Índice

Índice Figuras

1. Introducción

El título y objeto de este proyecto no concluye con una simple definición, sino que engloba todo un fenómeno.

Se trata de un término compuesto, que con la simple traducción al castellano (red-robot), no expresa mayor relevancia. Pero, ¿a qué tipo de robots se hace mención en una *botnet*? ¿Qué tareas automatiza este robot? ¿Qué interés especial supone que conformen una red?

Estas cuestiones serán objeto de estudio en este documento, pero para la situación del concepto de *botnet* y tomar conciencia de las motivaciones que lo promueven, se debería empezar por la situación de sus promotores o creadores.

¿Que ha sido de la imagen del *hacker*[1] bien intencionado que investigaba los defectos o vulnerabilidades de la red, con el objetivo de mejorar la misma a cambio de la fama y el reconocimiento? ¿Son estos *hackers* una raza en peligro de extinción?, o ¿quizá hayan evolucionado hacia unos objetivos más oscuros?

Algunos expertos en seguridad han encontrado en los últimos años un modelo de financiación basado en la venta de servicios, o mejor dicho, en la venta de armas. No todos se han dejado seducir por el dinero de origen ilícito, sino todo lo contrario, han iniciado una cruzada contra esta actividad.

Los ciberdelincuentes exhiben sus últimas y más avanzadas armas, llamadas *botnet*, expresamente enfocadas al ánimo de lucro por medio de acciones ilegales. Convirtiéndose estos *hackers* oscuros en cómplices o ejecutores de todo tipo de delitos.

[1] Los programadores informáticos suelen usar las palabras *hacking* y *hacker* para expresar admiración por el trabajo de un desarrollador de software calificado.

Roban datos, invaden la privacidad de los usuarios, inundan la red y los correos de contenidos no deseados, bloquean las comunicaciones de las empresas, y así podríamos continuar con una larga lista de delitos. Todo ello promovido por un mercado negro o mafia que financia a estos delincuentes.

La magnitud de los ataques *botnet* cada vez tiene un mayor protagonismo en lo ya denominado por algunos expertos en seguridad como "ciberguerra".

Conscientes de la importancia que están alcanzando los sistemas *botnet,* la principal motivación en la realización de este proyecto será profundizar en su estudio, y así desvelar las claves en este enfrentamiento.

La red está plagada de resumidas e incompletas definiciones del término *botnet,* cada una de ellas contiene su pequeña verdad y a la vez deja un nuevo frente abierto. Dando lugar a infinidad de preguntas sin respuesta.

El objetivo es conocer el uso de estas aplicaciones y su actividad, la evolución de estas armas refleja la lucha entre ciberdelincuentes y sus contrarios, las empresas y expertos dedicados a la seguridad informática.

Apoyado por un ejemplo práctico, este trabajo descubre la compleja composición de estas herramientas, sus métodos, sus efectos y su sigilosa pero efectiva fuerza. La puesta en marcha de una red botnet experimental, permite tomar el papel de uno de estos ciberdelincuentes al mando de estas herramientas.

2. Descripción

Este apartado, es una descripción de un conjunto de infinitas armas y recursos, todos ellos pueden conformar un sistema *botnet*. Para ello se parte de su secuela u origen, éste es el *bot*[2], al cual a lo largo de los siguientes subapartados se le irán añadiendo características y funcionalidades, tratando de componer el significado completo de *botnet*.

2.1. El robot

El término robot se refiere al *software* o programa informático, que una vez instalado en un ordenador, realizará las tareas para las cuales ha sido programado, con total autonomía respecto al usuario de esa máquina, que ni se percatará del funcionamiento de este *software*.

Los *bots* son muy usuales en sistemas Windows, sin ir más lejos, cualquier antivirus se actualiza automáticamente. El código que ejecuta el antivirus comprueba si existen actualizaciones pendientes e incluso las descarga de manera autónoma, éste es un ejemplo de *bot*. Este comportamiento lo ha configurado el usuario durante la instalación del antivirus y automatizará la tarea, por lo tanto, en principio siempre actúa bajo el control y la confianza del usuario del equipo.

En el caso del antivirus, el *bot* ha sido instalado y configurado por y para el usuario del equipo, pero no siempre es así, cada vez más, terceras personas son responsables de la introducción de *bots* en el sistema ajeno sin consentimiento alguno, se trata de *bots* maliciosos a veces cómplices, otras ejecutores, de muchos tipos de actividad ilegal.

Un robot informático que es programado para establecer un canal de comunicación con su creador o promotor dándole acceso remoto, define otro tipo de *software* malicioso,

[2] Bot o robot: aplicación informática que realiza tareas automatizadas.

conocido como “troyano” o “caballo de Troya”. Éste otorga a una tercera persona el control del equipo.

Sin duda tiene mucho en común con los actuales *bots*, el troyano se podría decir que es el precedente de *botnet*, al cual sus creadores le han otorgado nuevas armas y funciones para hacer más eficaz y rentable su intrusión.

Los fines de estos intrusos son variados, utilizan los equipos ajenos para el envió de correo no deseado, robo de información personal: información bancaria, contraseñas, códigos de seguridad..., o simplemente incordiar de todas las formas posibles.

Como vemos estos *bots* trabajan en multitud de sistemas y misiones distintas, ya que no hay dos víctimas exactamente iguales. Sistemas operativos, antivirus y niveles de seguridad distintos eran en el pasado grandes inconvenientes, pues cada víctima debía ser estudiada y analizada antes de la intrusión para que el ataque surgiera efecto. Ahora explotando el ataque desde dentro, la mejor característica de estos *bots* es que son adaptables. Se basan en módulos de código[3], diseñados para descargar otros módulos e incorporarlos al ataque, con el fin de explotar debilidades específicas que encuentran en su víctima.

El *bot* inicial explota las vulnerabilidades que encuentra para hacerse con el control de su destino. Digamos que es un primer módulo el que coloniza el equipo de la víctima y se pone en marcha. Tendrá cierta capacidad de análisis, por ejemplo: averiguar el antivirus instalado en el sistema. A partir de esta información el mismo robot será capaz de descargar un segundo módulo o código, que se encargará de blindar al *bot* inicial frente a ese antivirus en cuestión y así evitar ser detectado, o quizá detener el *firewall*[4] del equipo para establecer nuevas comunicaciones sin levantar sospechas. El tercer módulo descargado puede que se encargue de escanear nuevas vulnerabilidades. Y así procede el

[3] Código de programación modular: consiste en dividir un programa en módulos o subprogramas con el fin de hacerlo más legible, manejable y de sencilla actualización.

[4] Un cortafuegos (firewall en inglés) es una parte de un sistema o una red que está diseñado para bloquear el acceso no autorizado.

avance del *bot* a esclavizar hasta el último reducto del sistema, teniendo así la capacidad de aprovechar al máximo las vulnerabilidades de cada víctima.

Con una sola infección la máquina podría estar enviando *spam*[5], capturando y enviando los datos bancarios del usuario y participando en ataques de denegación de servicio a alguna página web corporativa.

2.2. Infección

Insertar un primer módulo o *bot* que abra las puertas de la intrusión no es tan sencillo, por el momento el sistema solo obedece al usuario, y éste es el único capaz de ejecutar código. Por lo tanto, el objetivo será lograr que el usuario ejecute este primer módulo sin percatarse de los acontecimientos.

Para lograr la infección existen infinidad de métodos, sobretodo usando la ingeniería social, los más efectivos y conocidos se basan en camuflar el código en un archivo potencialmente deseable por el usuario, y ponerlo a su alcance, ya sea colgándolo en la red, enviándolo por correo, etc.

De esta forma cuando el usuario descargue este archivo al ordenador y lo abra, el código se ejecutará en segundo plano para no alertar al usuario de lo que está sucediendo. Una vez instalado este primer módulo el control de la máquina deja de ser competencia exclusiva del usuario para pasar a someterse al intruso.

Cuando nos referimos a archivos deseables, en caso de ser una víctima conocida, sería sencillo enviar vía programa de mensajería instantánea el archivo infectado, pero cuando se dirige la infección a víctimas desconocidas, el intruso se fija en los archivos con más

[5] *Spam*, correo basura o sms basura a los mensajes no solicitados, habitualmente de tipo publicitario, enviados en grandes cantidades (incluso masivas) que perjudican de alguna o varias maneras al receptor. La acción de enviar dichos mensajes se denomina spamming.

movimiento en la red: videos, música, fotos, *software*, etc. Además la prosperidad de las redes *P2P*[6] contribuye en gran medida a la sencilla y rápida distribución de estos archivos y en consecuencia de los *bots* camuflados que viajan con ellos. Sobre todo en contenidos multimedia encontraremos la red de distribución más grande de estos *bots*.

Existen otros métodos como la infección mediante la visita a una página web. En este caso el usuario decide visitar la web, o quizá esta web es un *pop-up*[7] que se abre automáticamente, o un enlace dentro de un correo basura,... En estos casos el usuario no ejecutará el código, sino que aprovechando las vulnerabilidades del navegador, el código malicioso o *bot* inicial será ejecutado automáticamente al cargar la página web junto con el código *HTML*[8], abriendo así las puertas al intruso.

El intruso se toma todas estas molestias para alcanzar el máximo número de víctimas, claro está que hay muchos usuarios que aun ejecutan cualquier archivo que caiga en sus manos, sea cual sea la fuente de origen. Pero por suerte éstos son cada vez menos.

Además el camuflaje no solo debe engañar al usuario, sino también incluye el blindaje de la imagen o archivo frente a la posible detección por parte de cualquier antivirus, ya que éstos actualmente analizan cualquier archivo recién llegado al sistema.

---

[6] Una red peer-to-peer (*P2P*) o red de pares, es una red de computadoras en la que todos los nodos se comportan como iguales entre si. *P2P* es un protocolo muy extendido para la compartición de archivos entre usuarios.

[7] El término denomina a las ventanas que emergen automáticamente (generalmente sin que el usuario lo solicite) mientras se accede a ciertas páginas web.

[8] HTML es el lenguaje de programación predominante para la construcción de páginas web

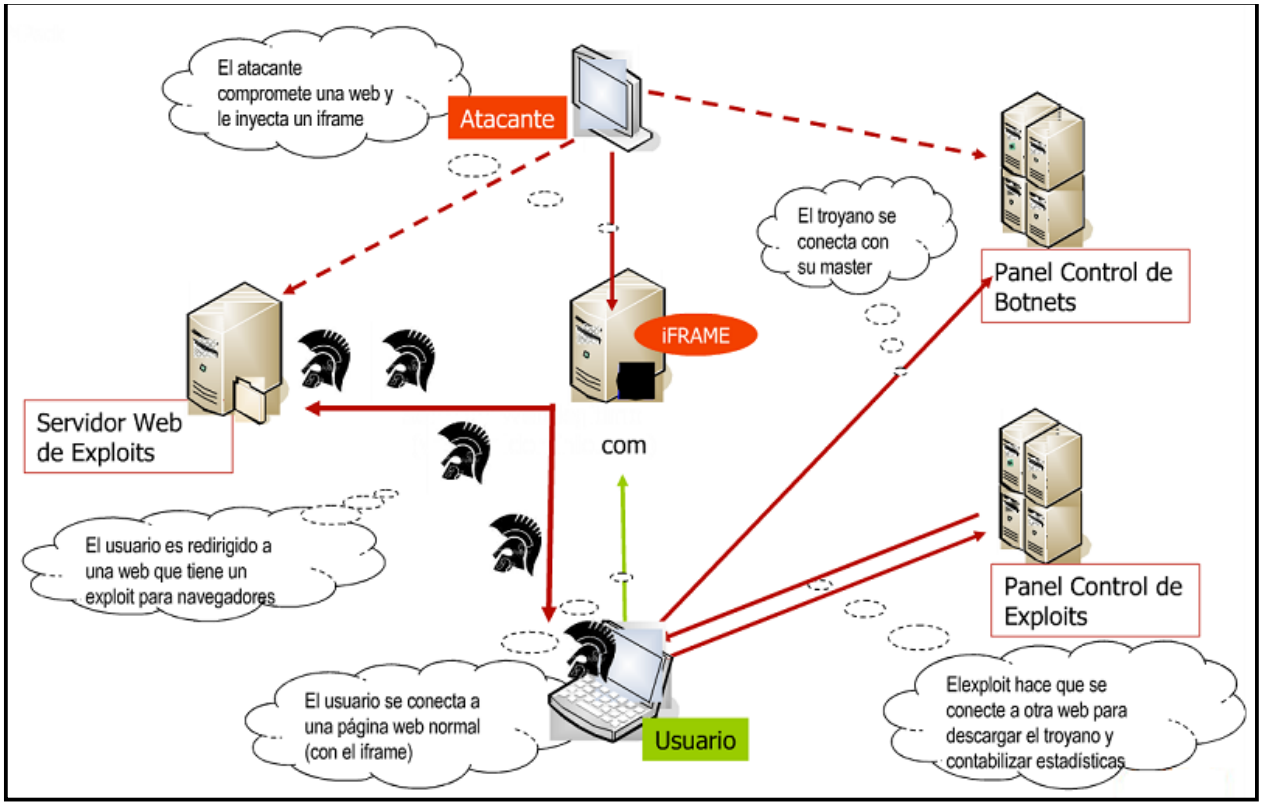

Figura 1: Infección mediante exploit en navegador web.

2.3. La red

¿Qué es mejor para un intruso que tener un equipo ajeno a su merced?
La respuesta es muy sencilla, mejor que un equipo, son miles de equipos zombi[9] dispuestos a cumplir sus deseos. Sobre todo para llevar a cabo ataques de denegación de servicio, que basan su poder en el número de máquinas que dirigen hacia el objetivo.

La distinción de *botnet* respecto del concepto de troyano, sería su enfoque a controlar el máximo número de ordenadores con un mismo robot adaptable o modular, mientras en el pasado el troyano iba dirigido a una sola víctima previamente estudiada.

[9] Se entiende por equipo zombi aquel que forma parte de una red *botnet*.

Esto es posible mediante el uso de sistemas de conexión cliente/servidor[10] ya en funcionamiento, el éxito de las redes *botnet* se basa en utilizar como plataforma de comunicación sistemas como servidores de chat, en este caso *IRC*[11], o redes *P2P*, entre otros. Donde además de simplificar los requerimientos del controlador de la *botnet*, se encuentra bien camuflada.

La *botnet* aprovechará así la arquitectura de redes preestablecidas para su funcionamiento, y con ella obtendrá todas sus ventajas. Como son la compartición de recursos, la sincronización de operaciones, etc.

2.4. La distribución

En el apartado de infección, se hacía mención al hecho de distribuir los *bots* para la infección, mediante el camuflaje de los mismos en archivos o contenidos potencialmente deseables en la red. Este método es quizá el más sencillo y extendido, aunque no es el único.

En el mercado donde cotizan las redes *botnet*, se las valora por la cantidad de ordenadores zombi que controlan y el beneficio que son capaces de aportar. El número de equipos infectados es proporcional al daño que puede causar una *botnet*. Es por eso que sus creadores invierten gran parte de sus esfuerzos por no decir todos, en el estudio de nuevos métodos que faciliten el crecimiento de estas redes.

El crecimiento puede basarse en inundar la red de enlaces a sus archivos infectados. O por el contrario y más inteligentemente, a incluir mecanismos de propagación en los mismos

[10] Esta arquitectura consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta.

[11] IRC (Internet Relay Chat) es un protocolo de comunicación en tiempo real basado en texto, que permite debates entre dos o más personas.

zombis. Si todo zombi tiene por misión infectar a sus vecinos, el crecimiento hasta ahora gradual de la red podría convertirse en exponencial.

Por ejemplo, el *botmaster*[12] implementa un nuevo módulo, por el cual en todos los equipos zombi se enviará un mensaje a través de programas de mensajería instantánea, en este ejemplo concreto "MSN Messenger", a todos los contactos del usuario con una foto, haciéndose pasar por el usuario del equipo. Hecho esto, todos los contactos del usuario infectado, al abrir esta imagen serán infectados también, y a su vez estos iniciarán con sus contactos el mismo proceso, iniciando una cadena de infecciones con un enorme potencial de crecimiento.

Al igual que en este caso "MSN" es la plataforma de infección, podría serlo cualquier software de comunicaciones, en este caso se eligió "MSN Messenger" por ser el más utilizado mundialmente y por ello tener un mayor alcance, pero estos métodos se han detectado en casi todos los programas de mensajería instantánea, en mensajes de correo e incluso recientemente en redes sociales como "Facebook" o "Twiter".

Este recurso para multiplicarse requiere de la ayuda del usuario, ya sea perteneciendo y conectándose a redes sociales, como con la compartición de archivos. Aunque existe una vertiente en la que el *bot* es capaz de replicarse con un proceso sin necesidad de la intervención de factores externos. El *bot* está dotado de un mecanismo de escaneo de red, mediante un algoritmo genera posibles *IPs* cercanas a la máquina donde reside, estas *IPs* son escaneadas y en caso de recibir respuesta desde una máquina cercana, utiliza vulnerabilidades en recursos compartidos y servicios de red para realizar la infección.

En la Figura 2 se puede observar un ejemplo de cómo el *bot* intenta aprovechar vulnerabilidades por acceso remoto en el servicio LSASS[13] sobre el puerto *139/TCP*[14] y *RPC*[15]*-DCOM*[16] sobre el puerto *445/TCP*[17].

---

[12] *Botmaster:* Dueño de una red *botnet*, lleva a cabo el comando y control de la misma.

[13] El proceso LSASS (Local Security Authority Subsystem), controla varias tareas de seguridad consideradas críticas, incluyendo control de acceso y políticas de dominios.

Targets	*Port*	*Month*	*Duration*
192.168.0.*, 192.168.1.*, 192.168.100.*, Random	445/tcp	May 2006	38 min
192.168.0.*, 192.168.1.*, 192.168.100.*	445/tcp	Jun 2006	51 min
192.168.0.*, 192.168.1.*, 10.1.*.*	445/tcp	Jun 2006	2.25 days
192.168.0.*	445/tcp	Jul 2006	6 min
Local net	445/tcp	Sep 2006	11 min
Local net	445/tcp	Sep 2006	2 h 8 min
Local net	139/tcp	Sep 2006	13 min
Local net	445/tcp	Sep 2006	12 min
Local net	139/tcp	Sep 2006	2 h 44 min
Random	139/tcp	Oct 2006	1 h 3 min
Random	445/tcp	Oct 2006	24 min
Random	445/tcp	Oct 2006	34 h 24 min
Random	139/tcp	Oct 2006	2.33 days
Random	139/tcp	Oct 2006	6.75 days
Random	139/tcp	Oct 2006	3.25 days
Random	139/tcp	Oct 2006	19 h 8 min
Random	139/tcp	Oct 2006	23 h 33 min
Random	139/tcp	Nov 2006	2 h 10 min
Random	139/tcp	Nov 2006	8 h 7 min
Random	139/tcp	Dec 2006	7 h 6 min
Random	139/tcp	Dec 2006	26 min
Random	139/tcp	June 2007	54 min

Figura 2: Mecanismo automatizado de infección.

---

[14] TCP (*Transmission-Control-Protocol*, en español Protocolo de Control de Transmisión) es uno de los protocolos fundamentales en Internet.

[15] Llamada de Procedimiento Remoto, es un protocolo utilizado por Windows, que proporciona un mecanismo de comunicación entre procesos internos, y que permite que un programa ejecutándose en una computadora pueda acceder a los servicios de otra, de manera transparente para el usuario.

[16] Modelo de Objeto Componente Distribuido) es un protocolo que nos muestra un conjunto de interfaces que permiten a los clientes y servidores comunicarse entre sí.

[17] Un atacante que tenga éxito en aprovecharse de estas vulnerabilidades, podría ejecutar un código capaz de tener todos los privilegios del sistema local de un sistema afectado.

3. Command & Control

Al extenderse el uso de *software* antivirus y *firewall* en los hogares ya no era efectivo el uso de conexiones cliente/servidor de los antiguos troyanos para la comunicación del bot con su dueño. Eran efectos del contraataque por parte de los *hackers* honestos, profesionales de la seguridad informática al servicio de las compañías de antivirus.

El trabajo de las empresas desarrolladoras de antivirus obligó a perfeccionar los bots o troyanos creados por los ciberdelincuentes, sobretodo sus mecanismos de camuflaje durante la infección. Teniendo que modificar sus armas para que el antivirus no las detectara.

Con este objetivo, minimizar la posibilidad de detección, los mecanismos de comunicación entre los *bots* y su dueño han tenido que evolucionar, hallando cobijo y liberándose de toda sospecha, entre las conexiones de servicios de uso normal en el sistema, evitando así la detección por establecimiento de conexiones nuevas. El uso de redes de servicios ya existentes para sus comunicaciones, como introducía el apartado 2.3. Ha mantenido su evolución, utilizando como plataforma de comunicaciones redes ya establecidas mediante protocolos[18] 100% fiables para cualquier cortafuegos, como: *HTTP*[19], *IRC o P2P*.

Otra de las marcas en la evolución de los sistemas de *C&C*[20] hacia el uso de redes y servidores preestablecidos, viene dada por el gran número de máquinas controladas en una red *botnet*. Con la infección de miles de máquinas, los recursos del *botmaster* soportando tantas conexiones simultáneas se agotarían, la máquina del botmaster finalmente se

[18] Un protocolo es un conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red.

[19] El protocolo de transferencia de hipertexto (HTTP, *HyperText Transfer Protocol*) es el protocolo usado en cada transacción de la Web.

[20] *Command&Control:* Referido al commando y control de una red botnet o troyano.

bloquearía como si de un ataque *DDOS*[21] se tratara. Por eso, esta carga en los recursos, es soportada actualmente por los servidores de la red ocupada y no por el equipo del *botmaster*. Gracias al uso como plataformas de comunicación de estos servidores y protocolos ya en funcionamiento (*IRC, P2P*,…), el ancho de banda y recursos del *botmaster,* no tienen por qué ser proporcionales al tamaño de la red controlada.

Poniendo como ejemplo un servidor *IRC,* se ve claramente los escasos recursos necesarios para el control de una *botnet*. En un servidor *IRC* normalmente se ofrece un servicio de chat, el intruso crea un canal protegido con contraseña dentro del servidor, al cual solo tienen acceso sus *bots* y él mismo como administrador, y es mediante ese chat que comunicará los comandos a ejecutar a los equipos zombi.

3.1. Arquitectura C&C centralizado

El servidor de *IRC* es el que soportará las miles de conexiones y no la máquina del intruso, y además estas comunicaciones serán bajo un protocolo confiable por defecto en cualquier máquina defendida con antivirus y *firewall*.

El mecanismo de comando y control basado en *IRC* es de carácter centralizado, muy efectivo en cuanto a sincronización de los ataques, aunque también supone un punto débil, ya que detectando y deshabilitando el servidor de *IRC* se está desarticulando por completo la *botnet*. Es por eso que en su trayectoria de evolución, el nuevo enfoque de *botnet* migra hacia la descentralización de estos sistemas de comando y control, haciendo de la *botnet* una red más robusta.

[21] Los ataques *DDOS* dejan incomunicado el equipo informático víctima, mediante el envió de gran cantidad de solicitudes hasta que ésta sea incapaz de procesarlas.

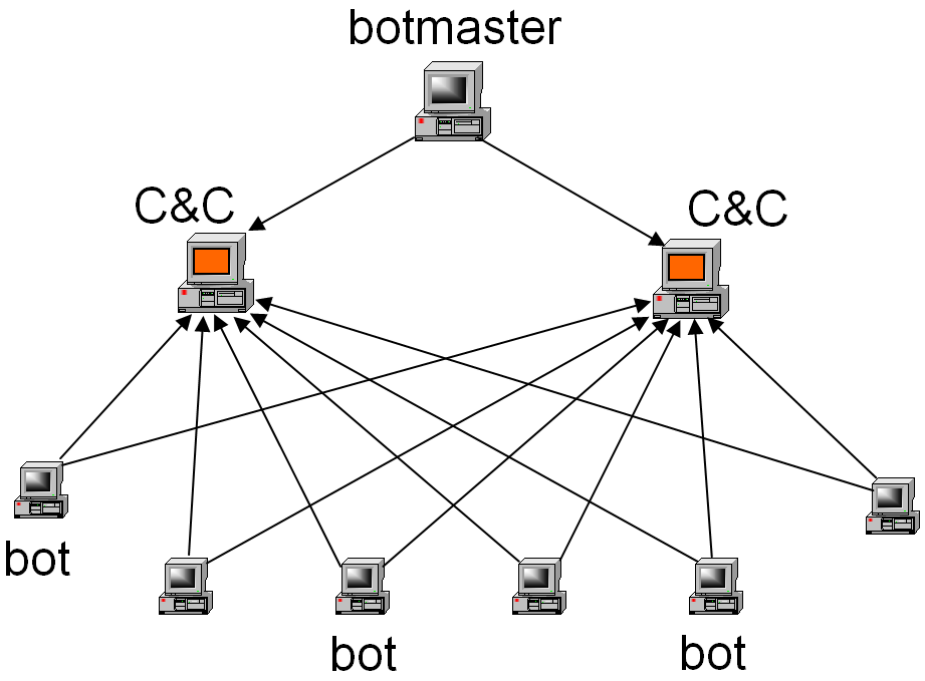

Figura 3: Estructura C&C centralizado.

Como modelo de red centralizado, una red *botnet* basada en *IRC* mantiene una comunicación directa con su *botmaster* a la espera de sus órdenes.

3.1.1. Push style

El método *push style* se identifica principalmente en *botnets* basadas en *IRC*, son las más comunes bajo esta arquitectura. Métodos centralizados de *C&C* en los cuales los comandos son enviados a los zombis.

Conocer el funcionamiento de las *botnets IRC* situará el concepto de *C&C* centralizado de forma clara.

Analizando el proceso de comunicación en *IRC botnet*, los *bots* se conectan a una canal mediante *Internet Relay Chat* y permanecen a la espera de la conversación únicamente del *botmaster*, el cual mediante ese mismo canal enviará los comandos ya sea en mensajes

masivos a todos los *bots* del tipo *TOPIC*[22] o vía mensaje privado para el control de las máquinas una a una(*IRC PRIVMSG*[23]), los *bots* entienden inmediatamente las tareas, las ejecutan y contestan en el mismo canal con el estado del proceso y el resultado una vez finalizada la tarea.

Para la entrada al canal, tanto el *botmaster* como sus zombis se identifican con 3 claves, una que les da entrada al servidor *IRC*, otra para conectarse al canal de chat y otra para desencriptar el cifrado del mensaje y entender las órdenes recibidas.

El uso de *IRC* como plataforma de comando y control está lleno de virtudes entorno al manejo y posibilidades de estas redes zombis, ya que la inmediata recepción de las órdenes permite al atacante, tanto sincronizar ataques de denegación de servicio más efectivos, como el control en todo momento del tamaño de la *botnet*, estando controlada en un solo canal.

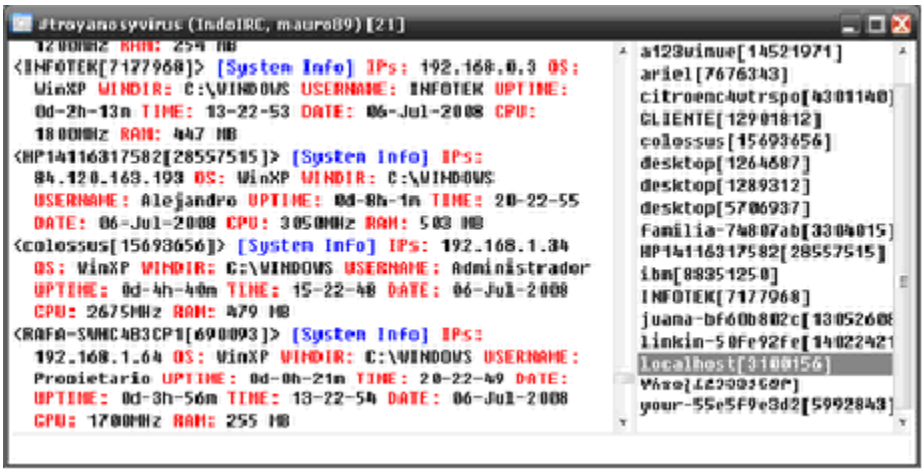

Figura 4: Comando bot.info en IRC botnet, zombis envían información del sistema.

Pero se ha comprobado que el beneficio de *IRC* como sistema de *C&C*, también ha llegado a ser su perdición, pues la *botnet* corre sobre un protocolo conocido y común a todos sus usuarios. Con este conocimiento previo se facilita su análisis, de manera que mediante la captura de tráfico en los servidores de *IRC* es posible detectar fácilmente la actividad

[22] Mensaje *TOPIC* es el que va dirigido a todos los usuarios conectados al canal de *IRC*.

[23] *PRIVMSG* mensaje que va dirigido a un solo usuario del canal *IRC*.

zombi, identificar máquinas infectadas o incluso desmantelar un canal de *C&C*, canal que representa el único punto de enlace entre *botmaster* y sus zombis, esto conlleva el fin de la *botnet*. Deduciendo que su carácter centralizado se convierte en su mayor debilidad.

Además bajo el código *IRC*, cualquier usuario infectado puede mediante el análisis de tráfico o el estudio del código del *bot*, averiguar las direcciones *IP* de todos los servidores *C&C* de la *botnet* a la que pertenece, conocer las claves de autentificación e incluso llegar a suplantar al *botmaster*, haciéndose con el control de la red zombi.

Con el tiempo, el uso de *botnet* basado en sistemas *IRC* se vio al descubierto, la aparición de aplicaciones para detectar y desarticular este tipo de *botnet* basándose en los patrones de tráfico disipó su uso.

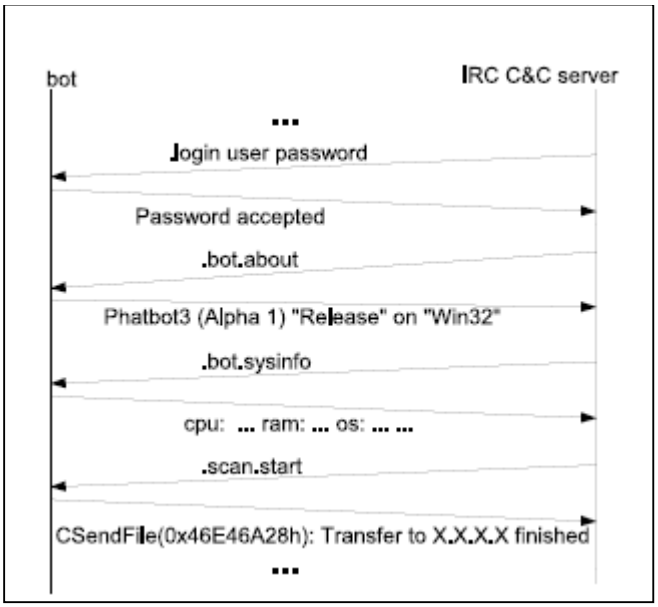

Figura 5: Comunicaciones de comando y control en botnet basado en IRC.

Una de las principales características del protocolo IRC es el uso de puertos[24] específicos, como son del 6665 al 6669. Sabiendo esto, cualquier firewall puede bloquear estos puertos

24

sin afectar a ninguna otra aplicación del equipo y deshabilitando con efectividad las comunicaciones de un posible *bot.*

Las aplicaciones antivirus ganaban la batalla por el momento. Hasta la llegada de *botnet* basado en *HTTP*.

Con este motivo surgieron principalmente los primeros *botnet* basados en HTTP, los cuales utilizaban el puerto 80 para sus comunicaciones, camuflando su tráfico entre el de cualquier navegador.

3.1.2. *Pull style*

Los servidores *IRC* se ven sustituidos por servidores web, y el sistema migra a una metodología basada en *pull style*, donde los zombis son los encargados de descargarse los comandos previamente colgados por el *botmaster* en servidores web comprometidos (Véase Figura 6). Esto evita que como pasaba en el caso de *IRC*, cualquier máquina infectada pudiera conocer la *IP* y la ubicación del resto de máquinas conectadas en aquel momento, ocultando también el tamaño de la *botnet*.

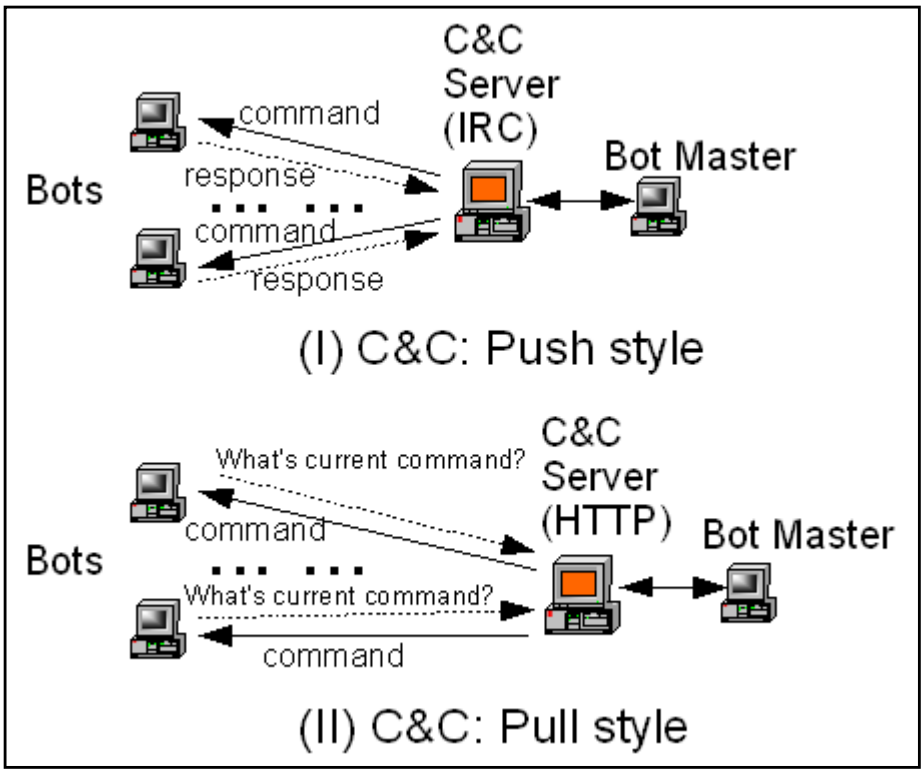

Figura 6: Métodos distribución de órdenes.

Pero trabajando con *HTTP* los *botmaster* conservan el peor de sus defectos, el que puede robarles por completo el control de su red zombi, el servidor C&C centralizado. Todo *botmaster* busca el anonimato y con un control centralizado está muy cerca de perderlo.

Todo *bot* se conecta a uno o varios servidores para descargar el código a ejecutar, resulta muy sencillo identificarlos, y siendo deshabilitados se perdería todo control sobre la red *botnet*.

Llegado este punto, con el sistema *pull* para la distribución de comandos de control, identificar el tamaño y miembros de la red zombi se convierte en una tarea de gran

dificultad para los posibles atacantes. La siguiente meta para los diseñadores de *botnet*, fue la de ocultar tras *Proxy*[25] los servidores de *C&C*.

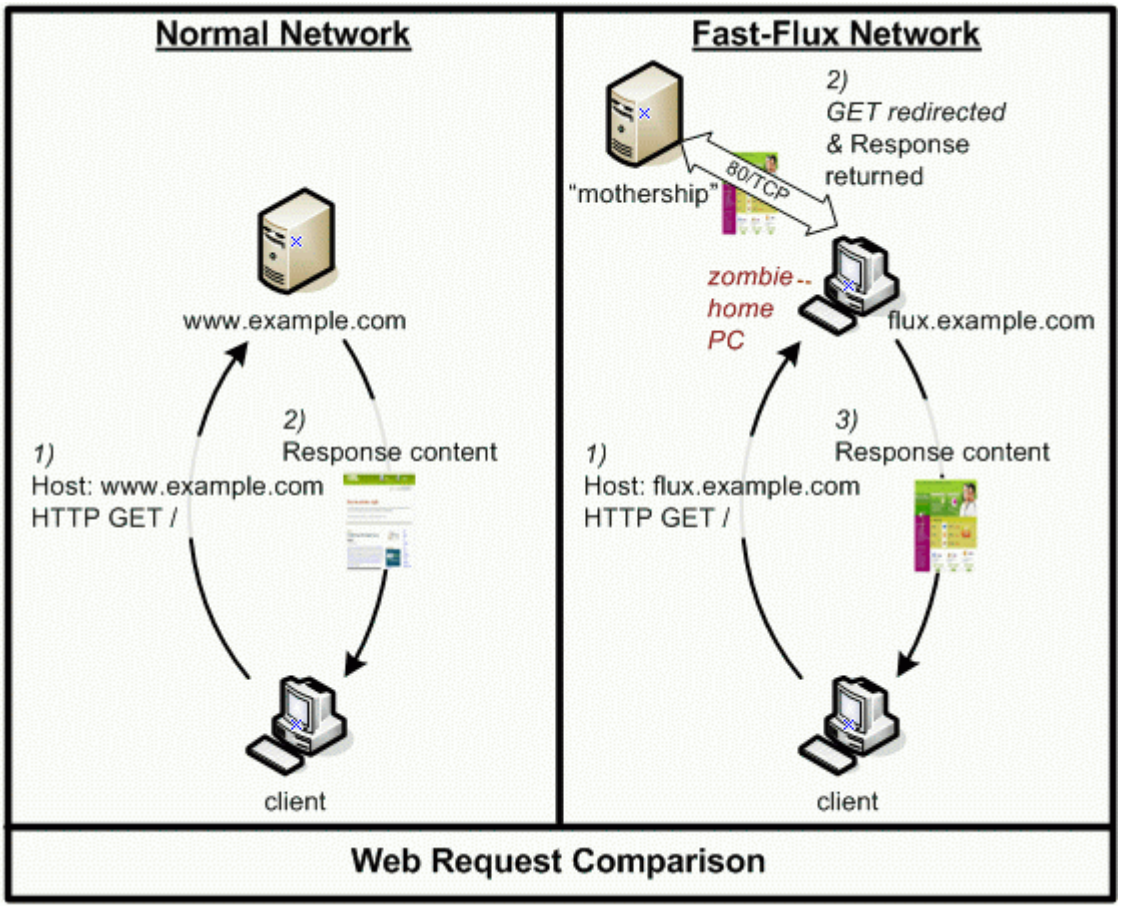

Figura 7: Uso de fast-flux en redes botnet.

3.1.3. *Fast-flux* (*single-flux, double-flux*)

El uso de un *Proxy* no era por si solo la solución a sus problemas, fue el uso de *DNS* lo que hizo ganar en robustez al sistema. Así se introdujo el uso de *Fast-flux* (Figura 9) en la arquitectura *botnet*.

[25] Un proxy permite a otros equipos conectarse a una red de forma indirecta a través de él. Cuando un equipo de la red desea acceder a una información o recurso, es realmente el proxy quien realiza la comunicación y a continuación traslada el resultado al equipo inicial.

Fast-flux es una técnica para el uso de miembros de la misma *botnet* como *Proxy* en las comunicaciones entre *bots* y servidores de *C&C*, con el fin de ocultarlos y así no sean tan vulnerables en un sistema de control centralizado.

Interponiendo así algunos obstáculos en el camino de los atacantes que intenten averiguar el origen real de los comandos de control. *Fast-flux* se encarga de que dichos obstáculos varíen constantemente evitando el rastreo a través de los mismos.

Fast-flux mantiene un ciclo de constantes cambios en la asignación de nombre de dominio[26], de tal forma que el *bot* se conecta a múltiples direcciones *IP* que van intercambiándose bajo un *TTL*[27] muy corto, estas direcciones pertenecen a otros *bots*, éstos actuaran como repetidores, enviando el contenido al punto de control real, conocido como *mothership*.

Este método se conoce como *single-flux*, y mediante su variante *double-flux* (Figura 10) no tan solo se modifica el registro A del DNS[28] para cambiar las IP a las que apunta, sino también los registros NS, cambiando así los servidores autorizados por uno que forme parte de la misma *botnet*. Logrando que la consulta DNS sea respondida por un miembro de la *botnet*.

---

[26] El propósito principal de los nombres de dominio en internet y del sistema de nombres de dominio (DNS), es traducir las direcciones IP de cada modo activo en la red, a términos memorizables y fáciles de encontrar. Esta abstracción hace posible que cualquier servicio (de red) pueda moverse de un lugar geográfico a otro en la red internet, aun cuando el cambio implique que tendrá una dirección IP diferente.

[27] TTL: *Time to live* o tiempo de vida.

[28] Registros *DNS:* A = *Address* – (Dirección) Este registro se usa para traducir nombres de hosts a direcciones IPv4. NS = *Name Server* – (Servidor de Nombres) Define la asociación que existe entre un nombre de dominio y los servidores de nombres que almacenan la información de dicho dominio.

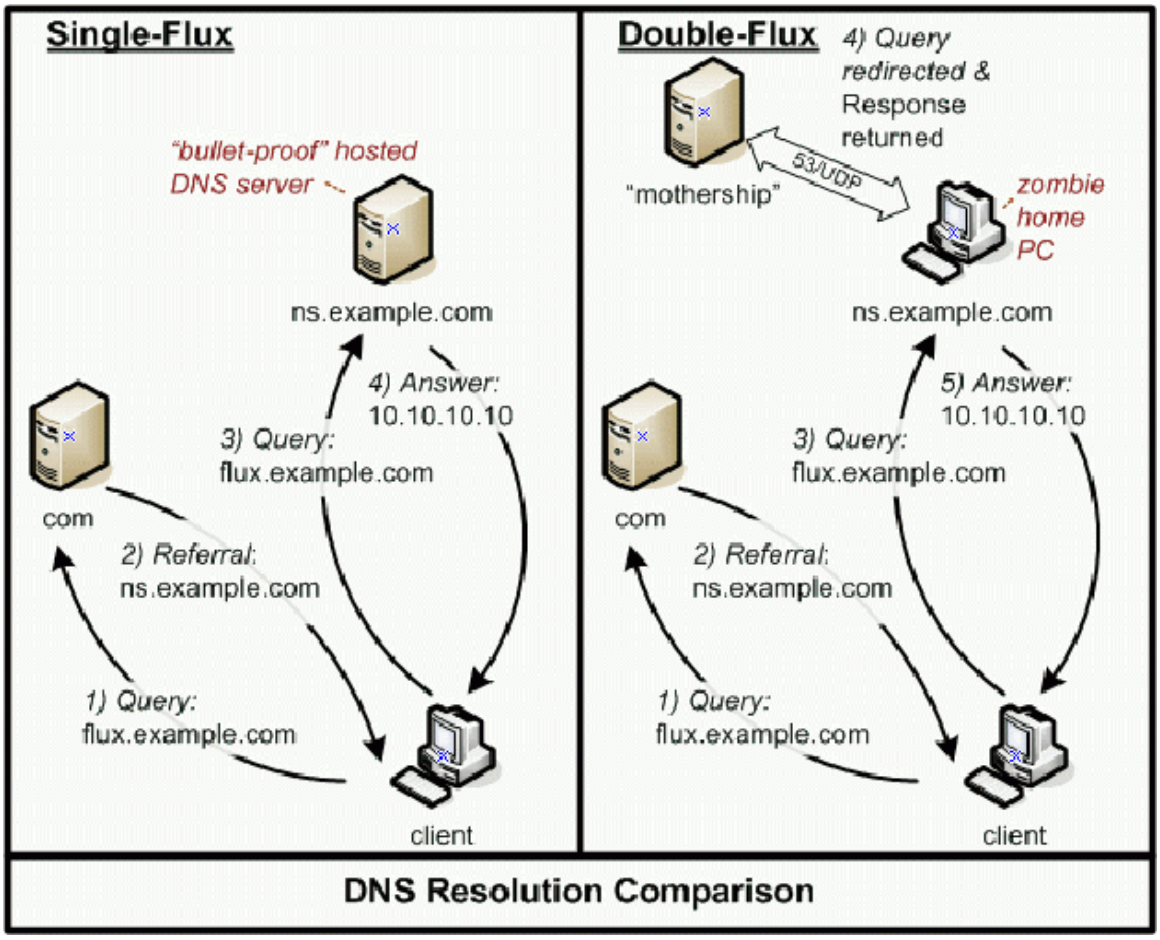

Figura 8: single-flux y double-flux.

La defensa ante este tipo de sistemas ya no se trata tan solo del bloqueo de direcciones *IP*, pues una *botnet* puede contener tantas asignaciones de nombre de dominio como *bots*. La única forma de detenerlo sería borrar a nivel de registro el nombre de dominio. Por desgracia, esto se convierte en una tarea complicada, requiere mucho tiempo y esfuerzo, especialmente si se tiene en cuenta que no todas las entidades registradoras de nombres de dominio atienden a las denuncias por abuso.

3.2. Arquitectura C&C descentralizado

A finales de abril del 2006 fue advertida por la comunidad anti-virus/malware la existencia de Nugache[29]. Se trataba de una *botnet* fácil de detectar y deshabilitar, ya que dependía de

[29] Nugache, primer botnet basado en tecnología *P2P*.

una limitada lista de servidores y sus conexiones salientes eran siempre a través del puerto 8 TCP.

A principios de septiembre de 2007 se lleva a cabo la detención del autor de Nugache. Pero con ella no acaba el daño realizado por esta red, pues Nugache ha sido la primera red *botnet* puesta en funcionamiento basada en *P2P* para su C&C pero no será la última.

Lo realmente preocupante era el precedente que sembraba Nugache, pues si su autor compartía con el resto de la comunidad las bases de su obra, muchos seguirían sus pasos dando pie a la aparición de nuevos y más sofisticados sistemas *botnet*, con sus deficiencias corregidas y funcionalidades añadidas.

Con el fin de entender la aportación de la arquitectura *P2P* a las redes *botnet* iniciaremos una breve descripción de la misma, marcando sus características más notables y diferenciando los diferentes tipos según su estructura.

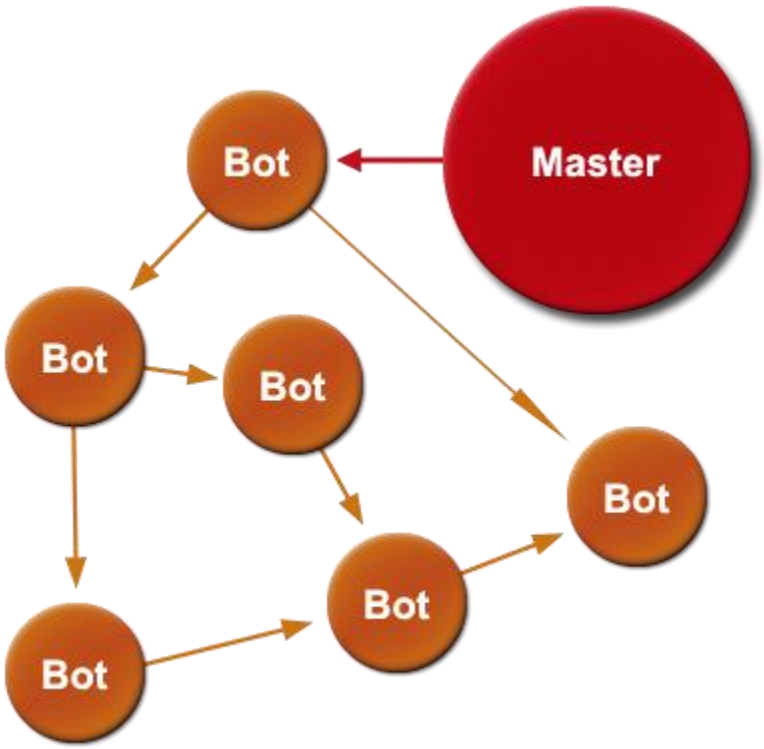

Figura 9: Estructura C&C descentralizado.

1

3.2.1. Redes *P2P*

Una *botnet P2P* representa un sistema más robusto que los utilizados hasta el momento, pues sobre la teoría, una red *botnet* basada en *P2P* no se puede interrumpir por completo, como mucho se puede irrumpir en las comunicaciones de algunos nodos, aun así solo se está deteniendo una parte de la red. El comando y control de la misma se realiza de forma totalmente descentraliza, las órdenes llegan a través de cualquier nodo de la red.

En una red *P2P* los *bots* realizan peticiones en base a un logaritmo solo conocido por su *botmaster*, de esta forma puede anticiparse generando el archivo, lo colocará en uno o varios nodos de la red y según lleguen las peticiones de sus semejantes se distribuirá dicho archivo.

Redes *P2P* no estructuradas

Cuando en una red *P2P* no existe ninguna relación específica entre nodos y contenidos, para la búsqueda de un archivo se envía la petición a todos los nodos a través de la red (flood). Es el caso de las llamadas redes *P2P* no estructuradas.

El funcionamiento óptimo de este tipo de arquitectura *P2P* se basa en que los contenidos sean mayoritarios, que un gran número de nodos compartan el archivo, pero cuando el contenido es minoritario el efecto del *flood*[30] incrementa muchísimo el tráfico en la red, reduciendo la eficiencia de la búsqueda.

[30] *Flood* es un término en inglés que significa literalmente inundación. Se usa en la jerga informática para designar un comportamiento abusivo de la red de comunicaciones, normalmente por la repetición desmesurada de algún mensaje en un corto espacio de tiempo.

Redes *P2P* estructuradas

En las redes *P2P* estructuradas todo contenido está asociado con el nodo que lo almacena. De manera que la búsqueda de cualquier archivo por raro y minoritario que sea, dará buenos resultados.

La herramienta más utilizada en este tipo de redes son las tablas de hash distribuidas[31], ideadas para localizar los nodos donde se encuentran los archivos deseados con mayor rapidez.

3.2.2. La idea de una red descentralizada

La Figura 10 incluye la evolución en sus inicios de las redes *P2P* en relación a los primeros *bots*.

[31] *Hash* se refiere a una función o método para generar claves o llaves que representen de manera casi unívoca a un documento, registro, archivo, etc., resumir o identificar un dato a través de la probabilidad, utilizando una función hash o algoritmo hash. Un hash es el resultado de dicha función o algoritmo.

Date	Name	Type	Distinguishing Description
12/1993	EggDrop	Non-Malicious Bot	Recognized as early popular non-malicious IRC bot
04/1998	GTbot Variants	Malicious Bot	IRC bot based on mIRC executables and scripts
05/1999	Napster	Peer-to-Peer	First widely used hybrid central and peer-to-peer service
11/1999	Direct Connect	Peer-to-Peer	Variation of Napster hybrid model
03/2000	Gnutella	Peer-to-Peer	First decentralized peer-to-peer protocol
09/2000	eDonkey	Peer-to-Peer	Used checksum directory lookup for file resources
03/2001	Fast Track	Peer-to-Peer	Use of supernodes within the peer-to-peer architecture
05/2001	WinMX	Peer-to-Peer	Proprietary protocol similar to FastTrack
06/2001	Ares	Peer-to-Peer	Has ability to penetrate NATs with UDP punching
07/2001	BitTorrent	Peer-to-Peer	Uses bandwidth currency to foster quick downloads
04/2002	SDbot Variants	Malicious Bot	Provided own IRC client for better efficiency
10/2002	Agobot Variants	Malicious Bot	Incredibly robust, flexible, and modular design
04/2003	Spybot Variants	Malicious Bot	Extensive feature set based on Agobot
05/2003	WASTE	Peer-to-Peer	Small VPN-style network with RSA public keys
09/2003	Sinit	Malicious Bot	Peer-to-peer bot using random scanning to find peers
11/2003	Kademlia	Peer-to-Peer	Uses distributed hash tables for decentralized architecture
03/2004	Phatbot	Malicious Bot	Peer-to-peer bot based on WASTE
03/2006	SpamThru	Malicious Bot	Peer-to-peer bot using custom protocol for backup
04/2006	Nugache	Malicious Bot	Peer-to-peer bot connecting to predefined peers
01/2007	Peacomm	Malicious Bot	Peer-to-peer bot based on Kademlia

Figura 10: Evolución de protocolos P2P y bots.

EggDrop[32] es uno de los *bots* más populares en IRC, por su gran capacidad de automatización de tareas y las innovadoras funcionalidades que incluía, como jugar a juegos, transferencia de archivos, etc.

Napster[33] fue el primer sistema *P2P* centralizado. Permitía compartir música con el resto de miembros de la red. Se basaba en un servidor centralizado que almacenaba los índices de contenido disponibles en las máquinas de los usuarios. Las búsquedas iban dirigidas a este servidor que daba la localización del archivo musical requerido para iniciar la descarga.

[32] Eggdrop, bot basado en IRC

[33] Napster fue un servicio de distribución de archivos de música (en formato MP3) Uno de los pioneros de las redes P2P de intercambio creado por Shawn Fanning.

Durante el uso de redes centralizadas y no estructuradas, los servidores contenían índices de contenidos que los convertían en claros cómplices del delito de violación de derechos de autor, y se veían en un breve espacio de tiempo, clausurados y denunciados por las instituciones defensoras de estos derechos.

Gnutella[34] fue la primera red *P2P* descentralizada. Aunque se trataba de *P2P* no estructurado, pronto nuevos protocolos como Kademlia[35], se iniciarían en el uso de tablas de hash para mejorar la eficiencia de las búsquedas.

3.2.3. Kademlia, como protocolo *P2P* más popular

Kademlia es un protocolo de la capa de aplicación[36] diseñado para redes *P2P* descentralizadas. Los nodos se comunican entre sí usando el protocolo sin conexión *UDP*[37]. Con Kademlia se crea una nueva red virtual sobre Internet, en la cual cada nodo de la red es identificado por un número (ID del Nodo).
Kademlia es un protocolo que no requiere servidores. Todos los nodos hacen la función tanto de cliente como de servidor.
Cuando se realiza una búsqueda, cada cliente actúa como un pequeño servidor y se le da la responsabilidad de ciertas palabras clave o fuentes. Esto añade complejidad al encontrar fuentes, ya que no existe un servidor central al que preguntar, pero a cambio se propagará la consulta a través de la red. Y puede ser cualquiera de los nodos el que responda

---

[34] Gnutella es un proyecto de software distribuido para crear un protocolo de red de distribución de archivos entre pares, sin un servidor central.

[35] Kademlia es un protocolo de la capa de aplicación diseñado para redes P2P descentralizadas. Especifica la estructura de la red, regula la comunicación entre nodos y el intercambio de información.

[36] Consultar Niveles OSI, Modelo de referencia de Interconexión de Sistemas Abiertos
http://es.wikipedia.org/wiki/Modelo_OSI

[37] User Datagram Protocol (UDP) es un protocolo del nivel de transporte basado en el intercambio de datagramas. Permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión, ya que el propio datagrama incorpora suficiente información de direccionamiento en su cabecera.

(característica perfecta para ocultar el origen de los archivos de comandos enviados por un *botmaster*).

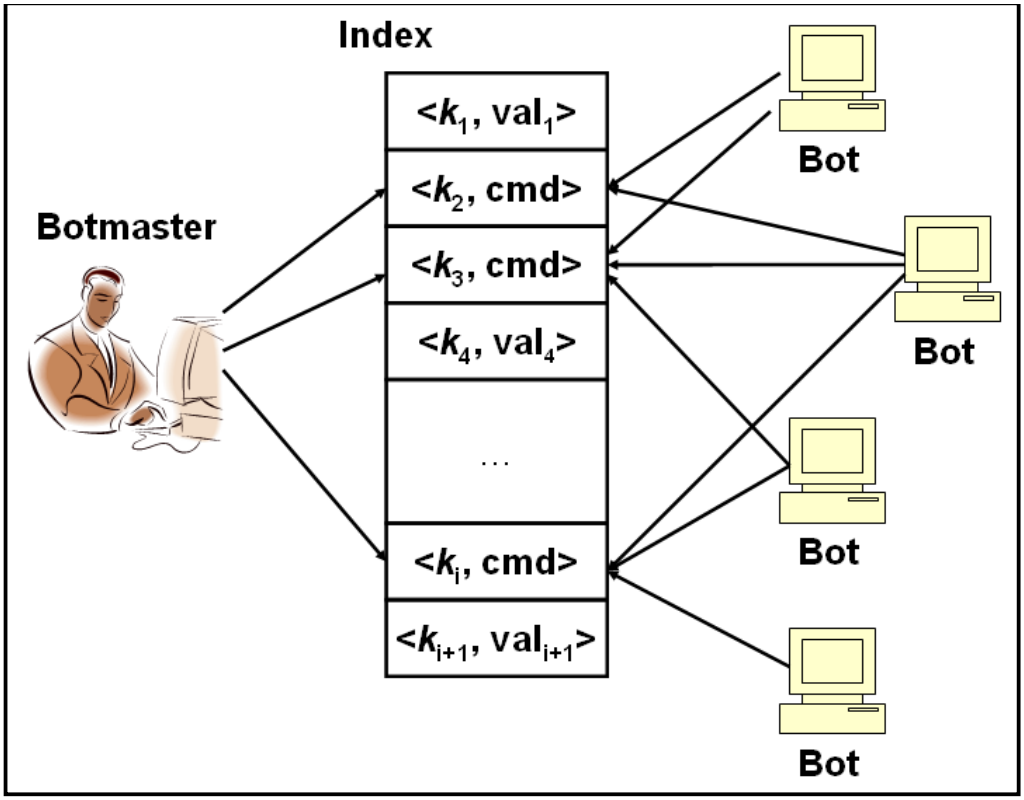

Figura 11: botnet basada en tablas de hash distribuidas.

Kademlia usa tablas de hash distribuidas para estructurar las búsquedas de archivos. Sus variantes más populares son eMule[38] o Bittorrent[39]. El uso masivo de estas variantes para el intercambio de archivos es un ambiente perfecto para camuflar una red *botnet*, confundiendo nuestros paquetes con los de cualquiera de estas redes *P2P* de uso popular en la actualidad.

[38] eMule es un programa para intercambio de archivos con sistema P2P utilizando el protocolo eDonkey 2000 y la red Kad, publicado como software libre para sistemas Microsoft Windows.

[39] BitTorrent es un programa que permite a un usuario establecer una conexión tipo P2P para descargar ficheros que otros usuarios poseen y que están dispuestos a compartir basándose en la filosofía del mismo BitTorrent (compartir por igual para todos) para facilitar el intercambio de los mismos.

Bajo esta idea surgieron sistemas *botnet* como Peacomm, más conocido como Storm, uno de los *bots* más extendidos bajo la arquitectura *P2P*. Éste está basado en la arquitectura de Kademlia.

Para la comunicación entre *botmaster* y *bot*, Peacomm se basa en valores de hash predecibles, ya que tanto *botmaster* como sus *bots* generan un valor de hash, introduciendo la fecha a una función conocida por ambas partes. De esta forma el *botmaster* puede almacenar comandos para los nodos bajo ese valor de hash, y a continuación los nodos requerir el contenido.

Los nodos miembros de la red tienen asignado un identificador de nodo de 160 bits, al igual que los archivos una *key*[40] del mismo tamaño. Los nodos normalmente contienen los archivos cercanos a su id de nodo. La *key* que identifica un archivo se calcula usando una función de hash, mientras los id de nodo son generados aleatoriamente por el mismo nodo.

Peacomm aprovecha estas características de Kademlia para evitar el contacto directo con todos los *bots* por parte del *botmaster*. Éste solo debe almacenar las órdenes o comandos bajo claves de búsqueda específicas. Los *bots* independientemente calcularán y realizarán una petición con esa clave de búsqueda obteniendo así los comandos.

Las compañías dedicadas a la seguridad informática seguían en la lucha, y con el estudio de la red hallaron la forma de anticiparse a la creación de estas claves. De esta forma podían entrar a formar parte de la red con un id de nodo cercano a la clave requerida y así identificar las *IP*s de los equipos comprometidos. Fue un duro golpe que inhabilitó gran cantidad de redes botnet.

Como sucedió anteriormente con las *IRC botnet*, la comunidad delictiva no tardó en rectificar sus herramientas y buscar alternativas para evitar su detección y eliminación.

---

[40] Key: clave

Una de las alternativas a Peacomm es Overbot. Al igual que Peacomm utiliza las búsquedas para obtener los comandos del *botmaster*. La diferencia está en el aspecto de esas búsquedas, ya que en Overbot solo el *botmaster* puede identificarlas como peticiones de la red *botnet*, para el resto de nodos se trata de tráfico legítimo perteneciente a la red *P2P*.

Incluso cuando el atacante controla gran número de *bots* y es capaz de analizar las peticiones de los mismos, éste es incapaz de identificar las peticiones dirigidas a la red *botnet*, gracias a que cada *bot* emite peticiones diferentes, evitando que el atacante (empresas de seguridad y antivirus) identifique un patrón en dichas búsquedas.

Este logro se debe a que el *botmaster* utiliza en la comunicación con sus *bots* encriptación asimétrica[41], con un par de claves, la privada del *botmaster* y la pública que conocen todos los *bots*. De esta forma solo el *botmaster* puede identificar usando la clave privada los mensajes pertenecientes a la *botnet*.

Este recurso complica mucho la identificación de estos *bots*, aunque mediante un largo análisis y estudio de comportamientos anómalos finalmente son diferenciables del resto de clientes *P2P*. Por ejemplo a nivel de red, un análisis estadístico apreciaría el gran número de peticiones que realizan las máquinas comprometidas en relación a los clientes *P2P* legítimos.

La lucha en busca de las debilidades contrarios es constante, igual que los cambios en las arquitecturas de *botnet*.

---

[41] La criptografía asimétrica es el método criptográfico que usa un par de claves para el envío de mensajes. Las dos claves pertenecen a la misma persona a la que se ha enviado el mensaje. Una clave es *pública* y se puede entregar a cualquier persona, la otra clave es *privada* y el propietario debe guardarla de modo que nadie tenga acceso a ella. Si el remitente usa la clave pública del destinatario para cifrar el mensaje, una vez cifrado, sólo la clave privada del destinatario podrá descifrar este mensaje, ya que es el único que la conoce. http://es.wikipedia.org/wiki/Criptograf%C3%ADa_asim%C3%A9trica

El nuevo recurso de los ciberdelincuentes es fragmentar la red botnet, de forma que no todos los nodos puedan verse entre sí, proceden a la segmentación de la red *botnet* en varias más pequeñas. Así la intervención de una de ellas no afecta a las demás, viéndose restringidas las conexiones de cada nodo a su subred no cabe posibilidad de comprometer al resto de subredes.

El objetivo final en los sistemas *botnet* sería acercarse cada vez más a los ideales:
Que un nodo capturado no comprometa al resto de la red.
Que un nodo de la red zombi no se diferencie en nada de un cliente *P2P* legítimo.
Que las tareas de C&C nunca comprometan al *botmaster*.

3.3. Dos caminos: *HTTP* y *P2P*

No se debe olvidar el parámetro más importante en la *botnet* ideal, como es su alta rentabilidad, que la acción resulte lo más lucrativa posible es la principal prioridad.

Según esta premisa ejecutan sus movimientos los *botmaster*, que actualmente optan por utilizar estructuras *botnet* menos robustas que las basadas en *P2P* pero poder rentabilizarlas al máximo el tiempo que permanezcan activas.

En la práctica una *botnet P2P* continúa siendo vulnerable, es cierto que su erradicación total conlleva un proceso más laborioso, ya que el cada vez más cercano concepto de descentralización total refuerza su subsistencia.

Pero en este proceso de descentralización han mermado algunas de sus armas, en la migración de *botnet* a sistemas *P2P* se han perdido características de control y supervisión de flota, o el poder de sincronizar un ataque masivo con efectividad.

De ahí que muchos *botmaster* sigan confiando en sistemas *HTTP*, y se dediquen todavía a su investigación y desarrollo. Dando resultados realmente sorprendentes y logrando un alto nivel de descentralización en la actualidad.

Citamos el informe "*MessageLabs Intelligence"* de 2009 (Informe Anual 2009 de actividad de cibercriminales Artículo 1.) emitido por Symantec[42], y observamos actualmente el peso que continúan teniendo redes *botnet* basadas en *HTTP*, en concreto Conficker[43], la cual describen como: "La mayor amenaza de seguridad de este año".

> […]
> La mayor amenaza de seguridad de este año ha sido el gusano **Conficker/Downadup**, que permite a sus creadores instalar software de forma remota en las máquinas infectadas. El gusano Conficker se originó a finales de 2008, pero el 1 de abril de 2009 se le añadió una actualización para malware, una funcionalidad adicional que lo mejoraba evitando su detección. Conficker es un gusano especialmente preocupante, puesto que todavía se desconoce como los ordenadores infectados podrían ser utilizados. Según Conficker Working Group, un grupo que ha contribuido a reducir al mínimo el papel que este malware podría haber jugado en 2009, este gusano infectó a más de seis millones de ordenadores.
> […]

Informe Anual 2009 de actividad de cibercriminales emitido por Symantec. [1]

Sistemas como *fast-flux* redirigen el tráfico de *C&C* en cuestión de segundos hacia las nuevas ubicaciones de los servidores de *C&C*, donde los zombis recogen sus órdenes. Miembros de la red de zombis son asignados como servidores de nombres de dominio (*double fast-flux*) o proxys al servicio de las comunicaciones de la red *botnet*. Estos rápidos y numerosos cambios dificultan su desarticulación, prolongando la vida de la *botnet* sin sacrificar ninguna de sus armas de explotación y manteniendo sus prestaciones.

[42] Symantec Corporation, empresa dedicada al sector de la seguridad informática.

[43] Conficker, botnet basado en la combinación de P2P y http en su comando y control.

4. Negocio *botnet*

En los últimos 10 años, las redes zombi o *botnets* han experimentado una evolución enorme, de pequeñas redes formadas por una decena de equipos y administradas desde un centro a convertirse en complicados sistemas distribuidos de administración descentralizada que constan de millones de equipos, los profesionales dedicados al sector de la seguridad informática se sorprenden cada día más, del alto nivel profesional con el que son elaboradas estas herramientas para el fraude.

Y es que un código de tal complejidad se atribuye a profesionales del sector del *software*, programadores profesionales con pocos escrúpulos son los autores de estos *bots*, muchos de ellos hasta el momento al margen de la actividad *hacker y* que atraídos por la alta rentabilidad de su trabajo al servicio del delincuente, han decidido cruzar la línea de la legalidad.

En la mayoría de ocasiones no serán los *hackers* quienes "aprieten el gatillo" sino que pondrán su trabajo a disposición del delincuente a cambio de compensación económica. Todo esto genera un mercado paralelo a la actividad *hacker*, donde se comercia con información y herramientas para el fraude en la red.

Aunque se trate de actividad ilegal y los tratos sean clandestinos, la relación proveedor cliente es exactamente igual que en cualquier empresa, pues el hacker ha encontrado un comprador fiable para su trabajo y quiere mantenerlo, un objetivo común los llevará a una estrecha colaboración, mantener la *botnet*. El programador continuará creando nuevos complementos para sus *bots*, con el fin de aumentar el tamaño de la red zombi y su explotación.

4.1. Métodos de explotación

Para su explotación veremos a continuación las principales tendencias: ataques *DDOS* (Ataque de denegación de servicio), robo de información confidencial, envío de *spam* (envío masivo de correo no deseado), fraude de clics y descarga de *software* malicioso.

Cualquier método es válido para rentabilizar una *botnet*, además un mismo *bot* puede utilizar todos los métodos simultáneamente, y mediante nuevos complementos, adaptarse para abordar nuevos fraudes que aparezcan en el futuro.

4.1.1. Ataques *DDOS*

Los ataques *DDOS* dejan incomunicado el equipo informático víctima, mediante el envió de gran cantidad de solicitudes hasta que ésta sea incapaz de procesarlas. Estos ataques son utilizados en entornos empresariales contra la competencia, o para la extorsión y chantaje por parte del dueño de la *botnet*, el cual pide un rescate a cambio de liberar las comunicaciones de la entidad.

En la red se encuentra gran cantidad de publicidad de servicios de ataque *DDOS*, al alcance de cualquiera en algunos foros específicos de este entorno delicitivo. Los ataques pueden costar desde 50 hasta miles de dólares, según la magnitud del ataque, cuantos zombis son necesarios para que el ataque logre bloquear a la víctima y durante cuánto tiempo se mantendrá la incomunicación.

4.1.2. Robo de información confidencial

La información que alberga en su máquina el usuario como números de tarjetas de crédito, documentos, contraseñas de servicios online, entre otros, son un suculento pastel para los ciberdelincuentes.

Con esta información podrían utilizar tarjetas de crédito ajenas, suplantar al usuario en aplicaciones de mensajería instantánea, utilizar cuentas de servicios como *PayPal*[44], etc.

Por parte del *hacker* toda esta información sustraída se pondrá a la venta en los numerosos escaparates que aloja Internet para el mercadeo ilícito. En estos casos, los datos bancarios son los más cotizados, según el saldo de la cuenta su precio puede oscilar hasta los 2000 dólares, aunque lo más normal es encontrar ofertas por la módica cantidad de unos diez euros. Y es que este mercado como cualquier otro disfruta de libre competencia, por lo que los precios se ajustan al máximo. Es casi tan importante la frescura de los datos como el saldo que alberga la cuenta. El caché del hacker en cuestión es un valor añadido en este tipo de ventas, directamente proporcional al tamaño y calidad de su *botnet*.

El resto de datos del usuario o las cuentas de correo electrónico también tiene lugar en estos foros *underground*[45] convertidos en tiendas virtuales.

Cuentas de servicios como *PayPal*, usuarios *FTP*[46], etc., conllevan fraudes no tan comprometedores como los datos bancarios, más difíciles de rastrear por las autoridades y aunque en menor medida, muy rentables.

Recopilar direcciones de correo para su posterior uso por parte de los *spammers* también da sus frutos, y aunque su precio es muy bajo, se pueden obtener sencilla y rápidamente, ya que cada máquina alberga una gran cantidad de direcciones.

Los datos personales completos de los usuarios sirven a todo tipo de delincuentes (no solo ciberdelincuentes) para abrir cuentas, falsificar documentos, y efectuar todo tipo de

---

[44] PayPal es una empresa estadounidense perteneciente al sector del comercio electrónico por Internet que permite la transferencia de dinero entre usuarios que tengan correo electrónico, una alternativa al tradicional método en papel como los cheques o giros postales.

[45] Underground (subterráneo en español) es un término inglés con el que se designa a los movimientos contraculturales que se consideran alternativos, paralelos, contrarios o ajenos a la cultura oficial.

[46] FTP (sigla en inglés de File Transfer Protocol - Protocolo de Transferencia de Archivos) en informática, es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP.

autentificación sin levantar la más mínima sospecha ya que los datos pueden ser contrastados con una identidad real.

Los datos de tarjetas de crédito son utilizados por falsificadores de las mismas. Hace dos años se arrestó a unos delincuentes brasileños que mediante este método llegaron a robar más de 4 millones de dólares de cuentas bancarias de los usuarios.

4.1.3. Hosting

Existe gran cantidad de sitios web dedicados a la actividad ilegal, que tras ser denunciados se ven clausurados de inmediato. Hablamos de contenidos como: venta de fármacos ilegal, *spam, scam*[47]*, phishing*[48], distribución de troyanos o *botnet*. Es por eso que está empezando a proliferar el llamado *hosting without abuse* (*Hosting* sin abusos), la característica especial de este tipo de *hosting*[49], es que al propietario de dicho servicio no le preocupa en absoluto el contenido que pongan sus clientes, además de comprometerse a ignorar todo tipo de denuncias y críticas de los usuarios.

En el mantenimiento de estos *host* es donde entran los sistemas *botnet* con sus recursos de redireccionamiento basados en tecnologías *fast-flux* y múltiples *proxys*, con ellos es posible el cambio de las *IPs* regularmente garantizando su subsistencia y alargando su vida.

4.1.4. Spam

---

[47] Scam (estafa en inglés) es un término anglosajón que se emplea para designar el intento de estafa a través de a un correo electrónico fraudulento (o páginas web fraudulentas).

[48] Phishing es un término informático que denomina un tipo de delito encuadrado dentro del ámbito de las estafas cibernéticas, y que se comete mediante el uso de un tipo de ingeniería social caracterizado por intentar adquirir información confidencial de forma fraudulenta (como puede ser una contraseña o información detallada sobre tarjetas de crédito u otra información bancaria).

[49] El alojamiento web (en inglés web hosting) es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía Web.

El *spam* o correo no deseado es utilizado para publicitar productos y servicios, el aumento de ventas es considerable, y mientras sea así seguirá proliferando esta práctica.

> [...]
> Pues se ha revelado, que tan solo 8 de tales redes de botnets son responsables de casi el 100% de todo el SPAM y crimen en Internet, y de esos 8, tan solo uno de ellos es responsable de casi el 40%.
> [...]

8 botnets responsable del casi 100% del SPAM global. [2]

> […]
> En cuanto a Pushdo, todavía se desplaza sin levantar grandes polvaredas, si bien está ubicado en el puesto número dos en la fila de botnets más grandes de todo el mundo. Según la especialista en seguridad Trend Micro, este código malicioso envía entre cerca de 8 mil millones de correos no deseados por día. Sí, es una cifra alarmante: esto significa que uno de cada 25 correos es enviado por Pushdo.
> […]

8 mil millones de *spam* diarios por un *botnet.* [3]

Los artículos 4 y 5 revelan la importancia de los sistemas *botnet* para tareas de envío de correo masivo, siendo los principales responsables de casi la totalidad del *spam* mundial.

En la actualidad y gracias a las *botnets* el uso de *spam* para publicidad se ha extendido a otros segmentos de la red, como son: redes sociales, aplicaciones de mensajería instantánea, foros de discusión e incluso a los blogs. Nuevos horizontes para la economía del delincuente.

4.1.5. Fraude de clics

Las compañías publicitarias que funcionan *online* según el esquema *PPC (Pay-per-Click)* pagan dinero por cada clic único en sus anuncios. Para los propietarios de *botnets*, engañar a estas compañías es una actividad bastante lucrativa.

Como ejemplo podemos tomar la conocida red Google[50] AdSense. Los anunciantes pagan a Google por los clics en los anuncios, con la esperanza de que el visitante compre algo en sus tiendas *online*[51].

Por su parte, Google pone los anuncios contextuales en los diferentes sitios participantes en el programa AdSense y pagan al dueño del sitio por cada clic. Desgraciadamente, no todos los propietarios de sitios son honrados. Así, disponiendo de una *botnet*, un hacker puede generar miles de clics únicos al día, uno en cada equipo zombi, para no despertar las sospechas de Google.

[50] Google Inc. es la empresa propietaria de la marca Google, cuyo principal producto es el motor de búsqueda del mismo nombre. Fue fundada el 4 de septiembre de 1998 por Larry Page y Sergey Brin.

[51] Tiendas dedicadas a la venta en Internet.

5. Toma de conciencia

Internet sigue creciendo a pasos agigantados, cada día hay más hogares conectados a la red, cada día más empresas cambian el fax por el e-mail, y así cada día el *hacker* ve más posibilidades, es consciente de este crecimiento y no dejará de aprovecharlo. En los inicios de la red de redes la extensión de su territorio era escasa, sólo unos pocos usuarios y entidades estaban conectados a la red, en consecuencia también era escasa la rentabilidad de la actividad ilegal, pero con el crecimiento de Internet el delito ha empezado a ser muy tentador a la vez que lucrativo.

5.1. Software antivirus

Crece Internet en número de conexiones y con ella el número de posibles víctimas, sin la toma de conciencia del usuario, cada nueva conexión a internet en el hogar o empresa estará alimentando esta actividad ilegal.

El artículo citado a continuación (Art.2) refleja los datos recogidos durante el mes de octubre de 2009 por PandaLabs[52] (Red de Laboratorios de Investigación de Virus), según los cuales, como refleja su título: "España encabeza el ranking mundial de infecciones de virus".

Al final del artículo hacen una recomendación: "Recordad tener un buen antivirus instalado y actualizado".

[52] Panda Security S.L., anteriormente Panda Software, es una compañía multinacional de seguridad informática fundada en 1990 por el ex director general de Panda, Mikel Urizarbarrena, en la ciudad de Bilbao, en España.

El usuario debe ser consciente de la continua evolución de los virus y *botnet*, viendo las continuas modificaciones y nuevas creaciones de *software* malicioso, una gran cantidad de personas, la gran mayoría parte de empresas dedicadas al desarrollo de antivirus, persiguen la forma de protegernos de las nuevas amenazas. Ese esfuerzo tiene como resultado actualizaciones casi diarias de nuestro software antivirus.

España encabeza el ranking mundial de infecciones de virus

por PDD20 » 08 Nov 2009, 13:07

Según los datos recogidos por **PandaLabs** durante el mes de octubre, España encabeza el ranking (con un 44,49% del total) de ordenadores infectados por *bots*, software que una vez introducido en el PC sirve para que los hackers tomen control remoto del mismo y puedan realizar diferentes acciones, como el envío de spam, de virus, gusanos y troyanos, etc.

Le sigue muy por debajo Estados Unidos, con un 14,41%; México, con un 9,37, y Brasil, que baja a un 4,81%. Entre los países menos infectados, figura Perú, Holanda y Suecia, con ratios por debajo del 1%.

Estas redes de *bots* las crean los hackers para utilizarlas con diferentes fines. Entre ellos, para el envío de spam o la distribución de virus. En nuestro país, Madrid encabeza el ranking de provincias con ordenadores infectados que están enviando spam, con un 25,92%; seguido por Barcelona, con un 15,52%, y Sevilla, con un 6,06%.

Recordad tener un buen antivirus instalado y actualizado.

España encabeza el ranking mundial de infecciones de virus. [4]

Es importante sigan trabajando con la misma intensidad que hasta ahora, pues de ello depende, en gran parte, el número de víctimas infectadas diariamente.

5.2. Responsabilidad del usuario

No se debe dar toda la responsabilidad a nuestro software antivirus, pues la explosión de *malware*[53] de 2009 y las nuevas amenazas están continuamente tratando de evadir los sistemas de detección tradicionales.

Los ataques tienen nuevos frentes que el antivirus no puede proteger, frentes como la ingeniería social son por ahora responsabilidad exclusiva del usuario final. Se conoce como ingeniería social cualquier método de engaño que logre inducir a la víctima a instalar el *bot* ella misma.

Este método utiliza herramientas como las redes sociales, foros, *spam*, o incluso los programas de mensajería instantánea. Ocultar el *bot* en cualquier tipo de objeto atractivo para la víctima, incluyendo el código malicioso en archivos como videos, imágenes, música, o en el mismo código *HTML* de una web, en cuyo caso será el navegador el que automáticamente instalará el *bot* al pulsar sobre el enlace.

En estos casos, poco tiene que hacer el software antivirus, pues aparentemente el software es instalado por el administrador del sistema y se reconoce como legítimo.

Por eso es igual o más importante la toma de conciencia del usuario que tener un antivirus actualizado. En esa dirección, la de poner en conocimiento del usuario ciertas precauciones que debe tomar en el uso de Internet, van las siguientes líneas, enumerando dichas precauciones.

5.2.1. Medidas contra la infección

Tener instalado *software* antivirus y actualizado es primordial como hemos dicho, por eso siempre se debe de configurar para que se actualice automáticamente.

---

[53] Malware (del inglés malicious software, también llamado badware, software malicioso o software malintencionado) es un software que tiene como objetivo infiltrarse en el sistema y dañar la computadora sin el conocimiento de su dueño, con finalidades muy diversas, ya que en esta categoría encontramos desde un troyano a un spyware.

En las opciones de seguridad del navegador, la configuración debe ser lo más restrictiva posible.

Además, mientras se trabaja con Internet no son necesarios permisos de administrador para ninguna de las tareas rutinarias, no se debe tomar por costumbre el uso del usuario "Administrador" como válido para todo en todo momento. Se debería crear un usuario con derechos limitados para el trabajo en Internet.

La actualización del sistema operativo y las aplicaciones instaladas en el sistema también previene las infecciones, ya que los estudios de los "ciberdelincuentes" desvelan nuevas debilidades continuamente.

Respecto al sistema operativo, cuanto menos lleve en el mercado, menos tiempo para estudiar sus debilidades han tenido los hackers, siendo menos vulnerable ante posibles amenazas.

En cuanto a evitar la infección mediante ingeniería social, son muchas las precauciones a tomar y en un corto periodo de tiempo se multiplican los métodos de infección conocidos, a continuación se nombrarán algunos a modo de prevención, pero la norma general que rige estos ataques es el desconocimiento, el "timo de la estampita" o el "dar gato por liebre".

Figura 12: Intento de infección IM.

Por norma general para evitar los ataques basados en ingeniería social, lo más importante es conocer el origen de todo el contenido que se visita o descarga de Internet y su fiabilidad.

Nunca haga clic en los archivos adjuntos en mensajes de *e-mail* a menos que pueda verificar su origen. El *spam* es un método de distribución de *bots* muy efectivo, *mails* con título: "Mira las fotos de la última fiesta", pueden albergar código malicioso.

> […]
> Una de las botnets utilizadas para enviar spam (denominadas spambots) más activas durante 2009, fue la red armada por Waledac. Este troyano se ha propagado desde finales de 2008 con diferentes técnicas de Ingeniería Social, logrando su crecimiento más importante en febrero de 2009 con la campaña de falsas postales por el día de San Valentín.
> […]

Waledac, el troyano enamorado. [5]

El mismo mensaje anterior, lo escribe un contacto mediante un programa de mensajería instantánea, y a continuación remite un archivo *ZIP*[54] o un enlace a una web. Puede que ese contacto este infectado por un *bot* y no tenga conocimiento de ello. No acepte ni entre en la *URL*[55] hasta asegurarse de que su contacto se lo envía y no es obra de un *bot* que pretende infectar su máquina.

No utilice análisis de seguridad "gratuitos" que aparecen en muchos sitios web. Con gran frecuencia, son falsos.

El simple clic sobre un enlace de dudoso destino, pude conllevar una infección, se debe tener cuidado con el uso de enlaces cortos, donde solo se muestra parte de la *URL*.

La infección mediante la carga de una web en el navegador, aprovecha *exploits*[56] o vulnerabilidades del mismo, pero en todos los casos lo hace ejecutando *Javascript*[57]. En

---

[54] En informática, ZIP o zip es un formato de almacenamiento sin pérdida, muy utilizado para la compresión de datos como imágenes, programas o documentos.

[55] Un localizador uniforme de recursos, más comúnmente denominado URL (sigla en inglés de uniform resource locator), es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, para su localización.

[56] Exploit (del inglés to exploit, explotar o aprovechar) es una pieza de software, un fragmento de datos, o una secuencia de comandos con el fin de automatizar el aprovechamiento de un error, fallo o vulnerabilidad, a fin de causar un comportamiento no deseado o imprevisto en los programas informáticos, hardware, o componente electrónico (por lo general computarizado).

este caso, configurar los navegadores para avisar antes de ejecutar *JavaScripts* eliminará posibilidades de infección.

Estas y otras muchas medidas forman parte de la ya conocida educación del usuario. Se trata de conseguir usuarios inteligentes tal como afirma Richi Jennings, (analista jefe para seguridad en el correo electrónico de Ferris Research) en su artículo para ComputerWorld (Art. 3).

> […]
> Conseguir usuarios inteligentes
>
> En materia de seguridad, la educación del usuario es un elemento de prevención clave que todos tenemos a nuestro alcance, tanto en una gran empresa, como en una pequeña compañía, en una oficina local o en el hogar. Según la mayoría de los expertos, todo el problema reside en que la gente no deja de hacer clicken prácticamente cualquier elemento. Así, es necesario educar a los usuarios para que sean conscientes de que, igual que hay unas prácticas seguras para conducir, las hay para utilizar ordenadores. Además, Jennings añade que "si la gente dejase de comprar productos de spam, éste desaparecería". ¿O no?

Cuatro pasos para luchar contra los *botnets*. [6]

[57] JavaScript es un lenguaje de scripting basado en objetos, utilizado para acceder a objetos en aplicaciones. Principalmente, se utiliza integrado en un navegador web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas.

6. Conclusión

Durante el último año, a ojos de los usuarios, las amenazas e infecciones parecen haber disminuido, se puede tener la percepción de estar venciendo esta guerra, dando pie a una falsa sensación de seguridad.

Lo que ha sucedido realmente ha sido una migración por parte de los ciberdelincuentes. De manera que los escandalosos gusanos de infección masiva que antaño acosaban a los usuarios inundando las máquinas de publicidad y interrumpiendo su actividad han sido desterrados, desterrados o mejor dicho sustituidos.

Los estudios de compañías dedicadas al sector de la seguridad informática siguen reflejando un crecimiento en los ataques y amenazas. Esta contradicción se debe al uso de *botnet* como nuevo medio de subsistencia de estos delincuentes.

Botnet es sin duda la más rentable de las herramientas y también muy sigilosa en comparación a los gusanos que la precedieron, por eso existe esa falsa sensación de seguridad por parte del usuario.

Además, el sistema *botnet* es capaz de alimentar no solo a sus dueños sino a toda una comunidad a la cual lleva de la mano. Todos los negocios cibernéticos delictivos son influenciados por las *botnet.*

Los propietarios de *botnet* son clientes de servicios como la creación y cifrado de códigos maliciosos, la investigación y creación de *exploits,* el *hosting* a prueba de abusos y la distribución de mecanismo de infección. Financiando toda esta actividad antes y durante el funcionamiento de la *botnet.* Por no hablar de *spam,* tarjetas de crédito y datos robados, donde el *botnet* se ha convertido en herramienta imprescindible.

Por eso la actividad *botnet* se ha convertido en el engranaje principal en todo movimiento delictivo en la red. Ingresando enormes cantidades de dinero por medio del envío de *spam*

y robo de datos a los usuarios, a la vez que financia al resto de la comunidad de ciberdelincuentes solicitando sus servicios para la mejora y mantenimiento de la *botnet*.

Mediante la colaboración de los expertos en seguridad, los proveedores de Internet y las fuerzas del orden se lleva a cabo la lucha contra este fenómeno. Aunque hasta ahora ha sido del todo inefectivo. Puesto que las leyes que condenan toda esta actividad son inexistentes en muchos países, en los cuales pueden refugiarse estos *hackers.*

Por ahora, sin la ayuda de los usuarios, la lucha no puede ser efectiva. Los ordenadores domésticos forman la mayor parte del ejército de las *botnets.* Es necesario tomar medidas en los hogares. La información de los usuarios los educará en el uso de la red, para evitar el clickeo constante sin criterio alguno. El concepto es educar a los usuarios.

Mirando al futuro, los expertos en seguridad basan sus esfuerzos en el estudio del comportamiento de las *IPs* y los usuarios para próximas versiones de protección antivirus. Dejando atrás criterios basados en firmas, para abordar el comportamiento del usuario y las aplicaciones que utiliza como plantilla para detectar y desactivar posibles amenazas a partir de anomalías en el comportamiento de la máquina.

David Perry es un viejo zorro en la lucha contra los virus informáticos. Ha pasado por las más importantes empresas del sector y hoy es director mundial de formación de Trend Micro.

[…]

P. ¿El antivirus del futuro es tratarnos como tontos?

R. Porque tenemos que proteger de cada vez más cosas: correo basura, gusanos, ataques por web, troyanos, zombis, virus publicitarios, programas espía. Por suerte, nuestras metodologías son cada vez mejores. **En los próximos dos años acabaremos con las botnets**, un software que controla remotamente tu ordenador.

[…]

ENTREVISTA A DAVID PERRY, director mundial de formación de Trend Micro. [7]

Anexo I: Demostración

En la adquisición de una *botnet* se pueden encontrar varios tipos de producto: la posibilidad de alquilar una red *botnet* en pleno funcionamiento con un número considerable de máquinas ya infectadas y listas para llevar a cabo su explotación, o por el contrario, el hacker o desarrollador suministra los ingredientes necesarios para la puesta en marcha de la botnet.

El ingrediente principal sería el *bot*, que realiza la infección inicial convirtiendo en zombis a sus víctimas.

A continuación y para reforzar la expansión de la botnet podrían añadirse complementos a este *bot* inicial, que por ejemplo aprovechen nuevos exploits o se distribuyan mediante nuevos métodos como *IM*. O por otro lado estos complementos podrían formar parte de la explotación de la *botnet*, por ejemplo enviando *spam* publicitario de casinos *online*. Existen cantidad de posibilidades.

En este caso en particular, se ha partido de un *bot* inicial, llamado RXbot[58], basado en *IRC*, y entrando en el papel de un *botmaster* real, se reconstruye la acción, desde la adquisición del *bot* malicioso, su preparación y puesta en marcha siguiendo las instrucciones del autor del *bot.*

Se ha montado un banco de pruebas donde observar la actividad de la red *botnet* a pequeña escala, formada por un equipo infectado por RXbot. Un segundo equipo que alberga el servidor *IRC* para el comando y control de la *botnet.* Y por último un cliente *IRC* en el papel de *botmaster* que podrá conectarse desde un tercer equipo o desde el mismo servidor *IRC*.

[58] RXbot

Para realizar esta demostración con recursos mínimos, se utiliza virtualización[59], mediante la herramienta VMWARE[60] se ponen en marcha 3 equipos con sistema operativo Windows XP SP2.

- En el que llamaremos equipo A, será necesaria la instalación de un servidor *IRC*. En este caso se ha utilizado IRCPlus[61].

- En el equipo B, utilizado por el *botmaster,* es necesario un cliente *IRC,* aunque cualquier cliente serviría, en este caso se ha puesto en marcha mediante MIRC[62].

- El equipo C hará el papel de víctima por lo que no necesita ningún cambio ni actualización.

[59] Virtualización: medio para crear una versión virtual de un dispositivo o recurso, como un servidor, un dispositivo de almacenamiento, una red o incluso un sistema operativo, donde se divide el recurso en uno o más entornos de ejecución.

[60] VMware Inc., (VM de Virtual Machine) filial de EMC Corporation que proporciona la mayor parte del software de virtualización disponible para ordenadores compatibles X86.

[61] Software servidor de IRC.

[62] mIRC el cliente IRC más extendido para plataformas Microsoft Windows.

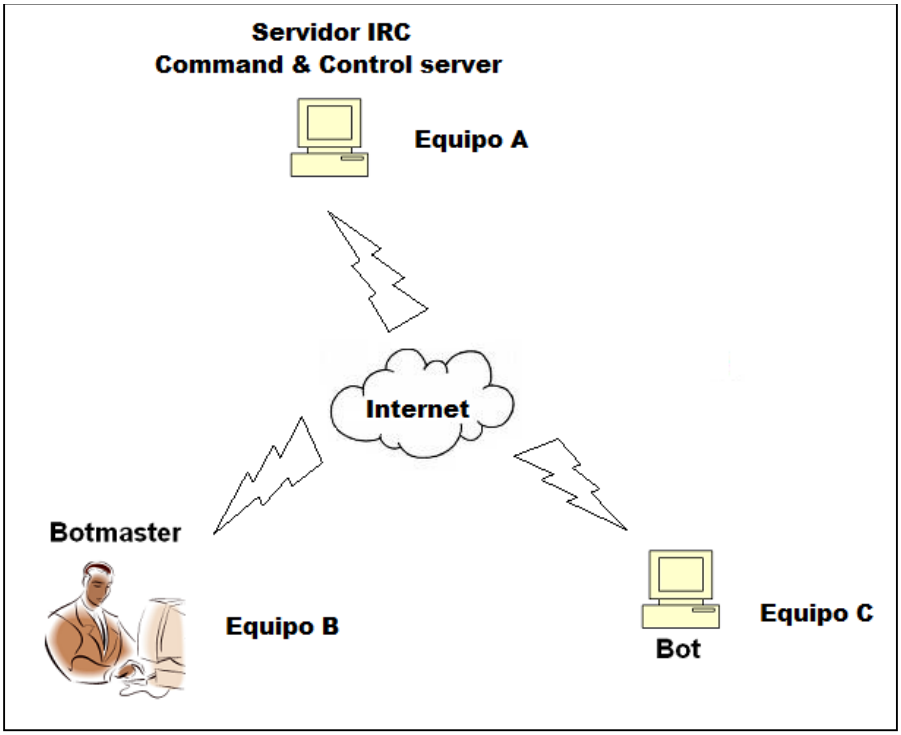

Figura 13: Estructura simulación red botnet.

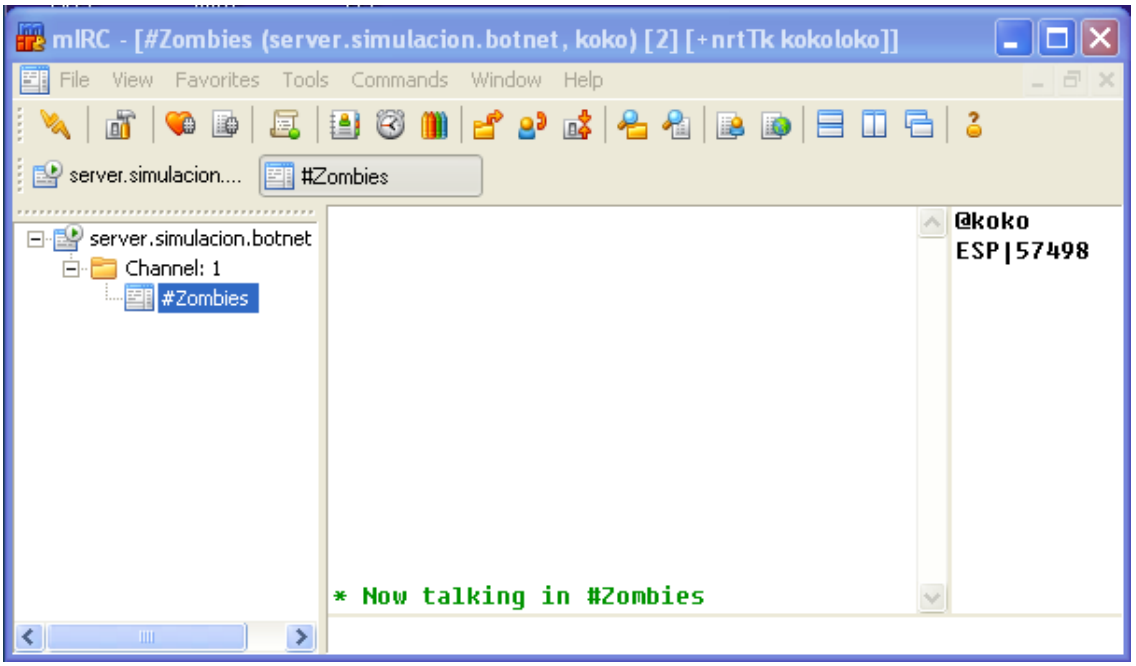

Figura 14:Canal #Zombies tras la infección del equipo C.

1. Se inicia la autentificación por parte del *botmaster* mediante el comando ".login" e inmediatamente el *bot* contesta "*Password accepted*."

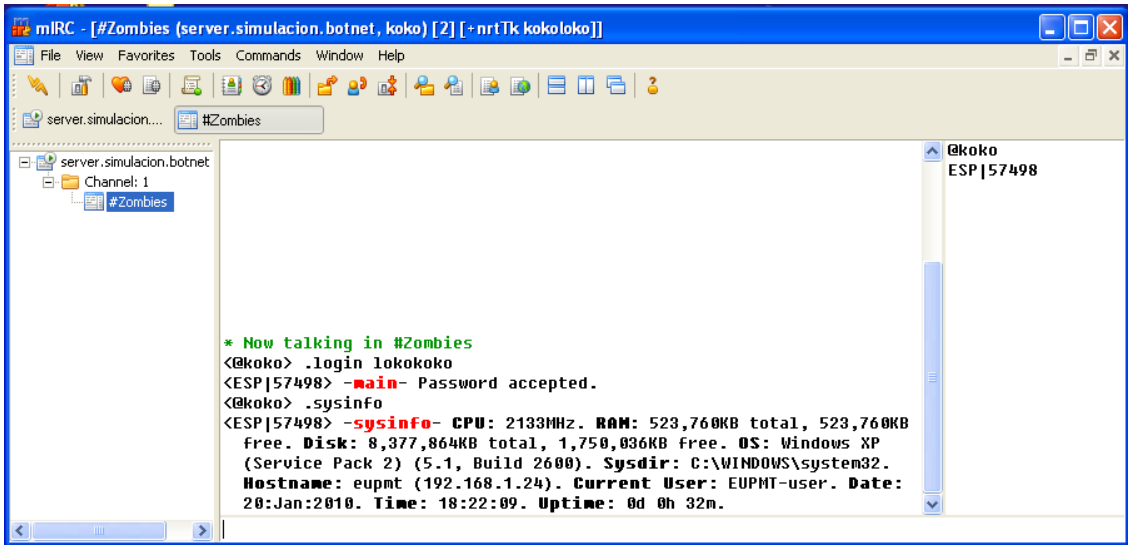

Figura 15: Comandos .login y .sysinfo.

2. En las siguientes capturas se demuestra realmente la infección del equipo C, se le ordena que haga una impresión de pantalla y la guarde en su disco duro.

3. El paso siguiente es requerirle que nos envíe dicha captura.

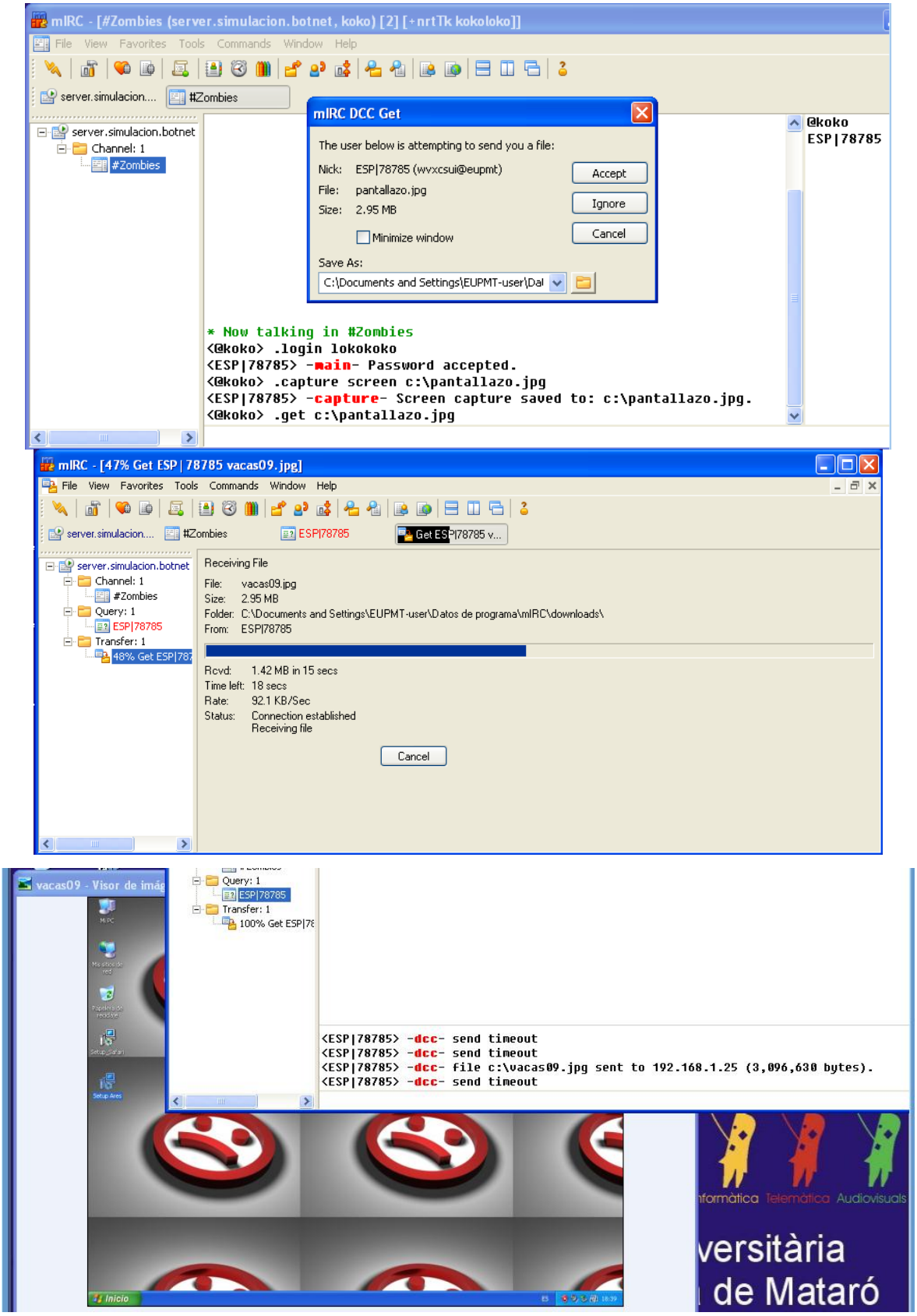

Figura 16: Captura de pantalla y posterior envió de la misma desde el equipo B.

4. La siguiente prueba inicia un proceso "keylogger"[63] desde el equipo B mientras la víctima en el equipo C consulta su cuenta de correo, obteniendo el *botmaster* toda la información de acceso a la cuenta.

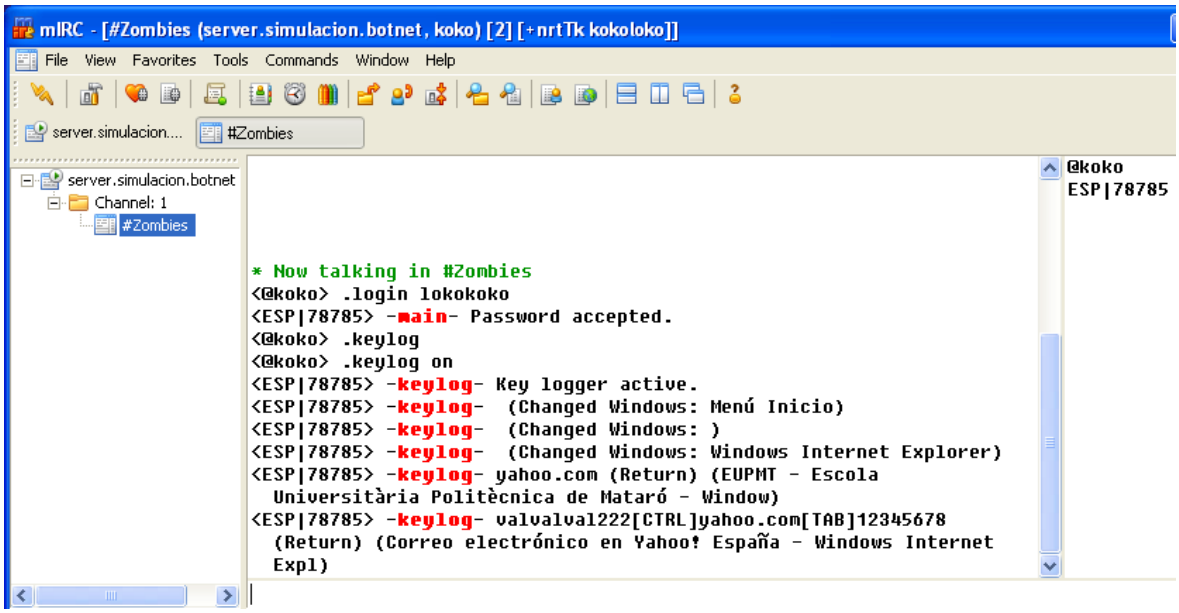

Figura 17: Pruebas keylogger en equipo B.

[63] Un keylogger (derivado del inglés: Key (tecla) y Logger (Registrador); registrador de teclas. Es un tipo de software que se encarga de registrar las pulsaciones que se realizan en el teclado, para memorizarlas en un fichero y/o enviarlas a través de internet.

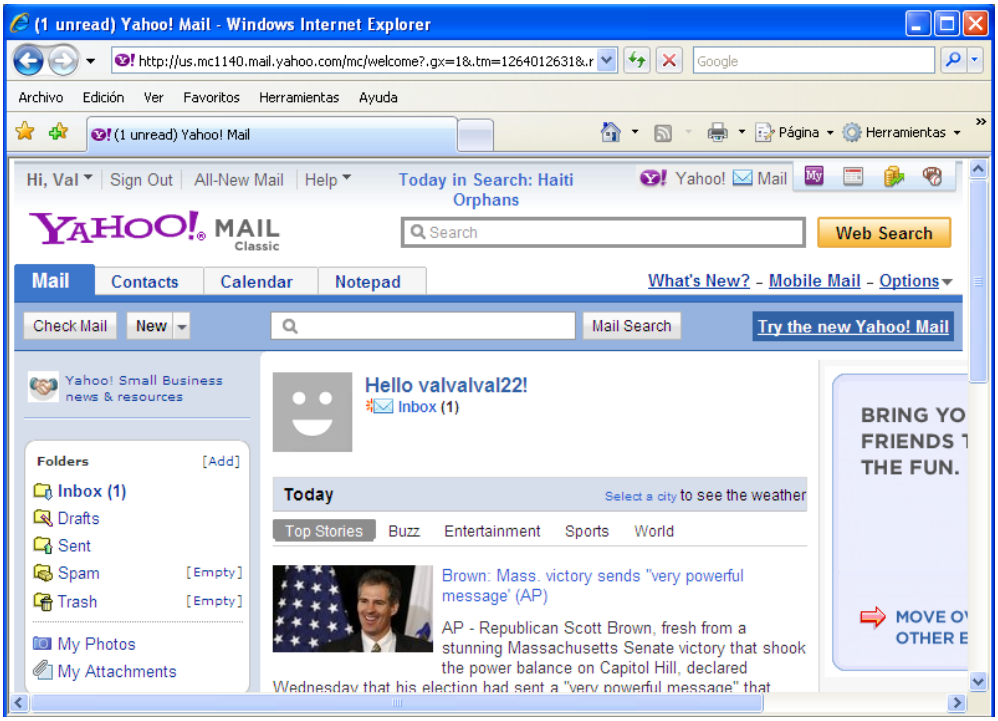

Figura 18: Acceso a correo durante keylog en equipo C.

Durante la experiencia, no tan solo se ha comprobado la efectividad y sencillez del sistema basado en *IRC*, sino también su fragilidad ante cualquier antivirus, o simplemente, la activación del *firewall* de Windows bloqueando el puerto 6667 dejaba inhabilitada la botnet.

Se trata de una herramienta utilizada en los inicios de las redes botnet, y que actualmente es inofensiva, pero en la práctica el resultado es mucho más visual y enriquecedor que sus predecesoras las *botnet HTTP o P2P*.

Por otro lado, resulta preocupante, lo sencillo que ha resultado poner en marcha esta simulación.

Bibliografía

[1] http://www.elgrupoinformatico.com/informe-anual-2009-actividad-cibercriminales-t6751.html
Symantec Corporation, *Informe Anual 2009 de actividad de cibercriminales*. Diciembre 2009.

[2] http://www.m86security.com/trace/traceitem.asp?article=567
M86 Security Labs, *8 botnets responsable del casi 100% del SPAM global*. February 29, 2008.

[3] http://www.rompecadenas.com.ar/articulos/millones-diarios-*botnet*.php
TrendMicro, *8 mil millones de spam diarios por un botnet.* Junio 09.

[4] http://www.elgrupoinformatico.com/espana-encabeza-ranking-mundial-infecciones-virus-t6538.html
PandaLabs, *España encabeza el ranking mundial de infecciones de virus*. 08 Nov 2009.

[5] http://www.eset-la.com/press/informe/waledac_troyano_enamorado.pdf
Autor: Sebastián Bortnik, Analista de Seguridad de ESET para Latinoamérica, *Waledac, el troyano enamorado.* 27 de Marzo del 2009.

[6] http://www.idg.es/computerworld/articulo.asp?id=183857
Computer World por Richi Jennings, *Cuatro pasos para luchar contra los botnets*. 04/05/2007.

[7]http://www.elpais.com/articulo/red/David/Perry/anos/desapareceran/botnets/elpeputeccib/20080214elpcibenr_1/Tes
Merce Molist. *Entrevista a DAVID PERRY, director mundial de formación de Trend Micro*. (5 de setiembre de 2007).

http://www.blackhat.com/presentations/bh-dc-07/Nazario/Presentation/bh-dc-07-Nazario.pdf
por Dr. Jose Nazario para Arbor Networks. *Botnet tracking: Tools, Techniques, and lessons learned*.
(dic 07)

http://pages.cs.wisc.edu/~pb/botnets_final.pdf
Barford, Paul; y, Yegneswaran, Vinod; *An inside look at botnets* (Una mirada interior hacia las botnets). Computer Sciences Department, University of Wisconsin, Madison.

http://www.slideshare.net/chemai64/asegrit-iv-botnets-20-presentation
David Barroso S21sec eCrime Director, *Botnets 2.0: Adquiriendo el control de Internet*.

(Octubre 08)

http://staff.washington.edu/dittrich/misc/malware08-dd-final.pdf
David Dittrich Applied Physics Laboratory University of Washington, Sven Dietric Computer Science Department in Stevens Institute of Technology. *P2P as botnet command and control: a deeper insight.*
(dic 2008)

http://www.isoc.org/isoc/conferences/ndss/08/papers/17_botsniffer_detecting_botnet.pdf
Guofei Gu, Junjie Zhang, and Wenke Lee, School of Computer Science, College of Computing. Georgia Institute of Technology. *BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic*. (Feb. 08)

http://searchsecurity.techtarget.com/news/article/0,289142,sid14_gci1286808,00.html
By Dennis Fisher, Executive. Editor *Storm, Nugache lead dangerous new botnet barrage*. (19 Dec 2007)

HTTP://WWW.TEAM-CYMRU.ORG/READINGROOM/WHITEPAPERS/2008/HTTP-BOTNETS.PDF
BY TEAM CYMRU, *A TASTE OF HTTP BOTNETS.* (july 2008)

http://downloads.hindawi.com/journals/wcn/2009/692654.pdf
Jing Liu, Yang Xiao,1 Kaveh Ghaboosi, Hongmei Deng, and Jingyuan Zhang Department of Computer Science, The University of Alabama, Tuscaloosa, AL 35487-0290, USA. The Centre for Wireless Communications, University of Oulu, P.O. Box 4500, FI-90014, Finland. Intelligent Automation, Inc., Rockville, MD 20855, USA. *Botnet: Classification, Attacks, Detection, Tracing, and PreventiveMeasures.*
(Received 25 December 2008; Revised 17 June 2009; Accepted 19 July 2009)

http://honeyblog.org/junkyard/paper/fastflux-malware08.pdf
By Jose Nazario & Thorsten Holz, Arbor Networks & University of Mannheim. *As the Net Churns: Fast-Flux Botnet Observations.* (September 5, 2008)

http://informaticoysegurata.blogspot.com/2008/10/estudio-sobre-botnets-fast-flux.html
By des para Informático y "Segurata"(Blog sobre seguridad informática), *Estudio sobre botnets fast-flux.*
(domingo 19 de octubre de 2008)

http://blog.s21sec.com/2008/07/tcnicas-de-ocultacin-redes-fast-flux.html
By Jonathan Barajas para S21sec, La seguridad digital del futuro, hoy. *Técnicas de ocultación: Redes Fast-Flux.* (23 julio 08)

http://honeyblog.org/junkyard/paper/storm-leet08.pdf

By Thorsten Holz, Moritz Steinery, Frederic Dahl, Ernst Biersacky, Felix Freiling University of Mannheim and Institut Eur´ecom, Sophia Antipolis. *Measurements and Mitigation of Peer-to-Peer-based Botnets: A Case Study on StormWorm.*(Dic. 08)

http://securitylabs.websense.com/content/Assets/Storm_Worm_Botnet_Analysis_-_June_2008.pdf
By Jun Zhang, Security Researcher, Websense Security Labs. *Storm worm*. (June 2008)

http://www.blackhat.com/presentations/bh-dc-07/Nazario/Presentation/bh-dc-07-Nazario.pdf
Dr. Jose Nazario para Arbor Networks. *Botnet tracking: Tools, Techniques, and lessons learned*. (dic 07)

http://www.eecs.ucf.edu/~czou/research/*P2P*-Botnet-ICCCN09.pdf
Ping Wang, Lei Wu, Baber Aslam and Cliff C. Zou, School of Electrical Engineering & Computer Science University of Central Florida. *A Systematic Study on Peer-to-Peer Botnets.* (May 2009)

http://www.usenix.org/event/hotbots07/tech/full_papers/wang/wang.pdf
Ping Wang Sherri Sparks Cliff Zou, School of Electrical Engineering and Computer Science University of Central Florida, Orlando, FL. *An Advanced Hybrid Peer-to-Peer Botnet.* (Dic 07)

http://www.watchguard.com/docs/whitepaper/wg_botnet-summary_wp_es.pdf
WatchGuard Technologies, Inc, *Derrotando a los Botnets del Futuro*. (2008)

http://www.xombra.com/go_news.php?articulo=3719
By Ing. Roiman Valbuena, Universidad Rafael Belloso Chacín, Maracaibo. Venezuela para Xombra Team. *La amenaza invisible de la tecnología Botnet*. (11-9-08)

http://www.cs.sjsu.edu/faculty/stamp/students/Morparia_Jeet.pdf
By Jeet Morparia, A Writing Project Presented to The Faculty of the Department of Computer Science San Jose State University. *Peer-to-Peer Botnets: Analysis and Detection*. (December 2008)

http://www.borrmart.es/articulo_redseguridad.php?id=1555
Sergio de los Santos. Consultor de Seguridad y director de Formación de Hispasec.
'SPAM', 'BOTNETS' Y TROYANOS BANCARIOS: UNA VISIÓN MÁS REALISTA.

http://www.symantec.com/es/mx/norton/theme.jsp?themeid=*botnet*
By Symantec Corporation, *Bots y Botnet: Una amenaza creciente*.

http://www.symantec.com/es/es/norton/theme.jsp?themeid=dozer_ddos
Por Symantec Corporation, *Ataque DDoS de Dozer*.

http://www.watchguard.com/docs/whitepaper/wg_*botnet*-summary_wp_es.pdf
Por WatchGuard Technologies, *Derrotando a los Botnets del Futuro*.

http://www.viruslist.com/sp/analysis?pubid=207271052
Por Karspersky Lab, *El ecosistema de las botnets*. (17.12.2009)

http://culturacion.com/2009/05/cifras-de-las-botnets-de-este-ano/
Roberto para Culturizacion.com, *Cifras de las Botnets de este año*. (22 Mayo, 2009)

http://www.viruslist.com/analysis?pubid=204792003
por Karspersky Lab, *The botnet bussines*. (May 13 2008)

http://www.eset-la.com/press/concurso/universitario/sexy_view_inicio_botnets_dispositivos_moviles.pdf
Castillo Carlos, Pontifica Universidad Javeriana. *Sexy View: El inicio de las Botnets móviles.*

http://www.honeynet.org/node/138
Por jamie riden, *REAL WORLD FAST-FLUX EXAMPLES.* (Sat, 08/16/2008)

http://staff.science.uva.nl/~delaat/sne-2006-2007/p17/report.pdf
By Reinier Schoof & Ralph Koning, System and Network Engineering, University of Amsterdam, *Detecting peer-to-peer botnets.* February 4, 2007.

http://www.vsantivirus.com/nugache-*P2P*-botnet.htm
Por Angela Ruiz, *Resurge la tecnología P2P para crear redes de BOTS.* (8 May 06)

http://www.iseclab.org/papers/securecomm08_overbot.pdf
Guenther Starnberger, Distributed Systems Group,Vienna Univ. of Technology Christopher Kruegel, UC Santa Barbara, Engin Kirda, Eurecom. *Overbot - A botnet protocol based on Kademlia.* (Dic. 09)

http://culturacion.com/2009/08/estudio-las-botnets-mas-buscadas/
Roberto para Culturización.com, Artículos tecnológicos educativos.
Estudio: Las botnets más buscadas. (18 Agosto, 2009)

http://www.m86security.com/TRACE/traceitem.asp?article=567
Por M86 Security Labs, *8 botnets responsable del casi 100% del SPAM global*. (18 de marzo del 2008)

http://antivirus.interbusca.com/noticias/DreamSystem-sistema-control-botnets-8660.html
Por Panda Labs, *DreamSystem, un sistema de control de botnets.*

http://myitforum.com/cs2/blogs/cmosby/archive/category/68.aspx?PageIndex=3

Week One: Microsoft Malware Protection Center - Threat Research & Response Blog, Microsoft Security Essentials. *Security and Anti-Virus: Microsoft Security Essentials.* Thursday, October 15, 2009