus on the defense mechanism of active intrusion, and we use dynamic extensible honeypot system to provide a special connection to the server and remove malwares to prevent the active spread of botnet.

Keyword：Bot、Botnet、Active intrusion、Honeypot、Active spread of botnet

1. 前言

殭屍網路(Botnet)是目前最嚴重的資訊安全威脅之一[1][2][3]，它是指一群感染惡意軟體並受駭客控制的電腦，受感染的電腦(Botclient)將會猶如一個殭屍而任由擺佈，攻擊者可利用這些殭屍電腦來竊取使用者私人的資料和帳號密碼，特別是電子商務行為的重要資訊，或者用來發送垃圾郵件及釣魚郵件可以用來進行分散式阻斷服務攻擊癱瘓特定電腦[3]。

在偵測的手法上也都努力的發展出相對應的方式去檢測，像是透過部署誘捕系統[4]，如：蜜罐(Honeypot)，去找出經常性的不正常對外連線，這是利用 Botclient 必須定期的向 C&C Server 聯絡以便取得最新的配置檔或是所發佈惡意指令，相較於一般電腦的正常連線，這項特殊的特徵就能夠來作為偵測的因素之一，在利用統計方法或是演算法[5][6][7]來設定偵測門檻值，進而判斷是否有殭屍網路中毒的電腦個體。殭屍網路近來常使用新興的隱藏技術「fast-flux」[8]，來使的他們更難被偵測到，這是針對防禦方法：「封鎖特定 IP」的改良策略，這技術會不斷地輪流改變殭屍網路 C&C Server 的 DNS 紀錄，讓殭屍網路難以被查獲。

根據實際安裝者的差異，我們可以分成兩種情境：主動式攻擊漏洞與被動式誘騙攻擊。前者係因受害者電腦存在某些漏洞，使得攻擊者去利用這些漏洞入侵到受害者電腦，因此取得權限或利用程

式去下載程式檔及同時安裝。後者則是透過垃圾郵件、網頁誘騙或是將惡意程式嵌入一些應用軟體，讓粗心的使用者去將殭屍網路病毒的程式安裝起來，這類型的受害者是在無知或是相關知識較為不足的情況下，遭攻擊者透過誘騙的方式去裝設了殭屍網路程式。

對於被動式誘騙的攻擊型態而言，即便清除病毒之後，因為具有同樣的不良使用習慣，往往在短期間內又讓電腦感染了殭屍病毒程式，而又成為了殭屍病毒的其中一員，要真的隔絕此類攻擊行為並不容易。針對這些困擾，我們希望可以先針對主動式攻擊型的殭屍網路傳播來提出一個新的防禦機制。

本論文後續章節架構如下：章節二探討相關背景知識及研究；章節三分析相關研究並並提出新的防禦機制；章節四實作系統並討論防禦機制可行性；章節五結論與未來發展。

2. 相關研究

2.1 宙斯殭屍網路分析

宙斯殭屍病毒是使用 HTTP Protocol 的通訊協定[9][10]，透過網站伺服器來下達命令給 Botclient，隱藏其通訊流量於正常網頁瀏覽所產生的流量，因 HTTP web port 80 是經常開啟讓網頁可以瀏覽下載文件的管道，因此不易被防火牆偵測出來，所以隱蔽性極高。

一般使用者只要依照需要置入所需要的相關配置資訊以後，就可以產生具感染力的宙斯殭屍病毒程式。

2.2 惡意資料洩漏分析與檢測

[11]提出一種透過通訊流量來分析檢測的方法，並用宙斯殭屍網路病毒來驗證理論。宙斯殭屍網路雖然是以 HTTP 通訊協定來試圖隱藏殭屍網路通訊的訊息，但相較於一般正常使用的情況下，網路的流量仍會具有部分的差異性，這些差異就可以用來作為偵測的依據。

利用信息理論的夏農熵(Shannon's Entropy)來做特徵選取的計算，這可以對特定結果計算隨機變數的不確定性，不確定性越高，計算出來的 Entropy value 也會越高。作者將一般正常使用者電腦與感染宙斯殭屍網路病毒的電腦進行長時間的網路封包紀錄，使用夏農熵計算式去算出 Entropy value[12][13]。在使用不同的資料集去交叉比對找出誤判率較低的值來作為偵測的門檻值(threshold value)。

2.3 Openflow switch 分析偵測

在[14][15]中，透過 openflow switch 具有封包轉向的功能，來實作受害者的通知機制。整個系統架構中，除了控制封包導向的 NOX controller 與 openflow switch 外，還有一台替身的伺服器(substitute server)與一台網站伺服器(web server)。

1. 替身伺服器其功能主要作為一個偽造的 C&C Server，當中毒者向 C&C server 發送訊息，會由 openflow switch 導至這台替身伺服器，由其與中毒者電腦來回應。
2. 網站伺服器會將中毒者的網頁內容導向事先架好的中毒通知頁面，以便提醒中毒者電腦有異常行為，以便後續進行惡意病毒移除的動作。

2.4 動態延伸蜜罐誘捕系統

[16]中提出一種動態延伸雙向蜜罐誘捕系統(Dynamic Extensible Two-way Honeypot，簡稱 DEH)的架構，two-way 指的是提供進出蜜罐的雙向網路，相較於一般蜜罐怕被作為攻擊者對外攻擊的跳板，往往只提供連入蜜罐的對內網路，而由蜜罐對外的網路連線多半是禁止的，但這樣一來容易被識破且造成整體誘捕效率不佳，攻擊者當然不願在蜜罐中被記錄下攻擊手法與資訊，同時，這更是提供了攻擊者一個很好測試是否身處蜜罐系統中的方

法，因為只要針對由內對外的網路連線測試就能得知了。DEH 架構如圖 1 所示。

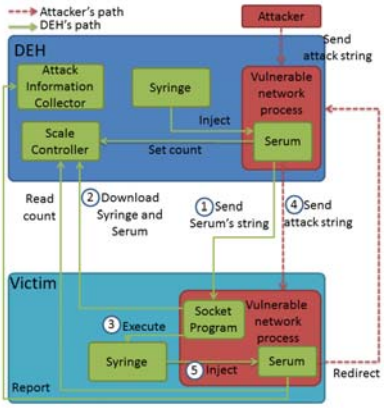

圖 1 DEH 系統結構流程圖

資料來源：[16]

3. 系統設計與架構

我們將提出的方法於虛擬機器中來進行實作，並且以宙斯病毒為測試對象，討論測試方法的可行性與實驗結果，我們期望中毒的受害者電腦可以透過我們提供的 Server 解毒，停止或降低受害者端的影響程度。

3.1 問題分析

我們利用 DEH 機制提供連上特殊的偽造 C&C 伺服器位置，由伺服器端來下指令給 Botclient 作為解毒的方法。

1. 解毒的行為從伺服器端來操作，這樣可更加快解毒的時程。
2. 若對於存在大規模需要解毒的情況下，從伺服器端解毒會是比較有效率的方式。

同時，我們會將 DEH 機制納入於系統設計中，DEH 是一個主動式注入通訊機制附帶於病毒惡意軟體跟隨著被執行的防禦方式，這正巧符合我們想提出主動式傳播殭屍網路的防禦功能，透過攻擊型態的區別，我們能避免一般的偵測機制常發生的問題：即便電腦多次完成清毒的行為，操作習慣不良的使用者往往容易在短時間內又成為殭屍網路受害者。

3.2 系統設計

系統主要包含四大部分，包括 DEH、Fake C&C Server、Switch 和 Victim PC。

DEH：

本系統設計以 DEH 作為一個基礎架構，由於 DEH 還提供動態延伸觀察範圍的機制，我們可藉此利用這項機制為來對受害者電腦對外網路進行控制，一旦感染的惡意程式嘗試與外部 C&C Server 作聯繫，DEH 便會自動注入 DLL 檔來修改建立 Socket 連線時所需的相關程式，並且更改封包傳送的目的端位置，如此一來，惡意程式便不會與真實的 C&C Server 有所聯繫。

Fake C&C Server：

Fake C&C Server 為本系統機制重要角色之一，因為它會接收惡意程式所送過來的封包訊息，並且可以對其下指令進行控制，受感染的電腦透過封包的導回作用，與此 Fake C&C Server 作聯繫，同時 Fake C&C Server 也會給予殭屍網路病毒的相關配置檔，進而修改原先殭屍網路設定值，方便後續解毒之用。

Switch：

我們將於 Switch 上裝設網路封包分析軟體 Wireshark。透過 Wireshark 去分析顯示出詳細的網路封包資料，目的在於我們希望可以得知 C&C Server 的 IP 位置，設定出 IP 黑名單作為封包導向的依據，黑名單中記錄的是來源端 IP，也就是哪些 PC 電腦曾經配置檔的請求。

Victim PC：

此為受害者電腦角色，它會被攻擊者入侵並安裝殭屍網路病毒。

透過流程圖來說明每個階段的運作機制，如圖 2 所示。

圖 2 主動式入侵防禦機制之系統運作流程

4. 實驗與討論

本章節為我們系統的實驗測試，將第三章所討論的架構採用 VMware 來進行環境的架設，所有 VM 環境皆為 Windows XP SP3，並以宙斯殭屍網路為例去實際測試提出方法的可行性。

4.1 實驗環境

在此實驗環境中我們安裝版本號: 1.2.7.19 的宙斯病毒，同時裝設 1.2.10.1 版本的 Control Panel 作為發佈命令的控制頁面。使用 XAMPP 工具程式架設網站，這是一個整合 Apache、PHP 和 MySQL 的工具套件，方便使用者輕易地建立網站。

4.2 黑名單的篩選機制

每個 Zbot 會依照配置檔內的 timer_config 設定值，定期的向 C&C server 發送 GET reques，我們可以利用宙斯殭屍網路的這項特性，一旦出現 GET config.bin 的訊息，我們將判定為是 Zbot 向 C&C Server 發送請求的訊息，因為在正常使用網路的情況下，並不會有這樣需求訊息出現。

4.3 DEH 系統修改

在原 DEH 設計中，Scale Collector 除了是一個 web server 提供相關的程式下載外，它是 DEH 控制觀察路徑長度的關鍵元件，每當有攻擊連線於 victim 端發送出前，Serum 會連至此處的相關網頁讀取 count 值作為參數，決定是否要修改 Inet_addr() 進行導回至 DEH。

在主動式攻擊防禦機制中，我們將修改導回的機制與導回的地點。Victim 端要發送連線之前，一樣會讀取相關網頁取值作為參數，而依照來源端的 IP 是否在 IP 黑名單中而得取不同的數值。1 表示來源端在 IP 黑名單中，故這個受害者電腦必定曾經發送出 GET config.bin 訊息而列為黑名單，它的連線請求都必須導回至 Fake C&C Server；0 則表示來源端 IP 正常，不需導向。

4.4 實驗測試與結果

4.4.1 實驗一：更新配置檔

測試是否可以利用偽造 C＆C Server 配送更改的 config 檔來達到停止 ZBot 的行為。我們在 Fake C&C Server 放入了偽造的配置檔案，希望透過 ZBot 來更新此檔並且接受 Fake C&C Server 的特定指令。首先，更改配置檔中的 timer_config 的設定值，如此一來，雖然 ZBot 成功於受害者電腦啟動，但短期間內並不會執行任何惡意指令，這樣可以視同為暫時終止 ZBot 的行為模式。對 ZBot 而言，更新後的配置檔內新的 C&C Server 就是 Fake C&C Server。

以實驗結果來說，我們可以透過偽造的 config 檔來達到停止 ZBot 的行為。

4.4.2 實驗二：更改登錄值

測試是否可以停止 ZBot 的執行。ZBot 感染後會修改的登錄值，如果可以將此登錄值更改成正常數值，讓每次受害者開啟電腦時，ZBot 相關程式不會自動被執行，那麼就可以確實終止了 ZBot 的惡意行為。因此，我們將透過偽造的 C&C Server 去下特定指令，讓修改的登錄值還原回來。「rexec」是個可以「下載指定檔案並且執行」的指令，我們

需要「rexec」這個指令來修改登錄值，將修改登錄值的指令寫入檔案中，讓 ZBot 去下載並且執行，不過「rexec」指令有其限制性，它無法執行會顯示於視窗的執行檔，也代表下載並執行的功能具有限制條件。我們決定使用執行批次檔的方式去符合這項條件，同時批次檔(.bat file)內部的指令亦是設定為強迫執行，以避免出現警告視窗。

Fake C&C Server 所下之 command：

rexec http://URL/xxx.bat

實際測試後，在移除這些登錄值並且重新開機，sdra64.exe 並不會自動將自身植入 winlogon.exe，而 Fake C&C Server 所下的 command 一直未等到 ZBot 來執行，所以我們可以透過移除 ZBot 所修改的登錄值來終止 ZBot 的行為。

4.4.3 實驗三：測試其他版本 ZBot

測試其他版本 ZBot，並驗證上述方法之可行性。我們將在與 ZBot 相同的環境下測試驗證版本號: 2.0.8.9 的宙斯殭屍網路病毒(以 nZBot 代稱之)，在實驗一的部分，透過配送 config 檔可達到停止 nZBot 的行為，實驗效果與預期功能相符。

但在實驗二的部分經過實驗卻發現由 C&C Server 所下的「下載指定檔案並且執行」的指令無法發揮作用，原因在於原本預期的機碼名稱並非 userinit 而是改成一個亂數字串，如此一來我們便無法使用實驗二的方法達到終止 ZBot 的行為。

為此我們提出了另一個方法來處理這個問題：透過 C&C Server 下達更換首頁的指令[19]，目的在於透過原先殭屍網路想要用來綁架受害者網頁的功能，將它轉作為我們通知受害者的方式之一，受害者的瀏覽器首頁會被轉到我們預先設定好的中毒通知警告頁面，一方面讓受害者知道自己的電腦已遭到危害，二方面在網頁上可提供相關的解毒訊息供受害者參考。

Fake C&C Server 所下之 command：

user_homepage_set http://URL/xxx.htm

如果在無法終止其他版本 ZBot 情形下，我們可以使用更改首頁位置達到通知受害者的功能，雖然沒辦法對 ZBot 作有效的終止行為，但即時的通知受害者可降低整體的受害程度。

5. 第五章 結論與未來工作

5.1 結論

我們提出的主動式攻擊防禦機制可以有效的停止宙斯殭屍網路的惡意行為，並有效的降低受害者電腦的受害情況。同時，我們提出的方式適用於主動式的攻擊型態，相較不受限於使用者的上網使用習慣，比多數的防禦方法更能有效避免受害者電腦在清除病毒之後又發生多次人為不良操作而中毒的情況。我們透過Server端嘗試對受害者電腦進行直接進行解毒的行為，一來可以提升解毒成功率，二來我們可以有效的大規模進行解毒。面對不同的殭屍網路病毒，此防禦機制可以透過C&C Server指令的下達，來停止殭屍網路的行為與提供中毒頁面的通知，相較於只有單一方面的偵測機制，我們的防禦機制更能防止殭屍網路的蔓延。

根據不同需求的殭屍網路會有不同樣的功能設計，因此，依照我們的系統設計限制，可歸納出本論文提出的防禦機制所適用的範圍條件：

可否偽造出C&C Server

在系統設計中，偽造的C&C Server是扮演下達偽造命令的角色，提供受害者電腦停止惡意行為的命令，如果無法偽造出C&C Server來下命令，那麼此防禦機制會因為無法修改受害者電腦內部被攻擊者更改的設定而無法進行防禦，所以偽造的C&C Server是關鍵因素之一。

殭屍網路命令功能的限制

在偽造命令的下達方面，本防禦機制會使用到「下載檔案並執行」這個動作，如果殭屍網路命令列表並沒有提供這項功能，這會無法讓受害者電腦去載入我們指定檔案的執行工作，進而影響防禦機制的運行。

5.2 未來工作

本論文中我們以HTTP的殭屍網路為例實際測試，但是殭屍網路的型態並非只有這一種，尤其是以P2P作為通訊協定的殭屍網路其拓樸架構為隨機式的，每個Botclient也可能扮演C&C Server的角色，這會加深防禦上的難度。在未來，防禦機制的自動化將是未來的重點項目之一，在黑名單的篩選以及偽造命令的下達方面，若能夠自動化的進行處理，將可提高此機制的效率，同時也可在最短時間內可防止惡意行為的發生。

參考文獻

[1] Abu Rajab, Moheeb, et al. "A multifaceted approach to understanding the botnet phenomenon", Proceedings of the 6th ACM SIGCOMM conference on Internet measurement. ACM, 2006, pp. 41-52

[2] Davis, Carlton R., et al. "Sybil attacks as a mitigation strategy against the Storm botnet", Malicious and Unwanted Software, 3rd International Conference on. IEEE, 2008, pp.32-40

[3] S. Stankovic and D. Simic. "Defense Strategies Against Modern Botnets", IJCSIS, June 2009, Vol. 2

[4] Zou, C.C. and Cunningham, R., "Honeypot-Aware Advanced Botnet Construction and Maintenance", Proceedings of the 2006 International Conference on Dependable Systems and Networks (DSN'06), pp.199-208

[5] Masud, M. M., et al. "Flow-based identification of botnet traffic by mining multiple log files", Distributed Framework and Applications, 2008 First International Conference on. IEEE, 2008, pp.200-206

[6] Livadas, C., et al. "Usilng machine learning technliques to identify botnet traffic" Local Computer Networks, Proceedings 2006 31st IEEE Conference, 2006, pp.967-974

[7] Gu, G., Perdisci, R., et al. "BotMiner: Clustering analysis of network traffic for protocol-and structure-independent botnet detection", 17th conference on Security symposium, 2008, pp. 139-154

[8] Jose Nazario and Holz, T., "As the Net Churns: Fast-Flux Botnet Observations", 2008 3rd International Conference on Malicious and Unwanted Software(MALWARE), pp. 24-31

[9] Binsalleeh, H., "On the Analysis of the Zeus Botnet Crimeware Toolkit", 2010 Eighth Annual International Conference on. IEEE, 2010, pp.31-38

[10] Falliere, Nicolas, and Eric Chien, "Zeus: King of the Bots.", Retrieved from Security Response Whitepapers Symantec Corp, 2009. [online] http://www. symantec. com/content/en/us/enterprise/media/security_response/whitepapers/zeus_king_of_bots. pdf

[11] A. Al-Bataineh and G. White, "Analysis and detection of malicious data exfiltration in web traffic", Malicious and Unwanted Software (MALWARE), 2012 7th International Conference, p26 - 31

[12] Lesne, A., "Shannon entropy: a rigorous mathematical notion at the crossroads between probability, information theory, dynamical systems and statistical physics." (2011).

[13] Pharwaha, Amar Partap Singh, and Baljit Singh, "Shannon and Non-Shannon measures of entropy for statistical texture feature extraction in digitized mammograms", Proc World Congress Eng Computer Sci. Vol. 2. 2009, pp. 1-6

[14] 彭士家, "Botnet Victim Detection and Notificationbased on Openflow Switch", 國立中央大學資訊工程所碩士論文 民國 99 年

[15] 黃勝獅, " Botnet Traffic Analysis and Dectection by Using OpenFlow Switch", 國立中央大學資訊工程所碩士論文 民國 100 年

[16] 趙亞略, "DEH: Dynamic Extensible Two-way Honeypot", 國立中央大學資訊工程所碩士論文 民國 101 年

[17] Raghava, N. S., Divya Sahgal, and Seema Chandna, "Classification of Botnet Detection Based on Botnet Architechture" Communication Systems and Network Technologies (CSNT), 2012 International Conference on. IEEE.

[18] 銳傑科技威脅通報[online] http://www.eranger.com.tw/virus.php?v_id=811

[19] A Brief Look at Zeus/Zbot 2.0[online] http://www.symantec.com/connect/blogs/brief-look-zeuszbot-20