Botnets: Mecanismos de Control y de propagación

María José Erquiaga
- Estudiante de Ing. Electrónica UTN FRM -
mariajoseerquiaga@gmail.com

Resumen. Este trabajo define al malware en general, y a las botnets en particular. Describiendo el ciclo de vida de las redes zombies, los BotnetMaster, que controlan estas redes, los mecanismos de propagación, y los mecanismos de control teniendo en cuenta las topologías de las botnets, este análisis es fundamental para la detección y mitigación de este tipo de amenazas. También se da a conocer los puertos más vulnerables que sufren este tipo de ataques.

Keywords: Botnet, BotnetMaster, Puertos, Mecanismos, Propagación, Control.

1 Introducción

El crecimiento acelerado de la informática trae como consecuencia el malware, o software malicioso. Con el desarrollo de la inteligencia artificial, esta se combina con el software dañino o malware, naciendo así las *botnets*, redes de pequeños robots (o *bots*) con el fin de atacar y tomar hosts para que formen parte de su red. Realizando este proceso de forma automática y autónoma, y creciendo a un ritmo vertiginoso.

Las *Botnets*, son llamadas también redes zombies, ya que los host se encuentran a merced de su *BotnetMaster*. Estas redes han logrado expandirse a lo largo de Internet afectando a miles y millones de usuarios, valiéndose de ingeniería social para "engañar" a sus víctimas. Un fenómeno que crece y difícilmente pueden atraparse a los culpables, ya que evolucionan constantemente para volverse casi indetectables.

2 Malware

2.1 Un poco de historia

Esta historia comienza con la era de informática, el avance y desarrollo de las computadoras y el software. En 1948 Von Neuman, es el precursor de la idea de autorreproducción de las máquinas, desarrolla un artículo: *"Theory and Organization of Complicated Automata"* (Teoría y organización de autómatas complejos). Luego Arthur Burks, en 1966 completa el artículo de Von Neuman, y publica el libro *"Theory of Self-Reproducing Automata"*(Teoría de auto-reproducción autómata)[10]

En 1959 en los laboratorios de *Bell Computer*, se crea un juego llamado *CoreWar* (Guerra de Núcleo), inspirado en la teoría de Von Neuman. Este es el precursor de los virus informáticos, el juego consistía en dos programas escrito en un lenguaje pseudo-ensamblador llamado *RedCode*. Estos luchaban entre sí, la victoria la obtenía el que lograra ocupar la memoria de su oponente [1]. *CoreWar* se mantuvo en secreto, hasta que en el año 1984, Ken Thompson, lo da a conocer e invita a la comunidad a

experimentar con estas criaturas. Además se forma la ICWS (International Core War Society) y se actualizan las reglas del juego con las que actualmente se sigue jugando en Internet.[1]

Creeper es el primer virus que logra infectar a una máquina IBM 360 en 1972, a través de una red ARPANET. Fue creado por Robert Thomas Morris, *Creeper* emitía un mensaje periódicamente que se mostraba por pantalla: *"I´m a creeper... catch me if you can!"* (Soy una enredadera, atrapame si puedes)". Para eliminarlo, se creó a *Reaper*, el primer antivirus.

Luego, John Walker en 1975, con el fin de distribuir un juego, da origen al primer Troyano. Este problema se solucionó creando una versión del juego que eliminaba las versiones anteriores del mismo. A fines de los `70 aparece el primer *worm* (gusano). Éste fue desarrollado por John Shoch y Jon Hupp, quienes crearon un programa con el fin de realizar tareas de mantenimiento y gestión durante la noche en el centro de investigación donde ellos trabajaban (Centro de Investigación Xerox de Palo Alto, California). Este programa auto-replicante se expandió por toda la red causando problemas y luego fue eliminado.

El crecimiento de los programas dañinos siguió aumentando en la década de los `80, hasta hoy. Se encuentra una gran variedad de software con el objetivo de dañar ordenadores y obtener información entre otros.

1.2 Clasificación y tipos de malware

Malware (contracción de malicious software), software malicioso también denominado software dañino. Es un código o porción de código insertado en una aplicación, programa o documento, que tiene como objetivo dañar equipos o apoderarse de ellos para luego utilizarlos con fines particulares, por lo general de forma maliciosa. En la figura 1.1 se muestra la taxonomía general del malware. Esta clasificación diferencia entre los que necesitan un programa anfitrión y los que no, es decir, en el primer caso, el código puede ser introducido en un documento, programa del sistema o aplicación. Podría decirse que el código malicioso está "embebido" en el código de un programa "anfitrión". Los independientes, en cambio, como la palabra lo dice, son programas puros *(denomino a programa puro a un código puramente infeccioso, no es una porción de código que se vale de una aplicación),* que pueden ejecutarse en cualquier momento en el sistema operativo.

Esta taxonomía es muy general, ya que hay muchos tipos de malware, muchas veces se utiliza el término "virus" para referirse al malware. Además pueden combinarse entre ellos, por ejemplo un bot por lo general contiene un gusano, o los gusanos o bots pueden también contener bombas lógicas. A continuación se describe cada uno de ellos brevemente.

- *Bombas lógicas,* están programadas para dañar los equipos en un momento determinado, se las puede programar para que "explote" en una fecha determinada o luego de que ocurra un evento particular, por ejemplo que el usuario ejecute determinado programa.

- *Caballos de Troya:* éstos simulan ser un programa útil para el usuario, como un programa de aplicación o juego, y cuando son ejecutados actúan de forma perjudicial para el equipo.

[1] www.corewar.info

- *Virus*: es un programa que afecta a otros programas modificándolos, realizando copias de sí mismo para continuar infectado equipos.

- *Gusanos:* se sirve de la red del usuario una vez que este es infectado para enviar copias de sí mismo a otros host, esto lo realiza de forma automática.

La diferencia entre los virus y los gusanos, es que los últimos tienen como finalidad causar daños a nivel de red, consumir ancho de banda por ejemplo, mientras que los virus causan daño a nivel físico del equipo.

Una característica del malware, es que tiene la capacidad de acceder de forma remota a un sistema, sin consentimiento o discernimiento del usuario. Deshabilita las medidas de seguridad (como firewall, o antivirus), perjudicando al sistema y a la información, y en algunos casos incluso dañando la parte física del computador.

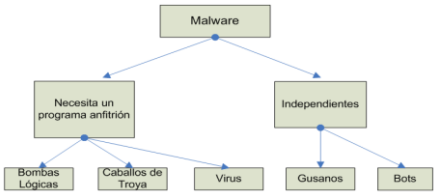

Imagen 1.1 *Taxonomía general del Malware*

3. Botnet

3.1 Definición

Para entender que son las *botnets*, hay que ir por partes. Primero: ¿Qué es un *bot*? *Bot* es una abreviación de *robot.* Son programas "inteligentes", se ejecutan automáticamente, no requieren de intervención humana y pueden realizar diversas funciones según se les ordene. Se utilizan para la actualización de programas, por ejemplo los antivirus utilizan bots para actualizarse.

Entonces, una *botnet* (también llamada "red zombie"), es una red de de "bots". Esta red, es controlada por el "*BotnetMaster*" (también llamado "pastor") de forma remota, y puede utilizar este ejército de "zombies" con diversos fines.

3.2 Ciclo de vida

Las botnets pueden ser creadas y mantenidas en cinco fases [2], éstas son:

1. *Fase de Infección inicial*: durante este periodo el atacante busca vulnerabilidades en algún host de una subred particular e infecta a la víctima a través de algún método determinado.
2. *Fase de inyección secundaria*: en el host infectado se ejecuta un script llamado *"Shell-code"*. Éste script incluye una imagen del bot ejecutable (o binario), y se instala a sí mismo. Esto puede hacerlo vía FTP, HTTP o P2P.
3. *Fase de conexión*: luego de establecer un canal *C&C*(Command&Control), el programa se conecta al servidor *C&C*. Una vez establecida esta fase, el host infectado pasa a ser parte de la red zombie.

4. *Fase C&C(Command&Control)*: en esta fase el *botnetmaster* utiliza el canal *C&C* ya establecido para controlar a su red zombie, así los bots reciben y ejecutan órdenes.
5. *Mantenimiento y actualización:* en esta fase, se les ordena a los bots a descargar actualizaciones. También muchas veces se les ordena migrar a otros servidores C&C, de esta forma, mantiene indetectable a su red.

En la figura 2.1 se puede apreciar el proceso explicado.

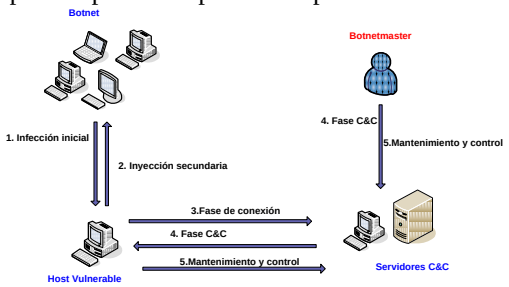

Imagen 2.1 *Ciclo de vida de una botnet*

3.3 BotnetMaster

BotnetMaster es quién dirige la red zombie, tiene bajo su control miles y hasta millones de host esperando sus órdenes. Por lo general son personas que les interesa vender o alquilar su red, es decir, esta actividad tiene un fin netamente monetario. Por otro lado hay muchos *hackers* con el sólo fin de divertirse, es muy fácil encontrar información sobre *Cómo armar su propia red zombie*. Esto por un lado, proporciona información sobre cómo funcionan estas redes, cómo se hacen, en qué lenguaje se basan, qué herramientas utilizan, pero por otro lado, se difunde y se enseña a realizar este tipo de actividades.

Se utilizan programas como VMware y compiladores como Microsoft Visual C++. Los bots se programan en lenguajes C/C++ o Perl.

A la hora de propagar sus bots, los BotnetMaster tienen en cuenta que deben ser ***FUD*** (Fully UnDetected: Totalmente indetectable), esto implica que no puedan ser detectados por ningún antivirus. Hay dos formas de convertirse en FUD, una es mediante la utilización de un *"cryptor",* que encripta los archivos, dejándolos indetectables. Otra forma es utilizando un *"binder"* también denominado *"Joiner"*, que combina dos archivos o programas en uno. Por ejemplo, Virus A+ SoftwareB= SoftwareC. Entonces cuando se corre el software C también se correrá Virus A. Puede entonces unir dos tipos de archivos, por ejemplo, un .exe con un .pdf o con un .jpg. Éstas herramientas pueden encontrarse fácilmente en la web, algunas de ellas son Diamond Binder, Advanced File Joiner v1.0, Billar Joiner 2.0, Jodedor 5X1, ShadeHacK Crypter, TYV Crypter 1.0, FUDSOnly Online Crypter V1 entre otros.

4. Mecanismos de Propagación

4.1 Definición

Se refiere a *mecanismos utilizados por bots en búsqueda de nuevos host*. ([2]) Estos mecanismos constan de escaneos en los puertos que se realizan para verificar si un puerto está abierto, cerrado, o protegido por firewall. Una vez que encuentran una vulnerabilidad en un puerto determinado, pueden comenzar a infectar. Otro mecanismo es mediante la utilización de información contenida en el equipo infectado.

4.2 Métodos de escaneos

Los métodos tradicionales consisten en:

- *Escaneo horizontal*: se escanea un puerto determinado variando un rango de direcciones IP. Es decir, se deja un puerto fijo, y se varía el número de direcciones IP.
- *Escaneo vertical*: se realizan sobre una IP específica, escaneando un rango de puertos.

Otro tipo de escaneo, es el "*Escaneo topológico*", que encuentra nuevos objetivos mediante la utilización de la información del ordenador de la víctima. Éste método es utilizando en gran medida por los gusanos, que obtienen una gran colección de direcciones de correo de los contactos de la víctima para conseguir nuevos host para infectar.

Estos escaneos se realizan de forma automática y constante, es por eso que los bots se propagan rápidamente.

Principales Víctimas

Los principales objetivos de los atacantes, son víctimas con las siguientes características [3]:

- *Gran ancho de banda*: les permite explotar los recursos de conectividad para realizar ataques DDoS, o bien para albergar archivos o software.
- *Disponibilidad*: host que se encuentren siempre encendidos, y dispuestos a recibir comandos.
- *Usuarios con poca capacidad de monitoreo*: usuarios con poca seguridad. Estos usuarios no actualizan su sistema operativo y/o aplicaciones, por lo que carecen de mecanismos de control como firewalls.
- *Ubicación:* buscan host que estén geográficamente lejanos de su propia localidad, con bajas de probabilidades de tener problemas legales o de ser localizados y atrapados.

5. Mecanismos de Control

5.1 Definición

[2] An Inside Look at Botnets. Paul Barford, Vinod Yegneswaran-Computer Sciences Department University of Wisconsin, Madison

"Los mecanismos de control de bots, hacen referencia al lenguaje de comandos y protocolos de control utilizados para manejar botnets de forma remota una vez que los sistemas han sido infectados".[3] Es decir, comprenden protocolos y comandos utilizados por los *BotnetMaster* para mantener y atribuir órdenes a sus bots. Esto lo realiza de forma remota, a través de diferentes métodos.

5.2 Tipos de Mecanismos de Control

5.2.1 Estructuras Centralizadas

En las estructuras centralizadas tienen una arquitectura árbol como se muestra en la figura 4.1, los bots se conectan al servidor C&C(Command & Control). El problema que tienen los atacantes con este tipo de mecanismo, es que una vez que se localiza el servidor C&C y se deshabilita, el *BotnetMaster* pierde su *botnet*. Por este motivo, los mecanismos de control están evolucionando a redes descentralizadas.

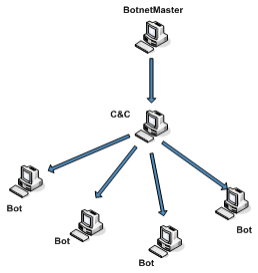

Figura 4.1. *Red C&C Centralizada*

Tipos de Redes Centralizadas

- *Push Style*:

Se comunican utilizando el protocolo IRC, se establece el canal de comunicación bajo este protocolo, y los bots esperan las órdenes de su *botnetmaster*. Este le da la orden al servidor C&C, quien le comunica la orden al bot, y el bot le responde. La figura 4.2 ilustra el mecanismo Push Style.

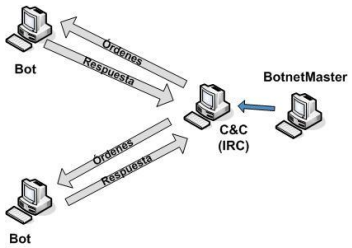

Figura 4.2. *C&C Centralizada: Push Style*

[3] An Inside Look at Botnets. Paul Barford, Vinod Yegneswaran- Computer Sciences Department University of Wisconsin, Madison

Se pueden enviar los mensajes de forma masiva a todos los bots, o enviar un mensaje con una orden específica a un bot en particular. El mensaje TOPIC se utiliza para enviar los mensajes masivos, los PRIVMSJ en cambio, están dirigidos a un bot en particular. La ventaja que le presenta este sistema al *Botnetmaster*, es que las órdenes se entregan en tiempo real [4].

Un ejemplo de éste tipo de comunicación puede verse en la imagen 4.3.

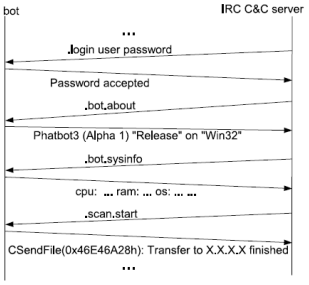

Imagen 4.3 *Comunicación C&C basada en IRC*

El primer paso que realiza el *botnetmaster* es ingresar el *password* para autentificarse, el bot acepta iniciar la comunicación y comienza el intercambio. Con el comando ".bot.about", se le pide identificarse. El bot le responde: *soy "Phatbot3 (Alpha 1)" "Realease" on "Win32"* (operando en Win32). Con el comando ".bot.sysinfo", le pide información sobre el sistema donde se encuentra alojado el bot, éste le responde con información sobre CPU, RAM, Sistema Operativo. Luego con el comando ".scan.start" comienza a realizar el escaneo. Hay varios comandos de este tipo en el que el atacante puede obtener información de sus bots, y darle órdenes.

- *Pull Style:*

El servidor C&C utiliza protocolo HTTP, el *BotnetMaster* carga las órdenes en el servidor C&C, que por lo general es un servidor HTTP que ha sido infectado. Los bots consultan cuál es la orden y el servidor les responde. Este tipo de funcionamiento puede apreciarse en la imagen 4.4.

Éste método no es del todo eficiente, ya que la comunicación no se produce en tiempo real. Sin embargo hay varios botnets trabajando de esta forma, que por ejemplo se conectan al servidor, accediendo a un URL para recibir órdenes y ejecutarlas [4].

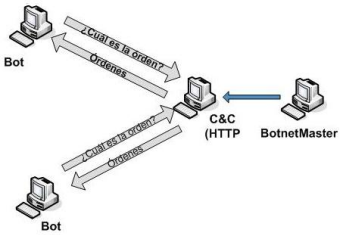

Figura 4.4. *C&C Centralizada: Pull Style*

- *Fast-Flux*

Es una técnica que implementa el uso de DNS, un dominio (FQD: Fully Qualified Name) puede tener múltiples números IP asignados, por ende puede realizar cambios rápidamente de direcciones IP para ese dominio.

Esta asignación cambia en función de un TTL (Time To Live) muy pequeño. Estos cambios de asignación puede estar dados por algoritmos como *round-robin*, que se implementa respondiendo solicitudes DNS A (A: "Address"). Alterna direcciones IP para el mismo dominio como puede verse en la imagen 4.4 [5].

```
;; ANSWER SECTION:
myspace.com.            3600    IN      A       216.178.38.116
myspace.com.            3600    IN      A       216.178.38.121
myspace.com.            3600    IN      A       216.178.38.104
```

Imagen 4.5. *Ejemplo de algoritmo DNS round-robin utilizando myspace.com*

5.2.2 Estructuras Descentralizadas

En este tipo de redes no hay un servidor central, por lo general se forman con redes par a par como se puede apreciar en la Imagen 4.6. Esto hace que sea más difícil encontrar al *BotnetMaster*. La utilización de las redes par a par o P2P (Peer-to-peer), es compartir información entre pares, no hay una relación cliente-servidor. Cada host puede actuar tanto como cliente o como servidor, solicitando o entregando información.

La ventaja en este tipo de redes es que hay una mejora en la conectividad y transferencia de información. Esto se logra mediante la administración del uso del ancho de banda [6]. Otra de las ventajas que presentan es que, si se cae un enlace, esto no deshabilita al resto de la red, sólo un par. Es decir que no se interrumpe las comunicaciones entre las otras redes P2P.

Las redes Par a Par, pueden o no tener una estructura. En el caso de las redes P2P estructuradas, cuentan con una tabla *Hash*. En la tabla *hash* se encuentra el camino más eficiente para llegar a otro punto en la red y localizar la información necesaria.

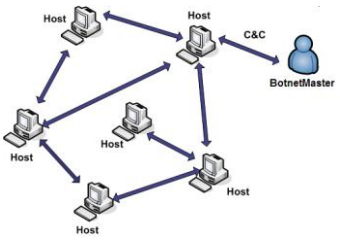

Imagen 4.6. *Red P2P descentralizada*

6. Vulnerabilidades en los Puertos

Los puertos son interfaces que se utilizan para comunicarse y trabajan a nivel de capa de transporte. Están numerados (utilizando 2 bytes) para identificar una aplicación o proceso determinado, y algunos están reservados en RFC 1700. Los puertos bien conocidos son los que se encuentran en el rango desde 1 a 1023. Éstos se utilizan para servicios estándar, por ejemplo para transferencia de archivos vía FTP se

utiliza el puerto 21. Los puertos desde 1024 al 65535 son los puertos registrados que no están estandarizados. [9]

En el caso de Windows, los puertos más afectados son [7]:

- Puerto 445/TCP (Microsoft-DS Service): utilizando para compartir archivos. Sistema Operativo: Windows 2000, XP, 2003.
- Puerto 139/TCP (NetBIOS Session Service): el uso es el mismo que el anterior. Sistema Operativo: Windows 9x, ME, y NT.
- Puerto 137/UDP (NetBIOS Name Service): este servicio provee información al host, ya sea su nombre de máquina, archivos que comparte, etcétera.
- Puerto 135/TCP: utilizado para implementar servicios RPC (Remote Procedure Call). Permite ejecutar un código de forma remota.

Según un experimento realizado por el proyecto HoneyPots [7]. El 80% del tráfico capturado se encontraba en éstos puertos. Sin embargo, éstos no son los únicos puertos afectados hay otros que presentan vulnerabilidades:

- 42 –Utilizado para Servicio WINS (Microsoft Windows Internet Name Service). Sirve para localizar host en la red, este servicio mantiene una tabla que relaciona direcciones IP y nombres NetBIOS.
- 80 – World Wide Web (vulnerabildades en los servidores de Internet)
- 903 - NetDevil Backdoor, le permite al hacker control remoto del host infectado.
- 1433 - ms-sql-s (Microsoft-SQL-Server). Gestiona base de datos relacionales SQL.
- 2745 - Puerta trasera del gusano Bagle, que envía correos de forma masiva.
- 3127 – Puerta trasera del gusano MyDoom(también conocido como W32.Novag.A), que envía correos de forma masiva con adjuntos que contienen archivos con extensiones de tipo .bat, .cmd, .exe, .pif, .scr o .zip.
- 3306 - MySQL UDF Weaknes. Utilizado para el sistema de gestión de Base de datos.
- 6129 - dameware (Administrador remoto Dameware) Permite administrar host infectados por Dameware de forma remota.

En el caso de Sistemas Operativos Linux, los puertos más afectados son los que utilizan servicios de acceso remoto, como telnet (TELecommunication NETwork, telecomunicación de red) o ssh (Secure Shell, intérprete de orden segura). Telnet se maneja desde terminal y utiliza el puerto 23 por lo general. El principal problema que se presenta con Telnet es que la información viaja sin cifrar, es decir como texto plano, por lo que fácilmente puede ser interceptada por cualquier sniffer. SSH, por otro lado utiliza el puerto 22, se implementa para acceder de forma remota a una pc, desde terminar, y también en entorno gráfico. La gran diferencia entre SSH y telnet, es que el primero utiliza cifrado de la información.

7. Conclusión

Las botnets son una combinación peligrosa de Inteligencia Artificial y malware, esto demuestra la tecnología avanza no solo en forma beneficiosa sino también de modo perjudicial y malicioso. Las *botnets* están en constante crecimiento, evolucionando permanentemente para volverse indetectables. Los mecanismos de control tienen a descentralizarse, lo que dificulta encontrar los responsables de los crímenes.

Detrás de todo esto hay un gran negocio, ya que se comercializan estas redes con diversos fines. El principal problema es la cantidad de información que hay en la web,

divulgando métodos y mecanismos de infección y propagación, proporcionando instrucciones para infectar y atacar. Es de vital importancia en análisis de estas redes, sus mecanismos de propagación y de control en particular para combatirlos. Conocer los puertos vulnerables, para lograr encontrar anomalías.

Referencias

[1] http://es.wikipedia.org/wiki/Core_War

[2] A survey of Botnet and Botnet Detection. 2009 Third International Conference on Emerging Security Information, Systems and Technologies (Maryam Feily, Alireza Shahrestani, Sureswaran Ramadass)

[3] BOTNET - A Network of Compromised Systems -Dr. Sanjeev Sofat,Prof. Divya Bansal Mayur Gupta - Department of Computer Science -Punjab Engineering College, Chandigarh.

[4] Botsniffer Detecting C&C Chanels in network traffic. Guofei Gu, Junjie Zhang, y Wenke Lee. School of Computer Science. Georgia Institute of Technology. Atlanta

[5] Measuring and Detecting Fast-Flux Service Networks. Thorsten Holz1 Christian Gorecki1 Konrad Rieck2 Felix C. Freiling

[6] http://es.wikipedia.org/wiki/Peer-to-peer

[7] The Honeynet Project and Research Alliance.Know Your Enemy, Tracking Botnets. http://www.honeynet.org/node/51

[8] Redes P2P y enrutamiento en capa de aplicación. Jairo A. Afanador J., Diego F. Ribero, Germán E. Ulloa. Politécnico Grancolombiano

[9] Redes Linux Con TCP/IP. Guía Avanzada. Pat Eyler. Prentice Hall. 2001

[10] Cronología de los virus informáticos. La historia del malware. Cristian Borghello, CISSP, Technical & Educational Manager de ESET para Latinoamerica. 2009

Bibliografía y Sitios Consultados

- IEEE Xplore (http://ieeexplore.ieee.org/Xplore)
- Science direct *(*www.sciencedirect.com)
- Tendencias 2011: las botnet y el malware dinámico. Laboratorio de ESET Latinoamérica. 22 de noviembre del 2010
- Discovery techniques for P2P botnets (David Dittrich: Applied Physics Laboratory, University of Washington, Sven Dietrich: Computer Science Department, Stevens Institute of Technology)
- Botnet Detection and Response. The Network is the Infection. David Dagon. Georgia Institute of Technology. College of Computing. OARC Workshop, 2005
- Botnets, redes organizadas para el crimen. Lic. Cristian Borghello,Technical & Educational de Eset para Latinoamérica. 2007
- Comunicación y redes de computadores. Stalling. 6ta edición. Prentice Hall
- Redes de Computadoras. Andrew S. Tanenbaum. 3ra edición. Pearson
- Fundamentos de seguridad de Redes. Aplicaciones y Estándares – William Stallings